Computer Aided Geometric Design 92 (2022) 102062

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

Implicit surface reconstruction based on a new interpolation/approximation radial basis function ☆

Yajun Zeng, Yuanpeng Zhu *

School of Mathematics, South China University of Technology, Guangzhou, 510641, PR China

ARTICLE INFO

Article history:
Received 5 April 2021
Received in revised form 17 November 2021
Accepted 29 November 2021
Available online 6 December 2021

Keywords:
Implicit surface reconstruction
Interpolation
Approximation
Radial basis function

ABSTRACT

Interpolation and approximation radial basis function (RBF) has been widely used in the implicit surface reconstruction. However, a simple interpolation or approximation method may either have unwanted oscillation around the scattered points, or have low accuracy at scattered points. In this paper, we construct a new representation of RBF, which unites interpolation and approximation. A further adaptive algorithm to determine interpolation and approximation is presented. With the new representation, we can develop quasi-interpolation method from an approximation method into a method combining interpolation and approximation. And we can determine interpolation or approximation at each of scattered points without solving large linear systems. Experiments show that the given method is precision-controlled and fast in implicit surface reconstruction.

1. Introduction

Implicit surface reconstruction plays an important role in Computer Aided Geometric Design (CAGD). Compared with other surface representations such as subdivision surface, a noisy, incomplete or non-uniform distributed data set is more suitable to be reconstructed with implicit surface, since it can handle geometries with complex topology and shape.

RBF is an effective method in implicit curve and surface reconstruction for scattered data sets, since it has a very simple form. And the surface reconstructed by RBF has obvious details. Many researches on RBF have been presented during the last three decades. In 1989, the variational nature of interpolation and approximation of scattered data with RBF was proved (Nira, 1989). Using RBF for surface reconstruction can change the function approximation problems into multivariate optimization, and this benefit was recognized and applied in reconstruction from scattered points (Savchenko et al., 1995) and medical imaging (Carr et al., 1997, 2001). However, some drawbacks of RBF revealed. Since the RBF methods are global methods, each interpolated value is influenced by all the data, resulting in high computational cost for solving large linear systems. And the interpolation matrix may become ill-conditioned when the number of interpolated points is very large.

Compactly supported radial basis function (CSRBF) has been introduced to overcome the problem of high computational cost (Wendland, 1995), which can significantly decrease computational cost. Because it leads to a system of linear equations with a sparse matrix. And it became a useful tool for surface reconstruction from scattered data (Kojekine et al., 2003; Levin, 2003). Several approaches to approximate or interpolate using CSRBF were presented (Ohtake et al., 2003), leading

☆ Editor: Rida T. Farouki.

* Corresponding author.
E-mail addresses: zhuyuanpeng@csu.edu.cn, ypzhu@scut.edu.cn (Y. Zhu).

https://doi.org/10.1016/j.cagd.2021.102062
0167-8396/

to a simpler and faster computation procedure of fitting scattered data. However, CSRBF has a narrow band support and is weak in repairing incomplete regions in the sampled data sets. Thus the support size should be chosen carefully, and a multi-level scheme should be applied (Ohtake et al., 2005; Liu and Wang, 2012).

Quasi-interpolation is also a practical method for approximation. A new kind of quasi-interpolation method using RBF was presented (Liu and Wang, 2012), which can avoid solving large linear systems. Later a quasi-interpolation method based on CSRBF was proposed (Liu and Wang, 2015). And a closed-form formulation (Liu et al., 2016) was presented to further avoid solving small-medium size linear systems. Quasi-interpolation can generate satisfactory results with small shape approximation error. And it has widely applied in several fields of industry like 3D models' establishing (Song et al., 2015a, 2015b; Hu et al., 2018; Crivellaro et al., 2017; Dou et al., 2019), industrial design (Li et al., 2019, 2020) and data-fitting (Bracco et al., 2017; Costarelli and Vinti, 2019). However, the quasi-interpolation schemes presented by Liu and Wang (2012, 2015) are just approximation methods, which can not meet the need for the different accuracy requests at each point.

For the choices of RBF, the two typical ones are radial power (Shepard, 1968) and Wendland's CSRBF (Wendland, 1995). The former as an interpolation method has higher accuracy, and we can get precise interpolation without solving linear systems, but unwanted oscillation may exist around the scattered points. On the contrary, the latter is a member of Wendland RBFs family. As an approximation method, it has lower shape approximation error but with lower accuracy at scattered points when used in reconstruction. When trying to get precise interpolation, we need to solve linear systems.

In this paper, we construct a new representation of CSRBF, which unites interpolation and approximation and combines the advantages of radial power and Wendland's CSRBF. The superiority of our method mainly lies on three parts. Firstly, our representation can determine whether to interpolate or approximate at each of given data points by shape parameters. Secondly, we develop the quasi-interpolation method by Liu and Wang (2012) into a method combining interpolation and approximation, so that we can realize premise-controlled surface reconstruction. Thirdly, a further adaptive algorithm to determine interpolation and approximation at each point is presented, with which we can interpolate all the input scattered points and holding a low shape approximation error in our multi-level scheme.

The rest of this paper is organized as follows. In Section 2, we construct a new representation of CSRBF which realizes the union of interpolation and approximation. And we introduce the quasi-interpolation presented by Liu and Wang in implicit surface reconstruction, and develop it from a simple approximation method into a method combining interpolation and approximation in Section 3. In Section 4, we discuss the choices for shape parameter a_i, and present an adaptive algorithm to determine interpolation and approximation at each of data points. We further discuss a suitable choice for shape parameter λ_i in Section 5. And we will demonstrate and discuss the advantages of our approach by experimental results in Section 6, and conclude in Section 7.

2. A new interpolation/approximation RBF

Given a set of scattered data pairs $V = \{(\mathbf{v}_i, f_i)\}_{i=0}^n \subset \mathbb{R}^3 \times \mathbb{R}$ sampled from a function $f : \mathbb{R}^3 \to \mathbb{R}$, the function f can be approximated by the sum of a series of weighted normalized radial basis functions:

$$\hat{f}(\mathbf{v}) = \sum_{i=0}^{n} c_i \varphi_i(\mathbf{v}) = \sum_{i=0}^{n} c_i \frac{\phi_i(\mathbf{v})}{\sum_{k=0}^{n} \phi_k(\mathbf{v})}, \tag{1}$$

where $\boldsymbol{v} \in \mathbb{R}^3$ and $c_i \in \mathbb{R}^1$ is weight coefficient. And

$$\phi_i(\mathbf{v}) = \phi(\lambda_i \frac{\|\mathbf{v}_i - \mathbf{v}\|}{\rho_i}), \tag{2}$$

with $\phi(r)$ a radial basis function and its value depends on the Euclidean distance between $\mathbf{v}_i$ and $\mathbf{v}$. And ρ_i is the support size at the point $\mathbf{v}_i$. The formula Eq. (1) is known as Shepard's method (1968). The value λ_i is assigned as $\sqrt{\mu_i}/\eta_i$, where μ_i is a shape parameter and $\eta_i = \inf_{k \neq i} \|\boldsymbol{v}_k - \boldsymbol{v}_i\|$. In this paper, we set $\mu_i = \eta_i^2$, and thus $\lambda_i = 1$. And we will further discuss the choices of μ_i in Section 5.

To interpolate the data pairs, c_i is determined by the following constraints

$$\sum_{i=0}^{n} c_i \varphi_i(\mathbf{v}_j) = f_j, \ \ j = 0, 1, \ldots, n, \tag{3}$$

and this can be written into a linear system as follows:

$$\boldsymbol{M}\boldsymbol{c} = \boldsymbol{f}, \tag{4}$$

where

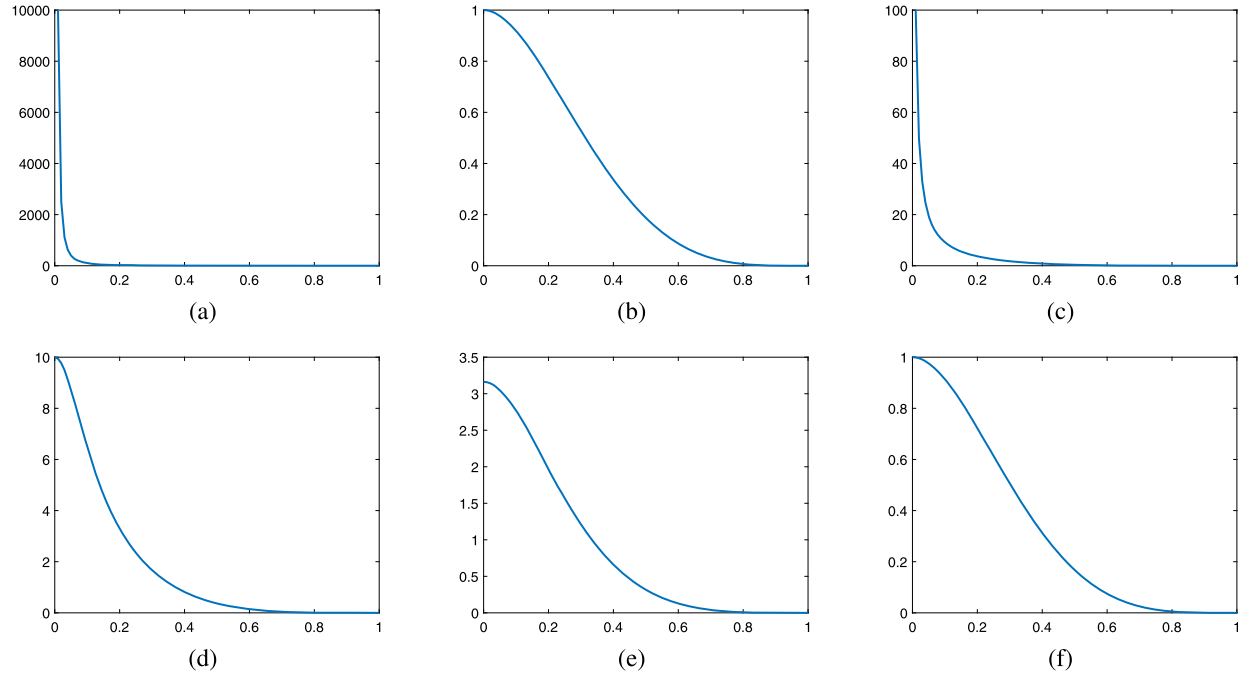

Fig. 1. The curve image of the three RBFs: (a) radial power Eq. (7), with $d = -2$, (b) Wendland's CSRBF Eq. (8), (c) our representation, with $a_i = 0$, (d) the given CSRBF Eq. (9), with $a_i = 0.01$ (e) the given CSRBF Eq. (9), with $a_i = 0.1$ and (f) the given CSRBF Eq. (9), with $a_i = 1$.

$$\boldsymbol{M} = \begin{bmatrix} \varphi_0(\mathbf{v}_0) & \varphi_1(\mathbf{v}_0) & \cdots & \varphi_n(\mathbf{v}_0) \\ \varphi_0(\mathbf{v}_1) & \varphi_1(\mathbf{v}_1) & \cdots & \varphi_n(\mathbf{v}_1) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_0(\mathbf{v}_n) & \varphi_1(\mathbf{v}_n) & \cdots & \varphi_n(\mathbf{v}_n) \end{bmatrix}. \tag{5}$$

When $\varphi_i(\mathbf{v}_j)$ satisfies the following Kronecker product property

$$\varphi_i(\mathbf{v}_j) = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases} \tag{6}$$

$\boldsymbol{M}$ becomes an identity matrix, and the solution of the linear system is $\boldsymbol{c} = \boldsymbol{f}$.

There are many choices for the RBF $\phi(r)$. The two typical ones are radial power Eq. (7) (Shepard, 1968) and Wendland's CSRBF Eq. (8) (Wendland, 1995). Radial power has a form as follows:

$$\phi(r) = r^{-d},\ d > 0, \tag{7}$$

which has high accuracy at scattered points as an interpolation method. By radial power, we can get precise interpolation without solving large linear systems since its coefficient matrix M is an identity matrix. However, its derivatives of each order at the interpolated scattered points are zero. And this will restrict the accuracy around these points, resulting unwanted oscillation as is shown in Fig. 3(a).

A widely-used Wendland's CSRBF in Wendland RBFs family is as follows:

$$\phi(r) = (1-r)_+^4 (4r+1). \tag{8}$$

The CSRBF in Eq. (8) can produce good results in curve and surface approximation. However, it has low accuracy at scattered points, as is shown in Fig. 3(b). And it needs to solve a linear system when trying to get precise interpolation, see in Ohtake et al. (2005).

To make full use of the advantages of these two RBFs, we construct a new CSRBF $\phi(r)$ as follows:

$$\phi(r) = (1-r)_+^4 (4r+1) \left(a_i + r^2\right)^{-0.5}, \tag{9}$$

where $a_i \in [0, \infty)$ is a shape parameter.

In Fig. 1, we can find that the new CSRBF Eq. (9) is similar as the radial power Eq. (7) when $a_i = 0$. When $a_i > 0$, the resulting function curve of the new CSRBF is similar as Wendland's CSRBF Eq. (8). And the smaller the shape parameter a_i is, the greater it changes with the support size.

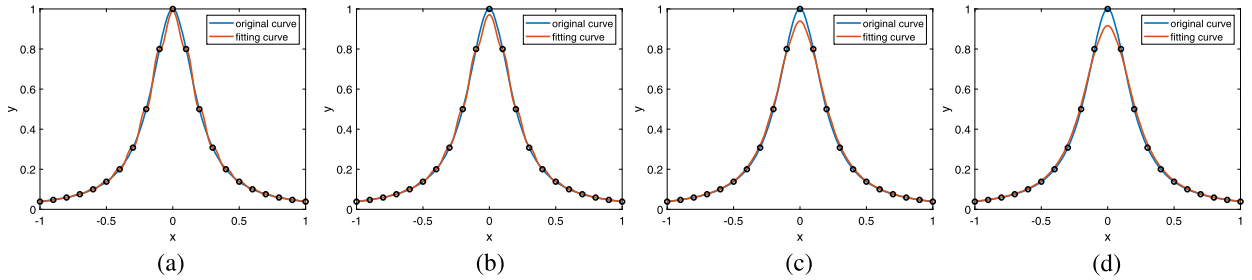

Fig. 2. Fitting results with different shape parameters a_i: (a) interpolate at all points, with $a_i = 0$, (b) approximate at all points, with $a_i = 0.01$, (c) approximating at all points, with $a_i = 0.1$, and (d) approximating at all points, with $a_i = 1$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

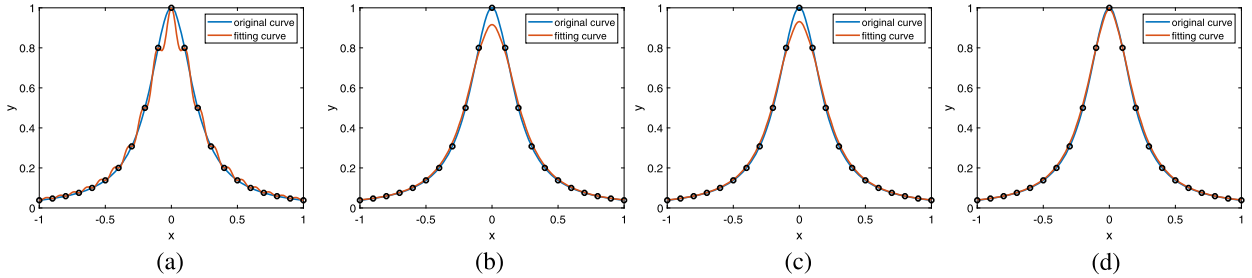

Fig. 3. The comparison of the results by three methods: (a) Radial power, with $d = 2$ in Eq. (7), (b) Wendland's CSRBF in Eq. (8), (c) our method, approximating at all points, with $a_i = 0.2$, and (d) our method, approximating at all points with $a_i = 0.2$ but $a_{11} = 0$, which means interpolating at $x = 0$.

In fact, the new representation unites interpolation and approximation together, which means we can determine whether to interpolate or approximate at each point by adjusting the shape parameter a_i. In fact, if we choose $a_i = 0$, the new CSRBF in Eq. (9) can be rewritten as follows:

$$\phi(r) = (1-r)_+^4 (4r+1) r^{-1}. \tag{10}$$

It is similar to the radial power Eq. (7). And when $r = 0$, $\phi_i(0) = \phi(0/\rho_i) = +\infty$. With normalization, we have

$$\begin{aligned}\varphi_j(\boldsymbol{v}) = \frac{\phi_j(\boldsymbol{v})}{\sum_{k=0}^n \phi_k(\boldsymbol{v})} &= \frac{(1-r_j(\boldsymbol{v}))_+^4 (4r_j(\boldsymbol{v})+1)(a_j + r_j^2(\boldsymbol{v}))^{-0.5}}{\sum_{k=0}^n (1-r_k(\boldsymbol{v}))_+^4 (4r_k(\boldsymbol{v})+1)(a_k + r_k^2(\boldsymbol{v}))^{-0.5}} \\ &= \frac{(1-r_j(\boldsymbol{v}))_+^4 (4r_j(\boldsymbol{v})+1) \prod_{m\neq j}(a_m + r_m^2(\boldsymbol{v}))^{0.5}}{\sum_{k=0}^n (1-r_k(\boldsymbol{v}))_+^4 (4r_k(\boldsymbol{v})+1) \prod_{m\neq k}(a_m + r_m^2(\boldsymbol{v}))^{0.5}},\end{aligned}$$

where $r_j(\boldsymbol{v}) = \|\mathbf{v} - \mathbf{v}_j\|/\rho_j$. Thus, when $a_i = 0$, we have

$$\varphi_j(\mathbf{v}_i) = \frac{\phi_j(\mathbf{v}_i)}{\sum_{k=0}^n \phi_k(\mathbf{v}_i)} = \begin{cases} 1, & j = i, \\ 0, & j \neq i, \end{cases} \tag{11}$$

which can meet the need of interpolation at the point $\boldsymbol{v}_i$. When $a_i \in (0, \infty)$, for any $r \geq 0$, the given CSRBF Eq. (9) is nonnegative, which forms an approximation CSRBF.

Here we define interpolation error and shape approximation error to display the performance of reconstruction. The interpolation error is defined by $|f(\boldsymbol{v}_i) - \hat{f}(\boldsymbol{v}_i)|$ at data point $\boldsymbol{v}_i$. And the shape approximation error is defined by $|f(\boldsymbol{x}_i) - \hat{f}(\boldsymbol{x}_i)|$, where $\boldsymbol{x}_i$ is the test points uniformly resampled. The number of test points is much larger than the number of given scattered points, which is usually a hundred times as many as the original data points. The two error statistics are of significance, because the interpolation error can reflect the accuracy at the given scattered points, while the shape approximation error can reflect the approximation effect on the whole curve or surface.

The shape parameter $a_i \in [0, \infty)$ determines interpolation and approximation at the given data points $\boldsymbol{v}_i$. The closer a_i approaches 0, the closer the result approaches interpolation at the data point $\boldsymbol{v}_i$, and it will have higher accuracy at the data point $\boldsymbol{v}_i$. On the contrary, the farther a_i away from 0, the result will have less shape approximation error, as is shown in Fig. 2. And with the adjustments of a_i at each point, our method is precision-controlled.

Considering the function $f(x) = \frac{1}{1+25x^2}$, there are some numerical experiments to demonstrate the superiority of our approach shown in Fig. 3, and the interpolation error and shape approximation error statistics of these methods are shown

Table 1
The error statistics of three methods in 1D case.

Method	Interpolation error		Shape approximation error	
	Average	Maximum	Average	Maximum
Radial power in Eq. (7)	0	0	0.0172	0.1201
Wendland's CSRBF in Eq. (8)	0.0122	0.0839	0.0129	0.0839
Our method with all $a_i = 0.2$	0.0100	0.0688	0.0110	0.0688
Our method with all $a_i = 0.2$ but $a_{11} = 0$	0.0051	0.0185	0.0066	0.0236

in Table 1. For all three methods, we set the support size $\rho_i = 0.25$, which actually can be set to be any positive value. The parameter d of radial power is set as 2 for the best. The result in Fig. 3(a) by radial power in Eq. (7) is rough, with frequent oscillation around the scattered points, because it is a complete interpolation method. As for Wendland's CSRBF in Eq. (8), although the fitting curve is smooth, the error at $x = 0$ is huge, as is shown in Fig. 3(b). In Fig. 3(c), we set all $a_i = 0.2$ for the best when regarding our method as a complete approximation method. And it shows that our new CSRBF is similar to Wendland's CSRBF in Eq. (8) when considering approximating only. But we can adjust the value of a_{11} to zero to make the fitting curve pass the sampled point at $x = 0$. As is shown in Fig. 3(d), the resulting curve is smooth and the shape approximation error is the smallest, with high accuracy at each scattered point.

3. United interpolation and approximation scheme

To make this paper be self-contained, we will briefly introduce quasi-interpolation presented by Liu and Wang (2012). In fact, quasi-interpolation is totally a method of approximation. It can not interpolate scattered points unless solving large linear systems, especially the points are reorganized in multi-level case. And in this section, we develop the quasi-interpolation from a simple approximation method to a scheme combining interpolation and approximation. And we take some significant change in our multi-level scheme. With our strategies on choosing shape parameter a_i, our multi-level scheme can interpolate all the given scattered points and produce good results at the same time.

3.1. Single-level scheme

Our united interpolation and approximation scheme in single level is similar as the quasi-interpolation scheme presented by Liu and Wang (2012) in single-level, which has the following form:

$$g(\boldsymbol{v}) = \sum_{i=0}^{n} (c_i + h_i(\boldsymbol{v}))\, \varphi_i(\boldsymbol{v}), \tag{12}$$

where $\varphi_i(\boldsymbol{v}) = \phi(\lambda_i \|\mathbf{v} - \mathbf{v}_i\|/\rho_i) / \sum_{i=0}^{n} \phi(\lambda_i \|\mathbf{v} - \mathbf{v}_i\|)$, c_i is unknown coefficient, and $h_i(\boldsymbol{v})$ is a local quadratic approximation in a small vicinity of $\mathbf{v}_i$. In the above formula, a local orthogonal coordinate system (u, v, w) is created at each point $\boldsymbol{v}_i$. In the local orthogonal coordinate system, we define $\mathbf{e}_3$ to be the normal at this point, $\mathbf{e}_1$ to be the rotation of the normal, $\mathbf{e}_2$ to be the cross product of $\mathbf{e}_3$ and $\mathbf{e}_1$, where $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ are the axis of the coordinate system. And $h_i(\boldsymbol{v})$ can be set to be $h_i(\boldsymbol{v}) = w - w(u, v)$, where $w(u, v) = au^2 + 2buv + cv^2$ is a quadric polynomial used to fit the local surface at $\boldsymbol{v}_i$.

For exact interpolation, the form Eq. (12) needs to satisfy the constrains Eq. (3). For implicit curve or surface reconstruction, we have $f_j = 0$, $j = 0, 1, .., n$, so the constrains Eq. (3) can be written as follows:

$$\sum_{i=0}^{n} c_i \varphi_i(\boldsymbol{v}_j) = -\sum_{i=0}^{n} h_i(\boldsymbol{v}_j) \varphi_i(\boldsymbol{v}_j), \quad j = 0, 1, ..., n, \tag{13}$$

and an approximate solution can take

$$c_i = -\sum_{j=0}^{n} h_j(\boldsymbol{v_i})\, \varphi_j(\boldsymbol{v_i}). \tag{14}$$

Hence, for implicit curve and surface reconstruction in single-level case, quasi-interpolation scheme presented by Liu and Wang (2012) can be written as follows:

$$g(\boldsymbol{v}) = \sum_{i=0}^{n} \left(-\sum_{j=0}^{n} h_j(\boldsymbol{v_i})\, \varphi_j(\boldsymbol{v_i}) + h_i(\boldsymbol{v}) \right) \varphi_i(\boldsymbol{v}), \tag{15}$$

With the $\varphi_i(\boldsymbol{v})$ chosen by our new CSRBF in Eq. (9), if the shape parameter $a_i = 0$ for data point $\boldsymbol{v}_i$, we have

$$\varphi_i(\mathbf{v}_j) = \begin{cases} 1, & i = j, \\ 0, & i \neq j, \end{cases} \tag{16}$$

resulting in $c_i = -\sum_{j=0}^{n} h_j(\boldsymbol{v_i})\varphi_j(\boldsymbol{v_i}) = -h_i(\boldsymbol{v_i})$. Thus we will have $g(\boldsymbol{v}_i) = 0$, which mean that the fitting function $g(\boldsymbol{v})$ can interpolate the point $\boldsymbol{v}_i$. Since the shape parameters a_i can vary at different points, the interpolation at point $\boldsymbol{v}_i$ will not affect the choices of interpolation and approximation at other points. And the scheme of single-level quasi-interpolation can be translated into a scheme combining interpolation and approximation together.

In our single level scheme, the shape parameter a_i will be determined by a center value a as well as an adaptive adjustment algorithm, which we will state in Section 4. And the support size ρ_i is determined by the density of $V = \{\boldsymbol{v}_i\}_{i=0}^{n}$ similarly as the one in Ohtake et al. (2005), where the scattered data points are organized into an octree with each leaf cells containing no more than 8 points. And the support size ρ_i is equal to 3/4 of the average diagonal length of the leaf cells for all points.

The computation of single level scheme is very fast since the evaluation for coefficients is independent. However, single level scheme has weaknesses in repairing incomplete data and dealing irregularly sampled data. And the reconstruction surface has a narrow band support, and it requires the polygonization grid to be smaller than the width of the support band. Hence, a multi-level scheme is need to solve these weaknesses. Experiment results are shown in Section 5.

3.2. Multi-level scheme

Similar to the multi-level quasi-interpolation method by Liu and Wang (2012), a multi-scale hierarchy of point set $\{V_k\}_{k=1}^{M}$ (where $V_k = \{\boldsymbol{v}_i^k\}_{i=0}^{n^k}$) is constructed in our multi-level scheme. And the result will be refined by progressively adding basis functions according to the points at different levels in our scheme.

The approximation functions are in the following form:

$$g^k(\boldsymbol{v}) = g^{k-1}(\boldsymbol{v}) + \delta^k(\boldsymbol{v}), \tag{17}$$

where $k = 1, 2, \ldots, M$ and

$$\delta^k(\boldsymbol{v}) = \sum_{i=0}^{n^k} \left(c_i^k + h_i^k(\boldsymbol{v})\right) \varphi_i^k(\boldsymbol{v}), \tag{18}$$

is in the form of our single-level scheme. In formula Eq. (18), n^k is the number of points in level k, $h_i^k(\boldsymbol{x})$ is local quadratic approximation to V_k, and a basis function is defined as $g^0(\boldsymbol{v}) = 0$. With the same idea in single-level case, an approximate solution for the coefficients c_i^k can take as follows:

$$c_i^k = -g^{k-1}(\boldsymbol{v}_i^k) - \sum_{j=0}^{n^k} h_j^k\left(\boldsymbol{v}_i^k\right)\varphi_j^k\left(\boldsymbol{v}_i^k\right). \tag{19}$$

However, in multi-level quasi-interpolation method by Liu and Wang (2012), we can not interpolate the given scattered points without solving large linear system, since Wendland's CSRBF in Eq. (8) is a totally approximation CSRBF, it can not interpolate the scattered points. And we should know that the points used in the multi-level quasi-interpolation are reorganized, not the original given scattered points. That is, $V_M \neq V$ in multi-level quasi-interpolation. If we consider multi-level interpolation method by Ohtake et al. (2005), large linear systems need to be solved.

In our multi-level scheme, we let $V_M = V = \{\boldsymbol{v}_i\}_{i=0}^{n}$. That is, in the final level, we take the original given scattered points $V = \{\boldsymbol{v}_i\}_{i=0}^{n}$ instead of the reorganized points. With our scheme, we can overcome the weaknesses of single level scheme and interpolate the given scattered points without solving large linear systems at the same time.

If we take $a_i^M = 0$ at the point $\boldsymbol{v}_i^M$ in the final level V_M, we have

$$\begin{aligned} \delta^M(\boldsymbol{v}_i^M) &= \sum_{j=0}^{n^M} \left(c_j^M + h_j^M(\boldsymbol{v}_i^M)\right)\varphi_j^M(\boldsymbol{v}_i^M) \\ &= c_i^M + h_i^M(\boldsymbol{v}_i^M) = c_i^M \\ &= -g^{M-1}(\boldsymbol{v}_i^M) - \sum_{j=0}^{n^M} h_j^M\left(\boldsymbol{v}_i^M\right)\varphi_j^M\left(\boldsymbol{v}_i^M\right) \\ &= -g^{M-1}(\boldsymbol{v}_i^M) - h_i^M\left(\boldsymbol{v}_i^M\right) = -g^{M-1}(\boldsymbol{v}_i^M). \end{aligned}$$

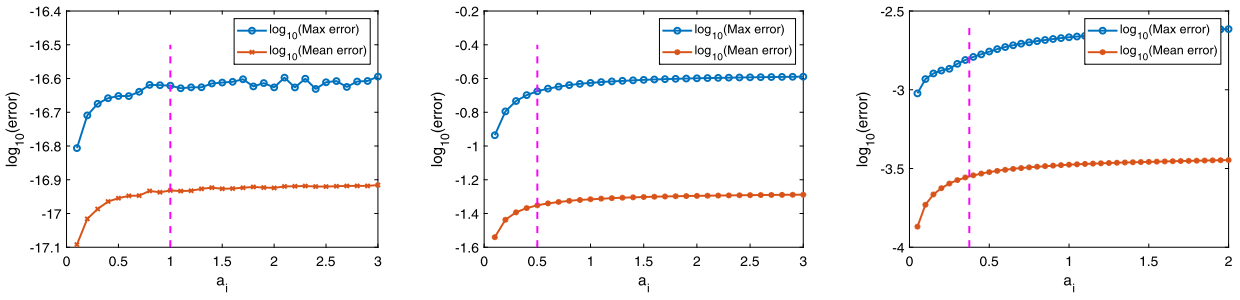

Fig. 4. The relationship between interpolation error and the shape parameter a_i on different data sets. All the points have the same value on a_i, and the dotted line is the $a = 2\hat{\rho}/\rho_1$: (a) circle, (b) heart, (c) Cassini oval.

Since we let the points set in final level V_M to be the given scattered points $V = \{\boldsymbol{v}_i\}_{i=0}^n$, we have

$$\begin{aligned} g^M(\boldsymbol{v}_i) &= g^{M-1}(\boldsymbol{v}_i) + \delta^M(\boldsymbol{v}_i) \\ &= g^{M-1}(\boldsymbol{v}_i^M) + \delta^M(\boldsymbol{v}_i^M) \\ &= g^{M-1}(\boldsymbol{v}_i^M) - g^{M-1}(\boldsymbol{v}_i^M) \\ &= 0, \end{aligned}$$

which means that the resulting fitting function $g^M(\boldsymbol{v})$ interpolates the original scattered points.

The support size ρ^k in the k level is recursively defined by $\rho^k = \rho^{k-1}/2$ in our scheme. And an initial support size will take $\rho^1 = \alpha L$, where $\alpha = 0.75$ and L is the diagonal length of the bounding box of V. And the reconstruction level M is determined by $M = \lceil -\log_2 (\hat{\rho}/2\rho^1) \rceil$ as in Ohtake et al. (2005), where $\hat{\rho}$ is equal to 3/4 of the average diagonal length of the leaf cells that contain no more than eight points of V.

An adaptive parameter δ is defined by the minimum number of data points falling into the support of each scattered points $\mathbf{v}_i$. And it is always adopted to make sure the robustness of our scheme. If the number of points falling in the support ρ_i^k at point $\boldsymbol{v}_i^k$ is smaller than δ in level k, the support size shall be enlarged as $\rho_i^k = 1.1\rho_i^k$. And check the number of points in the support again, until the number of points falling in the support at this point is no less than δ or all the points in this level are in the support. A suitable choice of δ can set to be 16.

In our multi-level scheme, the shape parameter a_i^k in the level k is also assigned with a center value a^k and the same adaptive algorithm. Specially, we will take $a_i^M = 0$ for all the points in the final level M. This means that we can interpolate all the scattered points without solving large linear system with our multi-level scheme. And we can hold a less or almost the same shape approximation error as multi-level quasi-interpolation. Experiments results and a further discussion will show in Section 6.

4. Adaptive choices for shape parameter a_i

A main superiority of our method is that we can choose interpolation or approximation at each point by determining the local shape parameter a_i. To further make use of this superiority, we define a practical indictor and present an adaptive algorithm to determine the shape parameters a_i at each point, and it can help to get better implicit curve and surface reconstruction results.

4.1. Single-level scheme

Since the results in 1D case in Fig. 3 shows that a suitable choice of shape parameter a_i has relationship with the location and distribution of the input data points, we design some numerical experiments to find a suitable rule for a_i. For simplicity, we consider setting all a_i to be a same value a, and apply our single-level scheme into three curves: circle, heart and Cassini oval. The interpolation error and the shape approximation error are shown in Figs. 4 and 5.

From Figs. 4 and 5, we can find that the larger a_i is, the larger interpolation error is, and the smaller shape approximation error is. And in the test for circle, we find when the data is completely symmetric, the shape parameter a_i should be the same at each point. This inspires us to set a center value $a_i = a$ for all the points, and make adjustments on each point according to its distribution. In different cases, center value a can set to be any nonnegative value according to the actual demand. If we want to interpolate all the points, we can simply let $a = 0$. Otherwise, the larger a is, the less the shape approximation is, but with the less accuracy at scattered points. And a suitable a for approximation in our single-level scheme can set to be $a = 2\hat{\rho}/\rho_1$, where $\hat{\rho}$ is equal to 3/4 of the average diagonal length of the leaf cells that contain no more than eight points of V. And $\rho_1 = \alpha L$, where $\alpha = 0.75$ and L is the diagonal length of the bounding box of V. This

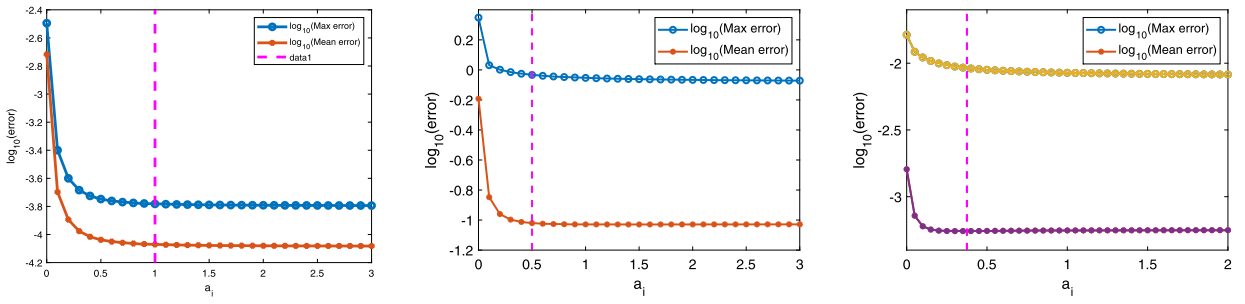

Fig. 5. The relationship between shape approximation error and the shape parameter a_i on different data sets. All the points have the same value on a_i, and the dotted line is the $a = 2\hat{\rho}/\rho_1$: (a) circle, (b) heart, (c) Cassini oval.

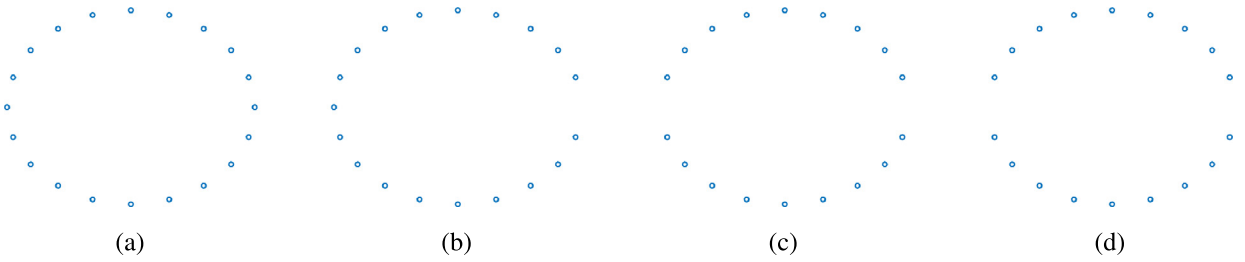

Fig. 6. Some circle data sets used in the experiments: (a) points uniformly distributed in circle, (b) the data in (a) with a missing point, (c) the data in (a) with two missing points which locate on the opposite of each one, and (d) the data in (a) with two missing points which locate two sides of a point.

choice for $a = 2\hat{\rho}/\rho_1$ is an empirical value. And we can calculate that $a = 1$, $a = 0.5$ and $a = 0.375$ for circle, heart and Cassini oval, which are indeed suitable choices.

In Fig. 6, we design some circle data sets to further discuss the suitable choices for shape parameter a_i. With experiments on data sets Figs. 6(b), 6(c) and 6(d), we find that the choice of a_i at $\boldsymbol{v}_i$ is concerned about the error of local quadratic approximation $h_i(\boldsymbol{v})$, and the detail location of the points in the support of $\boldsymbol{v}_i$. And we define a practical indicator as follows:

$$E_i = \sqrt{\sum_{\mathbf{v}_j \in S_i} h_j^2(\mathbf{v}_i)(1 - r_j(\boldsymbol{v}_i))_+^4}, \tag{20}$$

where $r_j(\boldsymbol{v}) = \|\boldsymbol{v} - \boldsymbol{v}_j\|/\rho_j$, with ρ_j the support size of $\boldsymbol{v}_i$, and S_i the support field of $\mathbf{v}_i$.

And we assign a suitable value to a_i with the indicator E_i by the following adaptive algorithm.

Algorithm Adaptive choices for shape parameters a_i.

Input: the coordinates and normals of data point $\boldsymbol{v}_i$, $i = 0, 1, ..., n$.
Output: the shape parameters a_i of each point $\boldsymbol{v}_i$, $i = 0, 1, ..., n$.

```
Select the center value a, and calculate the indicator E_i at each point by the formulas given in (20);
Calculate the mean value μ_e, maximum value max(E_i) and minimum value min(E_i) of indicator E_i;
for i = 1 to n do
  a_i ⇐ a;
  if max(E_i) − min(E_i) > 0 then
    x ⇐ (E_i − μ_e)/(max(E_i) − min(E_i));
    if x > 0 then
      a_i = a/(1 + x);
    else
      a_i = a(1 − x);
    end if
  end if
end for
```

The adaptive algorithm contains an easy judgement block for each data point $\mathbf{v}_i$ only, which will consume little time. Two steps are required in the adaptive algorithm. First, choose the center value a. Second, make adjustments to the shape parameters a_i at each point. We apply our adaptive algorithm into the data set in Fig. 6, and compare the differences on

Table 2
The error comparison on parameter adjustments in 2D case.

Data set	Number of points	Adjustments	Error			
			Interpolation error		Shape approximation error	
			Average	Maximum	Average	Maximum
6(a)	20		1.17E-17	2.39E-17	8.49E-05	1.65E-04
		✓	1.17E-17	2.39E-17	8.49E-05	1.65E-04
6(b)	19		5.49E-05	1.98E-04	4.47E-04	6.80E-03
		✓	1.60E-05	3.09E-05	4.17E-04	6.50E-03
6(c)	18		1.15E-04	1.97E-04	8.03E-04	6.80E-03
		✓	3.42E-05	6.66E-05	7.44E-04	6.50E-03
6(d)	18		2.58E-04	7.95E-04	1.20E-03	8.20E-03
		✓	1.69E-04	5.38E-04	1.20E-03	8.70E-03

Fig. 7. The reconstruction results on 3D model in our single-level scheme: (a) bunny, (b) bimba, (c) dragon.

whether adopting adjustments at each point or simply assigning a to each point in the following Table 2. For data set as shown in Fig. 6(a), the adjustments make no difference, since all the points are symmetric. For data sets shown in Figs. 6(b), 6(c) and 6(d), we find that the adaptive algorithm has significant reduction in interpolation error. That is, our method can obviously improve the accuracy on scattered points. The same strategies on choosing center value a and adaptive algorithm in single-level scheme are suitable for 3D scattered points, as is shown in Fig. 7.

Remark 1. The idea for constructing the indictor E_i given in Eq. (20) comes from Eq. (15). The main part of interpolation error of $g(\boldsymbol{v})$ in Eq. (15) at the point $\boldsymbol{v}_i$ is:

$$\sum_{j=0}^{n} h_j(\boldsymbol{v}_i)\varphi_j(\boldsymbol{v}_i) = \sum_{j=0}^{n} h_j(\boldsymbol{v}_i)(1-r_j(\boldsymbol{v}_i))_+^4(4r_j(\boldsymbol{v}_i)+1)(a_i+r_j(\boldsymbol{v}_i))^{-0.5}. \tag{21}$$

Since our target is to estimate a suitable value of the shape parameter a_i, we want to avoid a_i in Eq. (21). So we take the main part of Eq. (21) as $\sum_{j=0}^{n} h_j(\boldsymbol{v}_i)(1-r_j(\boldsymbol{v}_i))_+^4$. Moreover, we want to create an indictor which can reflect the errors of point $\boldsymbol{v}_i$ at all the local quadratic approximation, no matter it is positive or negative. That is, we require $\sum_{j=0}^{n} |h_j(\boldsymbol{v}_i)|(1-r_j(\boldsymbol{v}_i))_+^4$ or $\sum_{j=0}^{n} h_j^2(\boldsymbol{v}_i)(1-r_j(\boldsymbol{v}_i))_+^4$. With large quantities of experiments, we choose the indictor as the one given in Eq. (20).

4.2. Multi-level scheme

For our multi-level scheme, the adaptive algorithm also works. However, we need to develop a new strategy to choose the center value a, since in different level k a suitable choice for a^k may be different. Furthermore, in our multi-level scheme, we want to interpolate all the input scattered points in the final level, hence $V_M = V$ and $a^M = 0$ for the final level M. An empirical strategy is that the center value a^k in the level k can be set as $a^k = 1/k^2$, and in the final level M we set $a_M = 0$. Some experiments are shown in Fig. 8 and the corresponding error statistic table in Table 3. In the first row, we set $a^k = 1$ simply. The results for the first row seem to be smooth but we can not interpolate all the points. In the second row, simply let $a^k = 0$ in all level k. The results show that though we can interpolate all the scattered points in this way, but the reconstruction result is not smooth enough, and has higher shape approximation error. While we adopt the strategy $a^k = 1/k^2$ and $a^M = 0$, we can find that the resulting function $g^M(\boldsymbol{v})$ can interpolate all the scattered points and have a smooth reconstruction results at the same time. And the shape approximation error is also the smallest as is shown in Table 3.

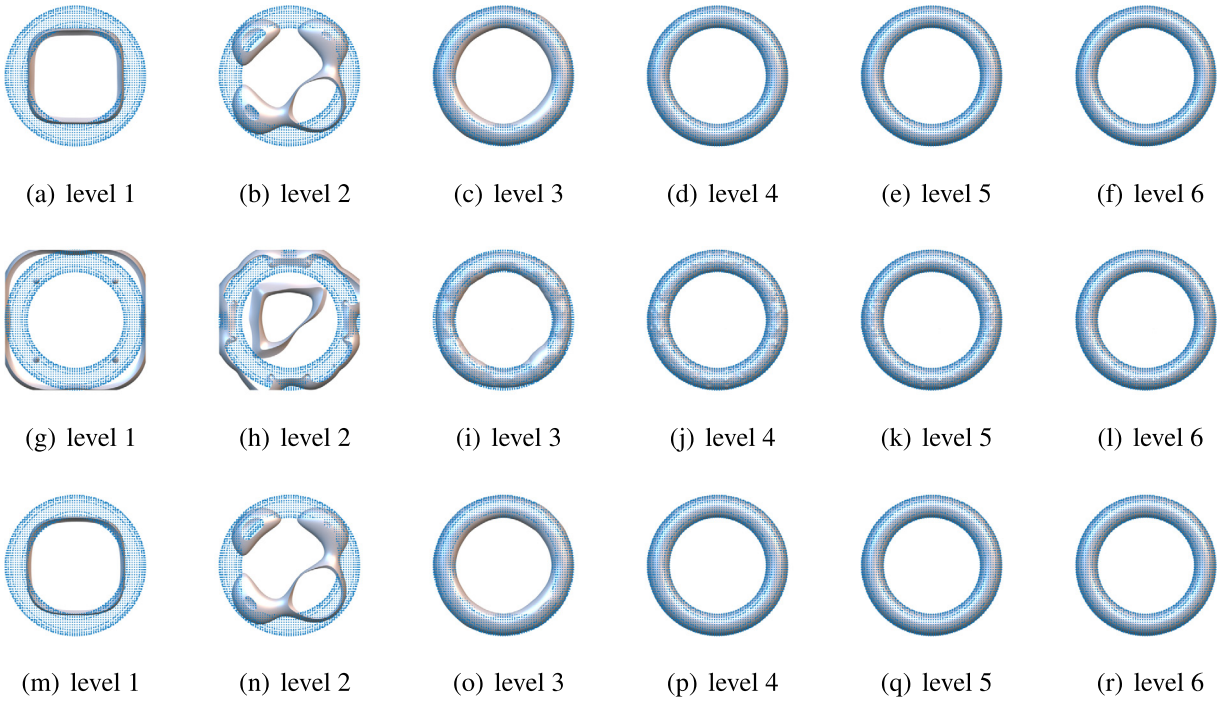

Fig. 8. The reconstruction results on torus when assigning different center value a by our multi-level scheme, and the blue points are the input data points. The first row is the reconstruction results in different level with all $a^k = 1$, the second row is the reconstruction results in different level with all $a^k = 0$, and the third row is the reconstruction results in different level with $a^k = 1/k^2$ and $a^M = 0$.

Table 3
The error comparison with different strategy for center value a^k in multi-level case.

Figure	Model	Number of points	Strategy for a	Error			
				Interpolation error		Shape approximation error	
				Average	Maximum	Average	Maximum
8(f)	torus	6.4k	$a^k = 1$	9.80E-05	6.86E-04	1.09E-04	3.74E-04
8(l)			$a^k = 0$	0	0	1.69E-04	1.16E-03
8(r)			$a^k = 1/k^2, a^M = 0$	0	0	1.09E-04	3.68E-04

Fig. 9. Reconstruction results on teddy with different shape parameter a_i in single-level: (a) $\mu = 0.00001$, (b) $\mu = 0.1\rho^2$, (c) $\mu = 0.001$, (d) $\mu = 0.01$, (e) $\mu_i = \eta_i^2$.

5. Adaptive choices for shape parameter λ_i

In this section we discuss a suitable choice for the shape parameter λ_i in Eq. (2). We take attempts to assign different value $\mu_i = \mu$. In Figs. 9 and 10, experiments results are shown. Detailed error statistics are in Table 4. And we can find that when we simply let $\mu_i = \eta_i^2$, good results produce. In this case, $\lambda_i = 1$, and we will have a simple form for Eq. (2):

$$\phi_i(\mathbf{v}) = \phi_i(\|\mathbf{v}_i - \mathbf{v}\|) = \phi(\frac{\|\mathbf{v}_i - \mathbf{v}\|}{\rho_i}). \tag{22}$$

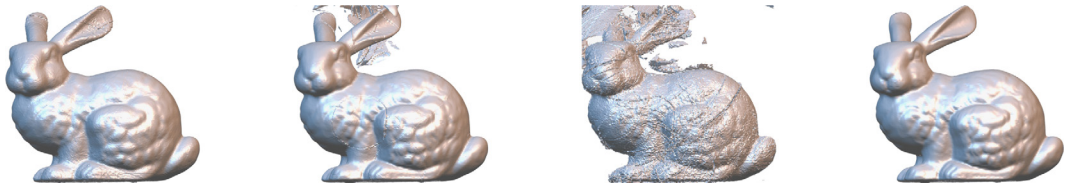

Fig. 10. Reconstruction results on bunny with different shape parameter a_i in multi-level: (a) $\mu = 0.0001$, (b) $\mu = 0.1\rho^2$, (c) $\mu = 0.01$, (d) $\mu_i = \eta_i^2$.

Table 4
The error comparison with different strategy shape parameter μ_i.

Figure	Model	Number of points	Strategy for μ	Error			
				Interpolation error		Shape approximation error	
				Average	Maximum	Average	Maximum
9(a)	teddy	13.3k	$\mu = 0.00001$	2.12E-04	5.30E-03	5.16E-04	1.90E-02
9(b)			$\mu_i = 0.1\rho_i^2$	8.62E-05	1.79E-02	5.24E-04	2.01E-02
9(c)			$\mu = 0.001$	7.33E-05	9.20E-03	5.40E-04	2.01E-02
9(d)			$\mu = 0.01$	9.94E-04	1.90E-02	8.38E-04	1.13E-02
9(e)			$\mu_i = \eta_i^2$	7.51E-05	2.78E-03	5.20E-04	2.00E-02
10(a)	bunny	34.8k	$\mu = 0.0001$	0	0	6.89E-05	1.18E-03
10(b)			$\mu_i = 0.1\rho_i^2$	0	0	5.39E-05	9.72E-04
10(c)			$\mu = 0.01$	0	0	1.32E-04	2.02E-03
10(d)			$\mu_i = \eta_i^2$	0	0	5.04E-05	9.70E-04

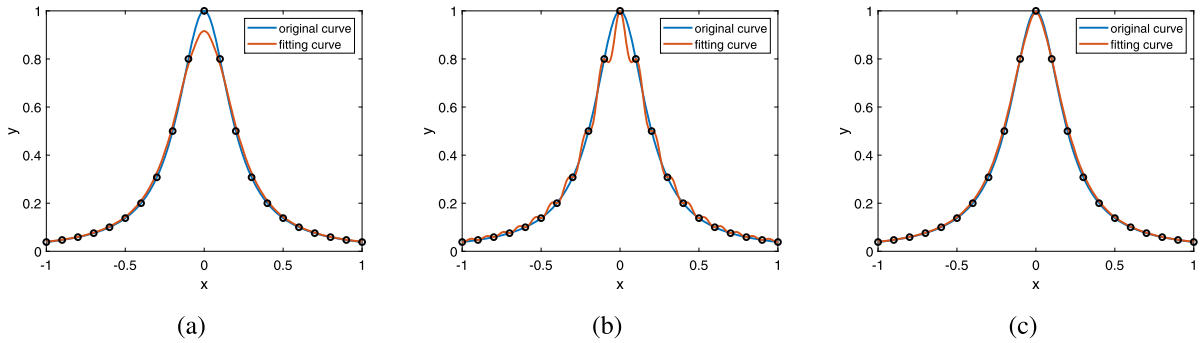

Fig. 11. The comparison of the results of by Liu and Wang's method and our method in single-level: (a) the reconstruction result by Liu and Wang's method, (b) the reconstruction result by Liu and Wang's method with radial power as RBF, and (c) the reconstruction result by our method, with all $a_i = 0.2$ but $a_{11} = 0$, which means interpolating at $x = 0$.

6. Results and discussion

In this section, we compare our method with quasi-interpolation (Liu and Wang, 2012), including single-level scheme, multi-level scheme, the robustness for dealing with noisy data and highly-non-uniformly data, and a further comparison with Poisson method (Kazhdan and Hoppe, 2013) and MPU method (Ohtake et al., 2003) is also included.

6.1. Comparison with the quasi-interpolation

In single-level, we can determine interpolation or approximation at each point $\mathbf{v}_i$ by our method, while quasi-interpolation is a simple approximation method. Similar to the Section 2, considering the function $f(x) = \frac{1}{1+25x^2}$, we observe the reconstruction results by quasi-interpolation and the results by our method. Comparing with the results in Figs. 11(a) and 11(c), we can find that our method is obviously superior than Liu and Wang's method. And if we want to interpolate the point at $x = 0$, we can simply let the corresponding a_i to be 0, without solving large linear systems. In Fig. 11(b), we change the Wendland's CSRBF in Eq. (8) into radial power in Eq. (7) in quasi-interpolation. And we can find that the reconstruction curve has frequent oscillation around the scattered points. It means that in single-level, a local interpolation sometimes is better than global interpolation for whole points. Moreover, with the choices of shape parameters a_i, we can control the accuracy at each data point, and realize precision-controlled reconstruction.

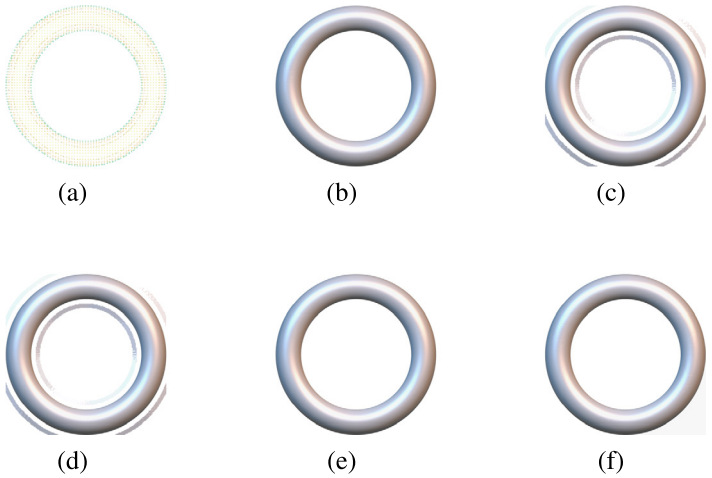

Fig. 12. Reconstruction results on torus by two methods: (a) the input points, (b) the original surface, (c) the result with Liu and Wang's method in single-level, (d) the result with our method in single-level, (e) the result with Liu and Wang's method in multi-level with adaptive parameter $\delta = 16$, (f) the result with our method in multi-level with adaptive parameter $\delta = 16$.

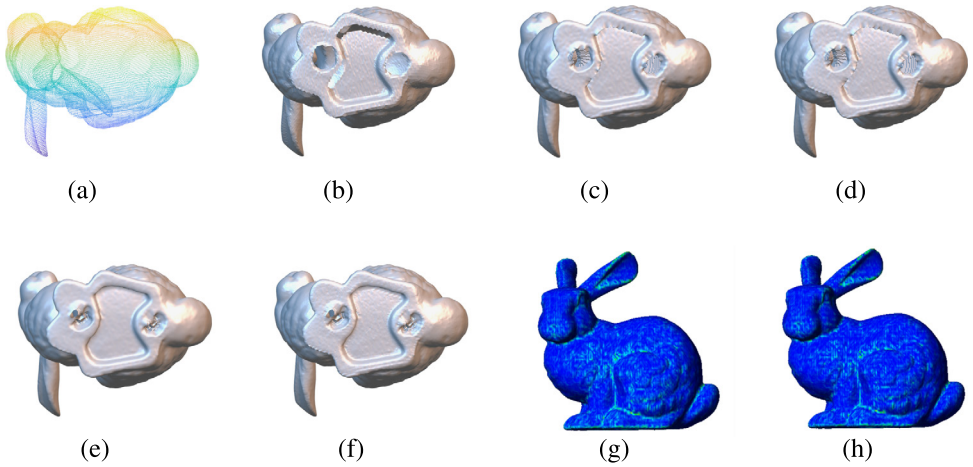

Fig. 13. Reconstruction results on bunny by two methods: (a) the input points, (b) the original mesh, (c) the result with Liu and Wang's method in single-level scheme, (d) the result with our method in single-level scheme, (e) the result with Liu and Wang's method in multi-level with adaptive parameter $\delta = 16$, (f) the result with our method in multi-level with adaptive parameter $\delta = 16$, (g) and (h) are the color maps of shape approximation error for (d) and (e). Among the color maps, the blue color is for zero error, the red color is for the shape approximation error larger than 0.000972033, and other colors denote the shape approximation error from 0 to 0.000972033.

In multi-level cases, in fact, multi-level quasi-interpolation (Liu and Wang, 2012) is totally an approximation method. And it can produce good shape approximation results. However, it can not interpolate without solving large linear systems. And we should notice that in multi-level quasi-interpolation all the points are reorganized. Even when we change the Wendland's CSRBF in Eq. (8) in multi-level quasi-interpolation into radial power in Eq. (7), we can only interpolate the reorganized points, not the original given scattered points. And our multi-level scheme overcomes the difficulty on interpolating the original scattered points by setting $V^M = V$ and all $a_i^M = 0$, while holding the advantages of multi-level quasi-interpolation at the same time.

Some experiments are done to show the feasibility and superiority of our multi-level scheme. In Fig. 12, we can find that there exist extra zero-level sets when we reconstruct the torus by the two methods in single-level. That is because the reconstruction surface has a narrow band support in single-level scheme. With our multi-level scheme, we can solve this problem in Fig. 12(f) similarly as the multi-level quasi-interpolation in Fig. 12(e). In Fig. 13, our multi-level scheme also overcomes the weakness in repairing holes in single-level similarly as the multi-level quasi-interpolation. Furthermore, when we apply the two methods in a highly-nonuniform point cloud in Fig. 14. We can find that our multi-level scheme also overcomes the weakness in dealing irregularly sampled data similarly as the multi-level quasi-interpolation, and can better preserve the details.

Detailed comparison of error statistics can be found in the Table 5. Note that mesh surfaces are generated from reconstructed implicit surfaces by Bloomenthal's method in Bloomenthal (1988). And the shape approximation errors are

Table 5
The error statistics of the two methods in 3D case.

Figure	Model	Number of points	Liu's and Wang's method		Our method		Type	Error			
			Times (sec.)					Interpolation error		Shape approximation error	
			Once core	Eight cores	Once core	Eight cores		Average	Maximum	Average	Maximum
12(c)	torus	6.4k	0.835	0.547			single-level	9.52E-06	4.90E-05	4.43E-04	5.56E-03
12(d)					0.833	0.535	single-level	8.82E-06	4.31E-05	4.43E-04	5.56E-03
12(e)			1.211	0.629			multi-level	1.23E-04	1.01E-03	1.06E-04	3.87E-04
12(f)					1.327	0.643	multi-level	0	0	1.08E-04	3.68E-04
13(c)	bunny	34k	0.139	0.081			single-level	1.35E-05	2.65E-04	3.60E-05	1.10E-03
13(d)					0.143	0.092	single-level	6.82E-06	1.47E-04	3.51E-05	1.15E-03
13(e)			1.648	0.605			multi-level	1.36E-04	2.42E-03	5.50E-05	9.85E-04
13(f)					1.823	0.746	multi-level	0	0	5.02E-05	9.72E-04
14(c)	MaxPlanck	25.4k	0.074	0.048			single-level	1.81E-03	5.01E-02	1.24E-02	4.49E-01
14(d)					0.081	0.051	single-level	1.10E-03	3.05E-02	1.19E-02	4.49E-01
14(e)			1.128	0.741			multi-level	1.55E-02	7.38E-01	1.08E-02	2.52E-01
14(f)					1.337	0.782	multi-level	0	0	8.62E-03	2.49E-01

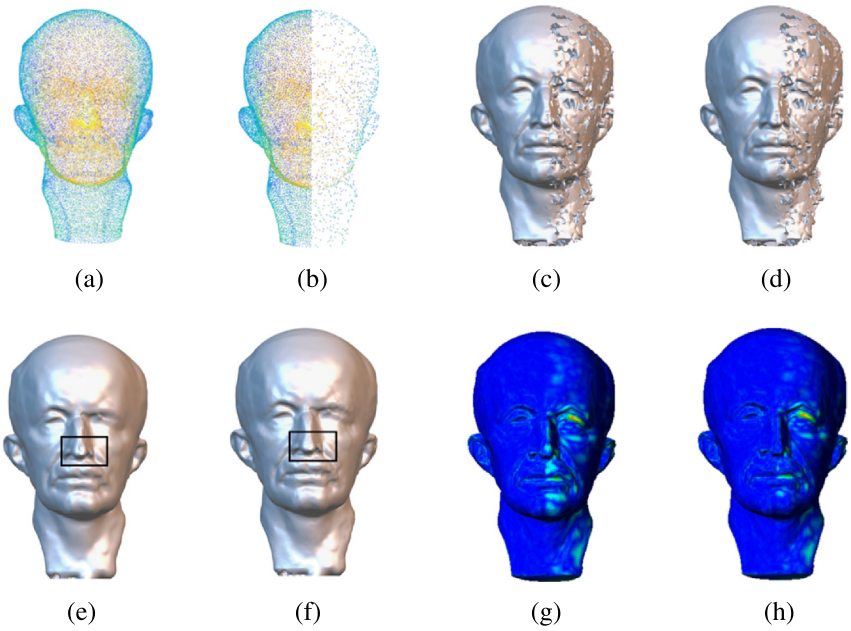

Fig. 14. Reconstruction results on MaxPlanck by two methods: (a) the original points, (b) the input point cloud, where only 10% points are reserved in the right side, (c) the result with Liu and Wang's method in single-level scheme, (d) the result with our method in single-level scheme, (e) the result with Liu and Wang's method in multi-level with adaptive parameter $\delta = 16$, (f) the result with our method in multi-level with adaptive parameter $\delta = 16$, (g) and (h) are the color maps of shape approximation error for (e) and (f). Among the color maps, the blue color is for zero error, the red color is for the shape approximation error larger than 0.248949, and other colors denote the shape approximation error from 0 to 0.248949.

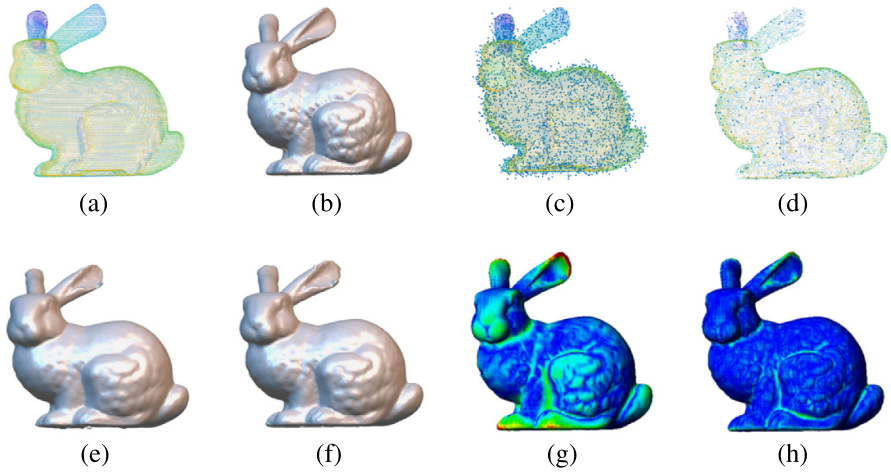

Fig. 15. Reconstruction results on a noisy point set of bunny in multi-level with an adaptive parameter $\delta = 16$: (a) original point set, (b) original mesh, (c) a noisy point cloud generated from (a), where 5% noisy points randomly distributed in the range of 1/50 of the bounding box's diagonal length, the blue points are the noisy points, (d) the resultant point set after removing outliers (Liu et al., 2012), and (e) the reconstruction result by multi-level quasi-interpolation, (f) the reconstruction result by our multi-level scheme, (g) and (h) are the color maps of shape approximation error for (e) and (f). Among the color maps, the blue color is for zero error, the red is for the shape approximation error larger than 0.00454061.

generated by the method presented in Silva et al. (2005). With our method, in single-level, we have significant reduction in interpolation error, while the shape approximation of our method is no larger than the one of Liu and Wang's method. While in multi-level scheme, we can also see that our method has the shape approximation error no larger than the one of Liu and Wang's method, and it can interpolate all the given data points at the same time.

6.2. Processing noisy point data

When processing noisy data, similar with Liu and Wang's method, we can extend our scheme by introduce a regularization parameter T_k, and $c_i = f_i/(1 + T_k)$. Instead of $T_k = 0.5k$ in Liu and Wang's method, we take $T_k = 0.25k$ in our scheme. In Fig. 15, we construct a large noisy points set from bunny model, then use the method presented by Liu and Wang (2012)

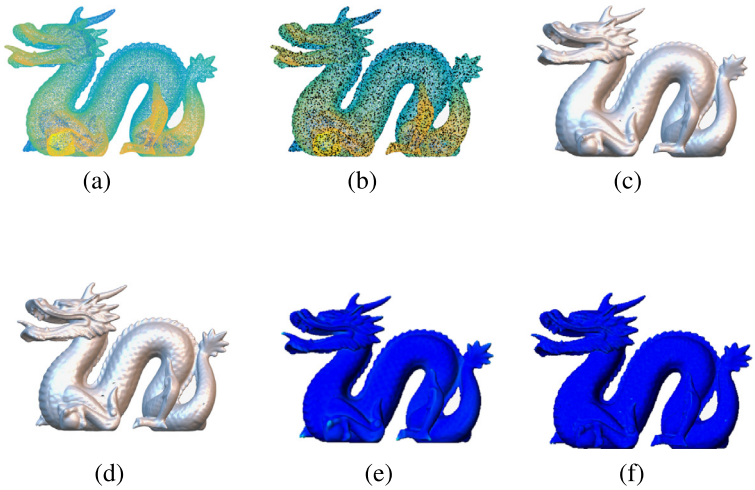

Fig. 16. Reconstruction results on dragon with noisy normals in multi-level with an adaptive parameter $\delta = 16$: (a) original point set, (b) point-rendered dragon model with noisy normals, 10% normals are set to zero, (c) the reconstruction result by multi-level quasi-interpolation from original noisy point set (b), and (d) the reconstruction result by our multi-level scheme from original noisy point set (b), (f) and (g) are the color maps of shape approximation error for (d) and (e). The blue color is for zero error, the red color is for the shape approximation error larger than 0.0130042.

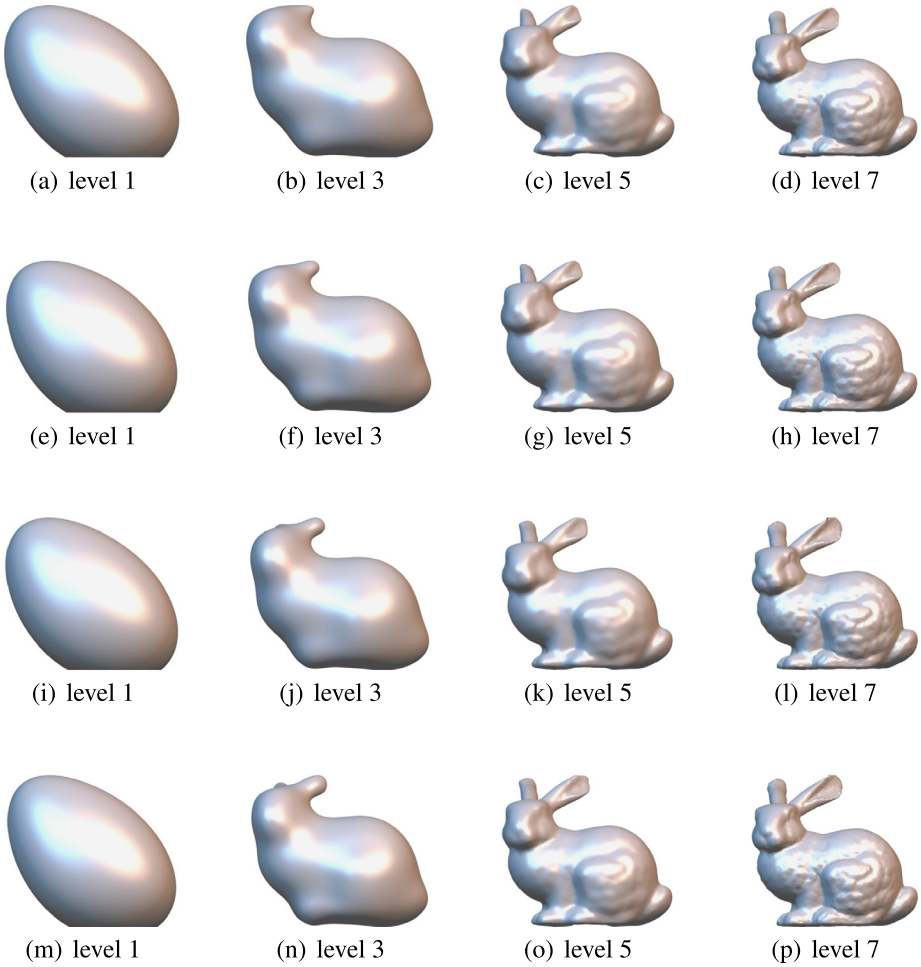

Fig. 17. The comparison of reconstruction results in different levels with an adaptive parameter $\delta = 16$. The first row shows the reconstruction results by Liu and Wang's method in levels $1, 3, 5, 7$ with $T_k = 0.5k$. The second row shows the reconstruction results by our method in levels $1, 3, 5, 7$ with $T_k = 0.5k$. The third row shows the reconstruction results by Liu and Wang's method in levels $1, 3, 5, 7$ with $T_k = 0.25k$. And the fourth row shows the reconstruction results by our method in levels $1, 3, 5, 7$ with $T_k = 0.25k$.

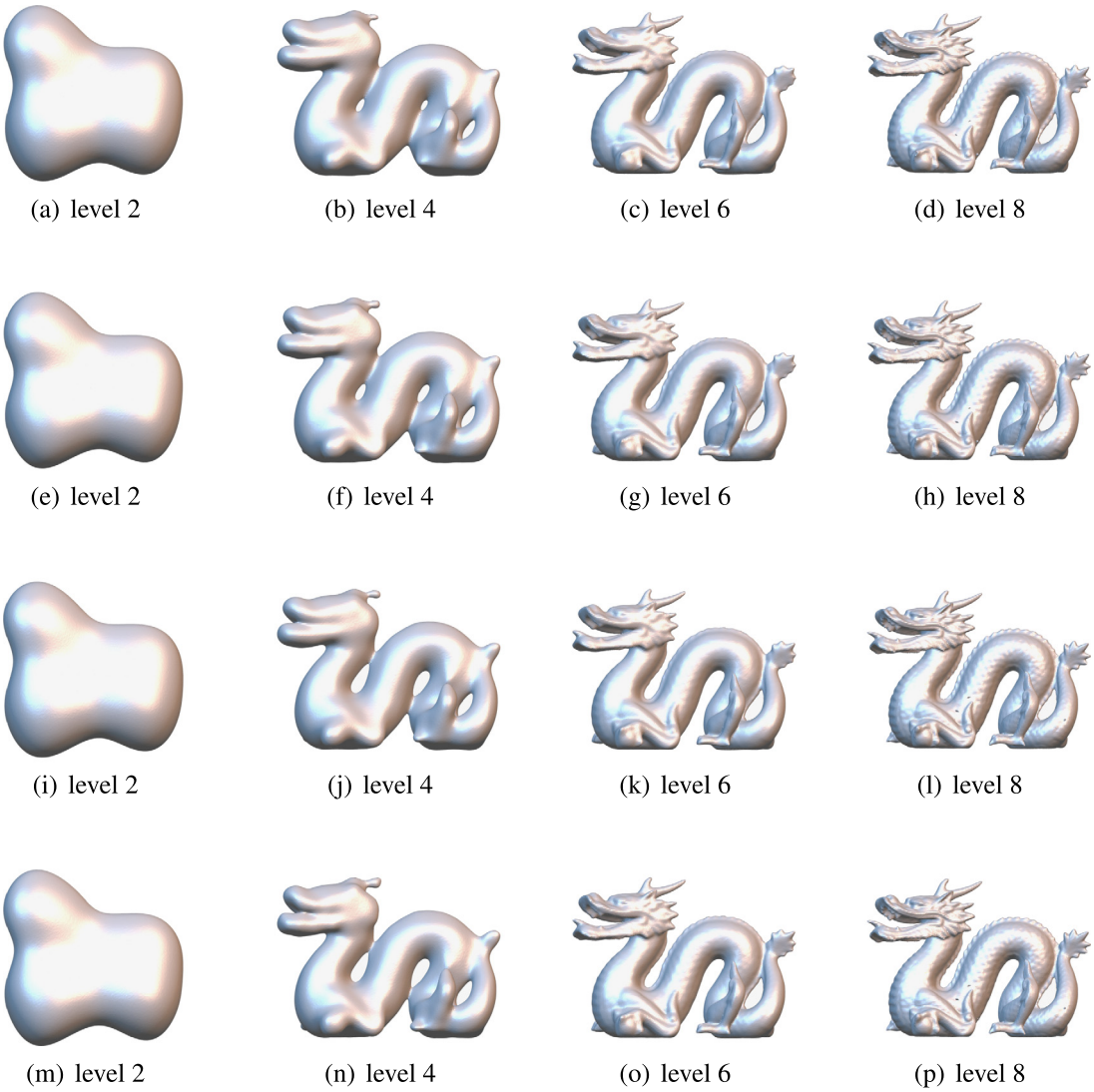

Fig. 18. The comparison of reconstruction results in different levels with an adaptive parameter $\delta = 16$. The first row shows the reconstruction results by Liu and Wang's method in levels $2, 4, 6, 8$ with $T_k = 0.5k$. The second row shows the reconstruction results by our method in levels $2, 4, 6, 8$ with $T_k = 0.5k$. The third row shows the reconstruction results by Liu and Wang's method in levels $2, 4, 6, 8$ with $T_k = 0.25k$. And the fourth row shows the reconstruction results by our method in levels $2, 4, 6, 8$ with $T_k = 0.25k$.

to remove the outliers as is shown in the Fig. 15(d). And we apply Liu and Wang's method and our method to reconstruct from the data set as Fig. 15(d). Comparing the results in Figs. 15(e) and 15(f), we can find that our method has more details. And the color maps of shape approximation error show that our method approximates obviously better than Liu and Wang's method when dealing wrong location information. In Fig. 16, 10% normals of the dragon model are set to zero, and we can also get a better implicit surface reconstruction results by our method. Further detailed reconstruction results in different levels are shown in Figs. 17 and 18.

From Figs. 17 and 18, we can observe that better results will produce when we take $T_k = 0.25k$ instead of $T_k = 0.5k$. A detailed error table can be seen in the Table 6, where we can find that our method can perform better in both the cases $T_k = 0.25k$ and $T_k = 0.5k$. And $T_k = 0.25k$ is a suitable choice for our method.

6.3. Comparison with Poisson reconstruction and MPU

Furthermore, we compare our results with Liu and Wang's method, Screened Poisson reconstruction method (Kazhdan and Hoppe, 2013), and MPU method (Ohtake et al., 2003). We create a highly non-uniform point set from the Armadillo model as is shown in Fig. 19, and compare the results of these methods. The reconstruction results by Screened Poisson reconstruction method are seen to be very smooth, while many details can not be observed as is shown in the Fig. 19(b),

Table 6
The error comparison by two methods for some models with noises in 3D cases.

Figure	Model	Number of points	Liu and Wang's method		Our method		T_k	Level	Error			
			Times (sec.)						Interpolation error		Shape approximation error	
			One core	Eight cores	One core	Eight cores			Average	Maximal	Average	Maximal
17(a)	bunny	34k	0.017	0.004			0.5k	1	8.44E-03	5.18E-02	1.04E-02	6.52E-02
17(e)					0.019	0.004	0.5k	1	8.56E-03	5.08E-02	1.05E-02	6.53E-02
17(i)			0.014	0.003			0.25k	1	8.60E-03	5.14E-02	1.05E-02	6.50E-02
17(m)					0.018	0.004	0.25k	1	8.74E-03	5.03E-02	1.07E-02	6.50E-02
17(b)			0.051	0.011			0.5k	3	7.60E-03	4.20E-02	5.20E-03	5.05E-02
17(f)					0.055	0.016	0.5k	3	7.12E-03	4.01E-02	4.84E-03	4.94E-02
17(j)			0.049	0.012			0.25k	3	6.31E-03	3.80E-02	4.12E-03	4.11E-02
17(n)					0.052	0.015	0.25k	3	5.77E-03	3.64E-02	3.79E-03	3.92E-02
17(c)			0.179	0.072			0.5k	5	4.02E-03	2.37E-02	1.49E-03	2.39E-02
17(g)					0.199	0.089	0.5k	5	3.74E-03	2.24E-02	1.37E-03	2.22E-02
17(k)			0.177	0.069			0.25k	5	2.35E-03	1.55E-02	8.42E-04	1.37E-02
17(o)					0.198	0.086	0.25k	5	2.12E-03	1.42E-02	7.48E-04	1.17E-02
17(d)			1.701	0.654			0.5k	7	2.39E-03	1.46E-02	5.63E-04	9.34E-03
17(h)					1.743	0.75	0.5k	7	2.20E-03	1.33E-02	5.17E-04	8.51E-03
17(l)			1.712	0.654			0.25k	7	9.85E-04	6.83E-03	2.81E-04	4.56E-03
17(p)					1.743	0.754	0.25k	7	8.46E-04	5.78E-03	2.59E-04	4.54E-03
18(a)	dragon	100k	0.018	0.006			0.5k	2	5.77E-03	2.83E-02	9.12E-03	3.02E-02
18(e)					0.019	0.007	0.5k	2	5.41E-03	2.84E-02	8.53E-03	3.10E-02
18(i)			0.017	0.006			0.25k	2	5.15E-03	2.84E-02	8.45E-03	3.13E-02
18(m)					0.019	0.007	0.25k	2	4.83E-03	2.84E-02	7.93E-03	3.20E-02
18(b)			0.107	0.032			0.5k	4	4.01E-03	2.19E-02	2.92E-03	2.03E-02
18(f)					0.128	0.043	0.5k	4	3.58E-03	1.92E-02	2.62E-03	1.85E-02
18(j)			0.107	0.032			0.25k	4	2.89E-03	1.85E-02	2.28E-03	1.74E-02
18(n)					0.128	0.043	0.25k	4	2.54E-03	1.59E-02	2.01E-03	1.66E-02
18(c)			0.992	0.473			0.5k	6	2.28E-03	1.26E-02	9.51E-04	1.58E-02
18(g)					1.007	0.569	0.5k	6	2.03E-03	1.06E-02	8.61E-04	1.64E-02
18(k)			0.993	0.471			0.25k	6	1.20E-03	9.44E-03	5.95E-04	1.49E-02
18(o)					1.012	0.577	0.25k	6	1.04E-03	8.52E-03	5.30E-04	1.50E-02
18(d)			7.423	3.682			0.5k	8	1.43E-03	8.30E-03	5.05E-04	1.52E-02
18(h)					7.656	3.811	0.5k	8	1.27E-03	7.50E-03	4.65E-04	1.58E-02
18(l)			7.421	3.681			0.25k	8	5.33E-04	4.54E-03	2.39E-04	1.35E-02
18(p)					7.656	3.812	0.25k	8	4.54E-04	3.91E-03	2.16E-04	1.30E-02

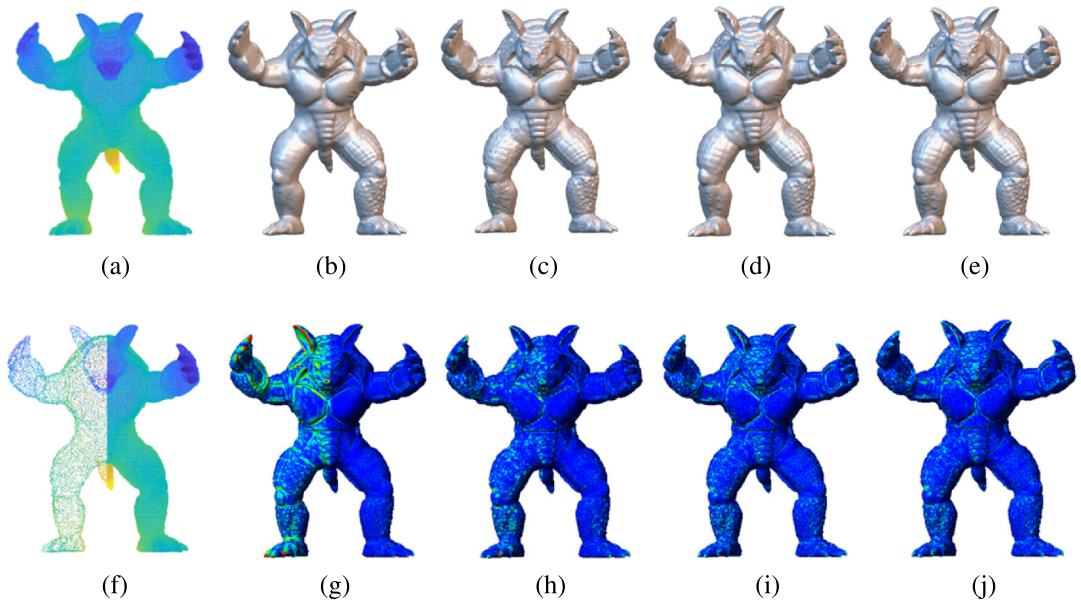

Fig. 19. Reconstruction results on a noisy point set: (a) original point set, (f) the highly non-uniform model created from (a), (b)-(e) are the reconstruction results by Poisson reconstruction, MPU method, Liu and Wang's method and our method respectively from (f), and (g)-(j) are the color maps of shape approximation error for (b)-(e). Among the color maps, the blue color is for zero error, the red color is for the error larger than 0.918231, and other colors denote the error from 0 to 0.918231.

Table 7
Error table for reconstruction of Armadillo.

Figure	Model	Number of points	Method	Number of resultant triangles	Times (sec.)	Error			
						Interpolation error		Shape approximation error	
						Average	Maximal	Average	Maximal
19(b)	Armadillo	94.1k	Poisson	169,978	total 3.655	0.30754	2.46716	0.09049	2.80857
19(c)			MPU	169,080	total 11.684	0.00136	0.01310	0.05681	1.23490
19(d)			Liu and Wang's method	169,248	1.876(33.513)	0.08519	3.50706	0.05023	1.00036
19(e)			Our method	169,508	1.924(34.622)	0	0	0.04319	0.91830

especially where the points are sparse. Note that the results by Poisson method shown in this paper are generated with the program in meshlab by Kazhdan and Hoppe (2013).

All the Poisson method, MPU method, Liu and Wang's method can not interpolate the scattered points. And the results by MPU method, Liu and Wang's method and our method have almost all the details of the original model, as is shown in Figs. 19(c), 19(d) and 19(e). From the color maps of error in Figs. 19(h), 19(g) and 19(j), we can find that the result by our method has smaller shape approximation error than other results. And a detailed error table can be seen in Table 7.

7. Conclusions

In this paper, based on the radial power Eq. (7) and Wendland's CSRBF Eq. (8), we present a new CSRBF that unites approximation and interpolation. We can determine whether to interpolate or approximate at each point by using the shape parameter a_i. Moreover, we develop the quasi-interpolation scheme from a simple approximation method into a method combining interpolation and approximation. And we define a practical indicator and present an adaptive algorithm to determine interpolation and approximation adaptively at each point. Furthermore, in our multi-level scheme, with $V_M = V$ and $a^M = 0$ in the final level M, we can interpolate all the given scattered points while we have the shape approximation error by our method no larger than Liu and Wang's method. With experiments results, we find that our method is robust for noise. And we can get better implicit reconstruction results compared with Liu and Wang's method, while almost the same time consume as Liu and Wang's method is needed.

CRediT authorship contribution statement

Yajun Zeng: Conceptualization, Data curation, Investigation, Methodology, Software, Writing – original draft. **Yuanpeng Zhu:** Conceptualization, Methodology, Project administration, Resources, Supervision, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors acknowledge the associate editor and the referees for their insightful comments and valuable suggestions. The research is supported by the National Natural Science Foundation of China (No. 61802129), the Natural Science Foundation Guangdong Province, China (No. 2018A030310381), and the National Training Program of Innovation and Entrepreneurship for Undergraduates (No. 202010561124).

References

Bloomenthal, J., 1988. Polygonization of implicit surfaces. Comput. Aided Geom. Des. 5 (4), 341–355.
Bracco, C., Giannelli, C., Sestini, A., 2017. Adaptive scattered data fitting by extension of local approximations to hierarchical splines. Comput. Aided Geom. Des. 52–53, 90–105.
Carr, J.C., Fright, W.R., Beatson, R.K., 1997. Surface interpolation with radial basis functions for medical imaging. IEEE Trans. Med. Imaging 16 (1), 96–107.
Carr, J.C., Beatson, R., Cherrie, J., Mitchell, T., Fright, W., McCallum, B., Evans, T., 2001. Reconstruction and representation of 3D objects with radial basis function. In: SIGGRAPH2001: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 67–76.
Costarelli, D., Vinti, G., 2019. An inverse result of approximation by sampling Kantorovich series. Proc. Edinb. Math. Soc. 62 (1), 265–280.
Crivellaro, A., Perotto, S., Zonca, S., 2017. Reconstruction of 3D scattered data via radial basis functions by efficient and robust techniques. Appl. Numer. Math. 113, 93–108.
Dou, W.Y., Zhang, L.L., Chen, G., Zhu, W.J., 2019. A combined radial basis function based interpolation method for fluid-structure interaction problems and its application on high-speed trains. Adv. Eng. Softw. 131, 143–152.
Hu, L., Li, Q.S., Liu, S.J., Liu, X.R., Wang, Z., 2018. Wavelet-based polygon soup consolidation. Comput. Graph. 70, 39–50.
Kazhdan, M., Hoppe, H., 2013. Screened Poisson surface reconstruction. ACM Trans. Graph. 32 (3), 1–13.
Kojekine, N., Hagiwara, I., Savchenko, V., 2003. Software tools using CSRBFs for processing scattered data. Comput. Graph. 27 (2), 309–317.
Levin, D., 2003. Mesh-independent surface interpolation. In: Geometric Modeling for Scientific Visualization, pp. 37–49.
Li, P., Huang, Y.B., Li, H.Y., Wang, K.F., Xia, N., Yang, H.T., 2019. Efficient modelling and optimization for double wishbone suspensions based on a non-adaptive sampling sparse response surface. Eng. Optim. 51 (2), 286–300.
Li, P., Chen, J.H., Li, H.Y., Huang, Y.B., Yang, S.Q., Hu, S.X., 2020. An efficient graphic processing unit parallel optimal point searching approach on complex product response surface. Adv. Eng. Softw., 102893.
Liu, S.J., Wang, C.C.L., 2012. Quasi-interpolation for surface reconstruction from scattered data with radial basis function. Comput. Aided Geom. Des. 29 (7), 435–447.
Liu, S.J., Chan, K., Wang, C.C.L., 2012. Iterative consolidation of unorganized point clouds. IEEE Comput. Graph. Appl. 32, 70–83.
Liu, S.J., Wang, C.C.L., 2015. CSRBF-based quasi-interpolation for accurate and fast data fitting. In: 14th International Conference on Computer-Aided Design and Computer Graphics (CAD/Graphics), pp. 65–72.
Liu, S.J., Wang, C.C.L., Brunnett, G., Wang, J., 2016. A closed-form formulation of HRBF-based surface reconstruction by approximate solution. Comput. Aided Des. 78, 147–157.
Nira, D., 1989. Interpolation and approximation by radial and related functions. Academic Press, pp. 211–234.
Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., Seidel, H., 2003. Multi-level partition of unity implicits. ACM Trans. Graph. 22 (3), 463–470.
Ohtake, Y., Belyaev, A., Seidel, H., 2005. 3D scattered data interpolation and approximation with multilevel compactly supported RBFs. Graph. Models 67 (3), 150–165.
Savchenko, V.V., Pasko, A.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.
Shepard, D., 1968. A two-dimensional interpolation function for irregularly-spaced data. In: Proceedings of the 1968 ACM National Conference, pp. 517–524.
Silva, S., Madeira, J., Santos, B., 2005. Polymeco-a polygonal mesh comparison tool. In: International Conference on Information Visualisation DBLP, pp. 842–847.
Song, H., Lv, J., Wang, Y.M., 2015a. Rapid 3D face modeling from video. In: Anonymous (Ed.). Springer International Publishing, Cham, pp. 201–211.
Song, H., Lv, J., Liu, H., Zhao, Q.J., 2015b. A face replacement system based on 3D face model. In: Anonymous (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 237–246.
Wendland, H., 1995. Piecewise polynomial. Adv. Comput. Math., 389–396.