

# Searching for the optimum value of the smoothing parameter for a radial basis function surface with feature area by using the bootstrap method

Khang Jie Liew<sup>1</sup> · Ahmad Ramli<sup>1</sup> ·  
Ahmad Abd. Majid<sup>1</sup>

Received: 22 March 2015 / Revised: 24 November 2015 / Accepted: 1 March 2016 /  
Published online: 18 March 2016  
© SBMAC - Sociedade Brasileira de Matemática Aplicada e Computacional 2016

**Abstract** This paper examines the bootstrap test error estimation of radial basis functions, specifically thin plate spline fitting in surface reconstruction. In the presence of noisy data, instead of an interpolation scheme, the approximation scheme for a thin plate spline is used; therefore, an appropriate value for the smoothing parameter is needed to control the quality of fitting using a set of data points. To find a better smoothing parameter, bootstrap-based test error estimation (the bootstrap leave-one-out error estimator) is applied in searching for the smoothing parameter for a point set model with features. Experimental results demonstrate that the proposed bootstrap leave-one-out error is able to yield the optimum value for the smoothing parameter, which will produce good data approximation and visually pleasing results.

**Keywords** Radial basis function · Bootstrap test error estimation · Smoothing Parameter · Surface reconstruction

## 1 Introduction

Surface reconstruction is the process of constructing a three-dimensional (3D) model surface representation from a set of 3D data points. It reproduces a surface based on the point set obtained from the 3D scanner. Surface reconstruction is used in applications in computer graphics and geometric modelling. The 3D data points are usually obtained from 3D

---

Communicated by Antonio José Silva Neto.

---

✉ Khang Jie Liew  
kenji\_liewkj@yahoo.com.my

Ahmad Ramli  
alaramli@usm.my

Ahmad Abd. Majid  
aamajid38@gmail.com

<sup>1</sup> School of Mathematical Sciences, Universiti Sains Malaysia, 11800 George Town, Penang, Malaysia

scanners in the form of point clouds. The scanned 3D data points can be organised, scattered, or noisy, and the coordinates  $x_i$ ,  $y_i$  and  $z_i$  are used to represent the 3D location of the points. The accuracy of the reconstruction is the main focus because the reconstructed surface may not resemble the original shape of the object if the methods are not chosen appropriately. Fong and Zhang (2013) mention that the commonly used fitting methods include interpolation by spline, interpolation by radial basis function, and the least squares approximation. The scanned data points themselves always contain certain levels of noise. Therefore, handling noisy data points must be considered during the process of reconstruction, as noisy data points can affect the accuracy of the fitting and hence explain the resulting error.

In the context of applied mathematics, error can be defined as the difference between the actual value and the approximation of that computed value. The possible fitting errors are those due to an overfit, underfit or best fit. During the modelling process, in the presence of noisy data, one can encounter the problem of overfitting. This problem arises because during the minimisation of the distance between the input data points and the approximated data points using the model, the noise in the modelled data points is included in the computation. The problem of overfitting can be identified easily using the training error. The training error is the average distance between the input training data and the model predictions. Minimising the training error in the presence of noise leads to a low quality of the model. Interpolating the training data using the interpolation method results in zero training error. The training error is expected to decrease monotonically as the model complexity increases. Most early works do not address noise in the data during the surface reconstruction process. The noise is handled during data pre-processing or model post-processing. However, some of the field's pioneers solved this problem by directly handling the noise during the surface reconstruction. Hoppe et al. (1992) describe and demonstrate a robust and computationally efficient algorithm to reconstruct the surface from an unorganised point set on or near an unknown manifold by taking into account the noisy data. Amenta et al. (1998) propose a Voronoi-based algorithm for the surface reconstruction based on unorganised 3D sample points using an approximation approach in the presence of noise. Carr et al. (2001) propose radial basis functions to reconstruct the surface from point-cloud data and repair incomplete meshes. They use the interpolation scheme for radial basis functions when the data are smooth and the approximation scheme for noisy data. Carr et al. (2003) further discuss approaches to handle noise in radial basis function fitting, especially with large amounts of scattered data.

Surface reconstruction based on local polynomial surfaces, such as moving least squares for the near best approximations to functions of arbitrary dimensions, has been discussed by Levin (1998). Alexa et al. (2003) apply the moving least squares method to approximate and model the surface from a set of points. Another approach based on moving least squares is proposed by Guennebaud and Gross (2007) to fit algebraic surfaces. These methods based on the moving least squares approach do not deal with noise directly.

To deal with noise in data, a statistical approach is implemented during the surface reconstruction. Qian et al. (2006) propose a statistical Bayesian model to reconstruct 3D mesh models from a set of unorganised, noisy data points. Their experimental results show that their method can be applied to smooth the noisy data, remove the outliers, enhance the features, and reconstruct the mesh. A similar approach is used by Jenke et al. (2006) to reconstruct piecewise-smooth surfaces with sharp features and compare the performance of the algorithm using synthetic and real-world data sets. These two Bayesian methods require the user to input the noise level. Yoon et al. (2009) developed a variational Bayesian method

to estimate the level of noise in a scanned point set, where the noise is represented using a normal distribution with zero mean and the variance determines the amount of noise. Ramli and Ivrišsimtzis (2009) propose the bootstrap method to estimate the test error of the model in the context of surface reconstruction using noisy point sets, which can directly be used to compare models.

In this paper, we apply the bootstrap-based method to estimate the test error of a thin plate spline, which is a kind of radial basis function fitting of locally parametrised 3D point sets with features. Efron (1979) introduced a model averaging method, called bootstrap method, in 1979 to estimate the sampling distribution. Using the bootstrap procedure, multiple data sets are created from the same sample that was used previously. Each new data set consists of  $N$  points sampled randomly from the data sample of size  $N$  with replacement. It is a kind of error that can be estimated by repeatedly creating subsamples of local neighbourhoods, fitting models on them, and using the data outside the subsample to measure the error. The type of bootstrap error that is considered is the bootstrap leave-one-out error. There is not much literature on using the bootstrap method to determine the approximation parameter for the approximation scheme of thin plate spline fitting for point set models with features, and hence, we investigate this problem here. A mathematical background on radial basis functions, the bootstrap method, and the types of error estimators will be introduced briefly. The methodology for finding an acceptable approximation parameter for thin plate spline using bootstrap error estimation will be described in Sect. 2. Section 3 will present the related numerical results of error estimators for thin plate spline fitting with different approximation parameters, as well as the graphical results that validate the obtained error estimators. In Sect. 4, a discussion will be presented based on the results of Sect. 3, and finally, a conclusion will be provided in Sect. 5.

## 2 Materials and methods

### 2.1 Radial basis function

The use of radial basis functions (RBFs) for surface representation dates back to the Hardy paper (1971). In 1975, another RBF, the thin plate spline based on the minimum bending energy theory of the thin plate surface, was proposed by Duchon (1977), and it is an example of the global basis function method. Duchon mentions that the RBF is invariant to translations and rotations of the coordinate systems over  $\mathbb{R}^n$ . In addition, it provides a solution for the scattered data interpolation problem in multi-dimensional spaces in the form of a polynomial. Generally, the interpolation problem can be described for a given set of distinct data points (also known as nodes)  $X = \{x_i\}_{i=1}^N \subset \mathbb{R}^n$  and a set of function values  $\{f_i\}_{i=1}^N \subset \mathbb{R}$ , and an interpolant  $s : \mathbb{R}^n \rightarrow \mathbb{R}$  is found, such that

$$s(x_i) = f_i, \quad i = 1, \dots, N \quad (1)$$

The concept of a global method can be described as the interpolant being dependent on all the data points, where any addition or deletion of a data point or a change in one of the coordinates of a data point will propagate throughout the domain of definition (Franke 1982). RBF is widely used in mesh repair, surface reconstruction applications such as range scanning, geographic surveys and medical data, 2D and 3D field visualisations, image warping, morphing, registration, and artificial intelligence (Qin 2014). Given a set of data points,

RBF can be used for interpolation and extrapolation. The property of extrapolation is very useful in mesh repair because it can be used to fill holes. In addition to the interpolating scheme, RBF has a non-interpolating approximation scheme when the data are noisy, which can make the surface smooth. The general RBF has the following form (Carr et al 2001):

$$s(X) = p(X) + \sum_{j=1}^N \lambda_j \phi(|X - X_j|), \tag{2}$$

where  $p$  is a polynomial of a low degree,  $N$  is the total number of distinct data points,  $\lambda_i$  is the weight of the centre  $X_i$ ,  $r = |\cdot| \geq 0$  is the Euclidean norm on  $n$ -dimensional  $\mathbb{R}^n$ , and the basis function  $\phi$  is a real-valued function on the interval  $[0, \infty)$ , which is usually unbounded and has non-compact support. There are two types of support, namely, non-compact (infinite) and compact (finite). A non-compact support means there is no specific interval and it can go to  $\infty$ , whereas a compact support means that the function value is zero outside of a certain interval.

If we consider the RBF with two variables, then  $s(x, y)$  satisfies the interpolation conditions  $s(x, y) = f_i$ , where  $i = 1, 2, \dots, N$ , and it also minimises the measure of smoothness of the function or bending energy, such that (Bennink et al. 2007):

$$\|s\|^2 = \int_{\mathbb{R}} \int_{\mathbb{R}} \frac{\partial^2 s(x, y)}{\partial x^2} + 2 \frac{\partial^2 s(x, y)}{\partial x \partial y} + \frac{\partial^2 s(x, y)}{\partial y^2} dx dy \tag{3}$$

$\|s\|^2$  is known as a semi-norm. Functions with a small semi-norm are smoother than those with a large semi-norm.

The number of basis function equals the number of data points,  $N$ , whereas the number of polynomials,  $m$ , depends on the precision of polynomial,  $v$ , required by the user. For functions of two variables,  $m = \frac{(v+1)(v+2)}{2}$  generally (Hickernell and Hon 1998). For example, if a function of two variables is being used and  $v = 1$  (linear polynomial),  $m = 3$  and the polynomial  $p$  will be  $p(x, y) = a_1 + a_2x + a_3y$ , where  $a_1, a_2, a_3 \in \mathbb{R}$ .

Some choices for RBF include polyharmonics and the Gaussian  $\phi(r) = e^{-cr^2}$ , multi-quadratic  $\phi(r) = \sqrt{r^2 + c^2}$ , and inverse multiquadratic  $\phi(r) = \frac{1}{\sqrt{r^2 + c^2}}$  splines. The value  $c$ , which is available for the Gaussian, multiquadratic and inverse multiquadratic splines, is a user-defined value. A good choice of polyharmonic for fitting a smooth function of two variables is the thin-plate spline  $\phi(r) = r^2 \log(r)$ , which has  $C^1$  continuity, whereas for fitting a smooth function of three variables, the biharmonic spline  $\phi(r) = r$  and triharmonic spline  $\phi(r) = r^3$  can be used. The biharmonic and triharmonic splines have  $C^1$  and  $C^2$  continuity, respectively. The Gaussian spline is mainly used for neural networks, whereas the multi-quadratic spline is used for fitting topographical data. The experiment results use the Gaussian spline, and piecewise polynomials as compact support for fitting surfaces using point clouds will introduce surface artefacts because of the lack of extrapolation across the holes (Carr et al 2001).

Suppose we want to interpolate the data of two variables and set the polynomial  $p$  as the linear form; then, the interpolant is  $s : \mathbb{R}^2 \rightarrow \mathbb{R}$ , such that

$$s(x_i, y_i) = a_1 + a_2x_i + a_3y_i + \sum_{j=1}^N \lambda_j \phi \left( \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \right) = f_i$$

Because there are  $N + 3$  unknowns, three additional solution constraints are added, such that

$$\sum_{j=1}^N \lambda_j = \sum_{j=1}^N \lambda_j x_j = \sum_{j=1}^N \lambda_j y_j = 0$$

which yields the linear system written in an  $(N + 3) \times (N + 3)$  matrix form as follows:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 0 & x_1 & x_2 & \cdots & x_N \\ 0 & 0 & 0 & y_1 & y_2 & \cdots & y_N \\ 1 & x_1 & y_1 & \phi(r_{1,1}) & \phi(r_{1,2}) & \cdots & \phi(r_{1,N}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ 1 & x_N & y_N & \phi(r_{N,1}) & \phi(r_{N,2}) & \cdots & \phi(r_{N,N}) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \lambda_1 \\ \vdots \\ \lambda_N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ f_1 \\ \vdots \\ f_N \end{bmatrix}$$

where  $\phi(r_{i,j}) = \phi(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2})$

The above matrix form can be further simplified as

$$\begin{bmatrix} 0 & P^T \\ P & A \end{bmatrix} \begin{bmatrix} a \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ f \end{bmatrix} \tag{4}$$

where  $P$  is the matrix with  $i$ th row  $(1, x_i, y_i)$ ,  $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_N)^T$ ,  $a = (a_1, a_2, a_3)^T$ , and  $f = (f_1, f_2, \dots, f_N)^T$ . By solving the linear system, the value of  $\lambda$  and  $a$  can be uniquely determined, and hence, the function  $s(x, y)$  is derived. It is appropriate to use the direct method to solve the above matrix for a problem with at most a few thousand points, that is,  $N < 2000$ . If  $N \gtrsim 2000$ , then the matrix will be poorly conditioned, and the solution will be unreliable. However, this problem is resolved using the fast method proposed by Carr et al. (2001); therefore, fitting and evaluating a large number of data points with a single RBF is now possible.

If noise is present in the data points, the interpolation condition of Eq. (1) is strict. Therefore, the condition is to be relaxed to favour smoothness as measured using Eq. (3). Let us consider the problem given the nodes  $\{x_i\}_{i=1}^N \subset \mathbb{R}^n$  and minimize

$$\rho \|s\|^2 + \frac{1}{N} \sum_{i=1}^N (s(x_i) - f_i)^2,$$

where  $\rho \geq 0$ . This problem is known as spline smoothing, and the parameter  $\rho$  controls the quality of approximation; in other words, it controls the trade-off between the smoothness and fidelity of the data (Wahba 1990). The solution for this problem is also a RBF of the form given in Eq. (2). A smaller value of  $\rho$  will provide a better approximation and will be an exact interpolation if  $\rho$  tends to 0. By modifying the Eq. (4), we have the following:

$$\begin{bmatrix} 0 & P^T \\ P & A + \rho I \end{bmatrix} \begin{bmatrix} a \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ f \end{bmatrix} \tag{5}$$

where  $I$  is an identity matrix. By solving the system of linear equations in (5), we get  $a$  and  $\lambda$  and then plug in the Eq. (2). A RBF in the form of an approximation is obtained.

## 2.2 Bootstrap method

The bootstrap method is based on repetition-based random resampling of the data and averaging of the results obtained from each sample. The reuse of the data as a result of the repetitive resampling is helpful when the available data are sparse. The following descriptions regarding the creation of bootstrap sets are based on Ramli and Ivrisstizis (2009).

Suppose a set of sample data consists of  $N$  data points or training points, such that  $V = \{v_1, v_2, \dots, v_N\}$ . Each element in set  $V$ ,  $v_i$ , is in 3D coordinate form, such that  $v_i = (x_i, y_i, z_i)$ . The bootstrap set is produced by randomly sampling the  $N$  elements from  $V$  with replacement. Therefore, the probability of picking a new element from  $V$  every time for the bootstrap sample will always be  $\frac{1}{N}$ . Because we use random resampling with replacement, the bootstrap sample contains elements that are different from  $V$  because some same elements will be picked for the second, third or  $n$ th time. In addition, there is the possibility that some elements are not selected and, therefore, will not appear in the bootstrap sample. Additionally, the expected number of distinct elements in bootstrap set  $V^{*b}$  is lower than  $V$ . The average number of distinct observations in each bootstrap set is about  $0.632 N$  (Hastie et al. 2001). The sampling procedure is repeated  $B$  times to produce  $B$  independent bootstrap sets,  $V^{*b}$ , where  $b = 1, 2, 3, \dots, B$ . There is a tradeoff between the accuracy and computational cost in fixing the user-input value  $B$ . A larger value of  $B$  implies that more bootstrap sets are created, and hence, the reliability of the estimation is increased.

## 2.3 Error estimations

### 2.3.1 Training error

Let us consider the 3D setting. Given a set of samples  $V$  of size  $N$  with data points  $(x_i, y_i, z_i) \in \mathbb{R}^3$  for  $i = 1, 2, 3, \dots, N$ , a function  $f(x_i, y_i) = z_i$  is estimated to fit the data. The training error of the model is the average loss over the sample. The training error is given as

$$e\bar{r}r = \frac{1}{N} \sum_{i=1}^N |f(x_i, y_i) - z_i|, \quad (6)$$

where function  $f$  is the model fitted to the whole data set  $V$ .

### 2.3.2 Bootstrap leave-one-out error

Given a set of samples  $V = \{v_1, v_2, \dots, v_N\}$  of size  $N$  with data points  $(x_i, y_i, z_i) \in \mathbb{R}^3$  for  $i = 1, 2, 3, \dots, N$ , the bootstrap method discussed in Sect. 2.3 is applied  $B$  times to produce  $B$  independent bootstrap sets,  $V^{*b}$ , where  $b = 1, 2, 3, \dots, B$ . If we consider the size of the bootstrap set to be same as the size of the sample  $V$ , then repetition of the same elements in the bootstrap set is possible. In order to fit a model with function  $f^{*b}$  from a bootstrap set  $V^{*b}$ , we need to unite all the elements inside the bootstrap set prior to the fitting. Thus, the

repeated elements of the bootstrap set can only be used once instead of multiple times. This explains why the size of the bootstrap set is unfixed, as mentioned in Sect. 2.3. The formula for the bootstrap leave-one-out error is

$$E\hat{r}^{(1)} = \frac{1}{N} \sum_{i=1}^N \frac{1}{n_{C_i}} \sum_{b \in C_i} |f^{*b}(x_i, y_i) - z_i|, \tag{7}$$

where  $C_i$  is the index set of the bootstrap sets  $V^{*b}$  that do not contain the point  $v_i$  such that  $C_i = \{b | v_i \notin V^{*b}\}$ , and  $n_{C_i}$  denotes the size of the set. Note that the bootstrap leave-one-out error mimics the cross-validation process, which can be used to limit the problem of overfitting. In leave-one-out cross-validation, which is known as  $N$ -fold cross-validation,  $N$  is the size of  $V$ . For each training, only one  $v_i$  is left out from  $V$ , the remaining elements of  $V$  are used for training, and the average error is obtained. The bootstrap leave-one-out error is similar to the leave-one-out error in concept. Leave-one-out cross-validation has a lower bias but higher variance, as the training sets are very similar to the data set, whereas bootstrap leave-one-out cross-validation has a higher bias but lower variance as the average number of distinct observations in each bootstrap set is about  $0.632 N$ , which means that the training sets are less similar to the data set (Efron and Tibshirani 1997).

### 2.3.3 Test error estimation and determination of the smoothing parameter $\rho$

In order to determine an appropriate parameter  $\rho$  for the approximation scheme of the thin plate spline for the feature point set model, we test the Stanford bunny point set model, which contains 11,146 data points. However, only 200 data points are selected randomly from the 11,146 data points. The randomly selected 200 data points are fixed for the whole study. For a particular point, we use the  $k$ -nearest neighbours algorithm and obtain a neighbourhood of size  $N = 50$ , which means that 50 data points are selected from the 200 data points. We then carry out the bootstrap procedure 50 times to produce 50 independent bootstrap sets,  $V^{*b}$ , where  $b = 1, 2, 3, \dots, 50$ . We now describe the algorithm as follows:

In our study, we use the point set Stanford bunny model with four different noise levels: 0.00, 0.25, 0.50, and 1.00. In order to determine an appropriate smoothing parameter  $\rho$  for the thin plate spline for a particular noise level, we first compute the training error and bootstrap leave-one-out error for the selected 200 data points using Algorithm 1. A list of different smoothing parameter values are chosen for testing. We know that a smaller value of  $\rho$  may provide a better approximation, but we do not know how small it can be. In the presence of noisy data, interpolating all the points will result in overfitting. To initiate the numerical computation, the  $h$  value will act as an indicator to predict a suitable range for the smoothing parameter  $\rho$ . Note that the variable  $h$  is the average distance between the two nearest points in a set of points. With the range of the smoothing parameter  $\rho$ , we can calculate the two types of test error estimators as mentioned earlier. For the following section, a plot of the training error against parameter  $\rho$  and a plot of the bootstrap leave-one-out error against parameter  $\rho$  will be shown. An appropriate parameter  $\rho$  will be chosen based on the plots. The graphical results will be used to validate the obtained results.

**Algorithm 1:****(A) Training error estimation,****(B) Bootstrap leave-one-out error****Input :** Selected data points  $V = \{v_1, v_2, v_3, \dots, v_N\}$ **Output:**(A) Average training error for the selected  $N$  data points(B) Average bootstrap leave-one-out error for the selected  $N$  data points**Set**  $N = 200$ ;**Set**  $k = 50$ ;**Set**  $B = 50$ ;**For**  $i = 1 : N$ Step 1.1: Using the  $k$ -nearest neighbours algorithm, select the  $k$ -nearest data points for  $v_i$ .

Step 1.2: Use the principal component analysis (PCA) method as discussed in Ramli (2012) to estimate normal vector.

Step 1.3: Reorient the position of the selected  $k$ -nearest data points to the ordinary XYZ axis position using the information from the normal vector from Step 1.2. To calculate the training error, proceed to (A); else, proceed to (B) for the bootstrap error calculation.**(A)**Step 1.4a: Fit the  $k$  data points using a function  $f$  and then calculate the training error  $e\bar{r}_i$  from equation (6).**End For**Step 2a: Calculate the average training error for the selected  $N$  data points by  $\frac{1}{N} \sum_{i=1}^N e\bar{r}_i$ .**(B)****For**  $b = 1 : B$ Step 1.1.1b: Create bootstrap sets  $V^{*b}$  of size  $k$ , and unite the elements in  $V^{*b}$ , which means the removal of duplicates in  $V^{*b}$ .**End For**Step 1.4b: Determine the index set of bootstrap sets,  $C_i$ , and its size  $n_{C_i}$ .Step 1.5b: Calculate the bootstrap leave-one-out error  $E\hat{r}_i^{(1)}$  from Eq. (7).**End For**Step 2b: Calculate the average bootstrap leave-one-out error for the selected  $N$  data points such that  $\frac{1}{N} \sum_{i=1}^N E\hat{r}_i^{(1)}$ .

### 3 Results

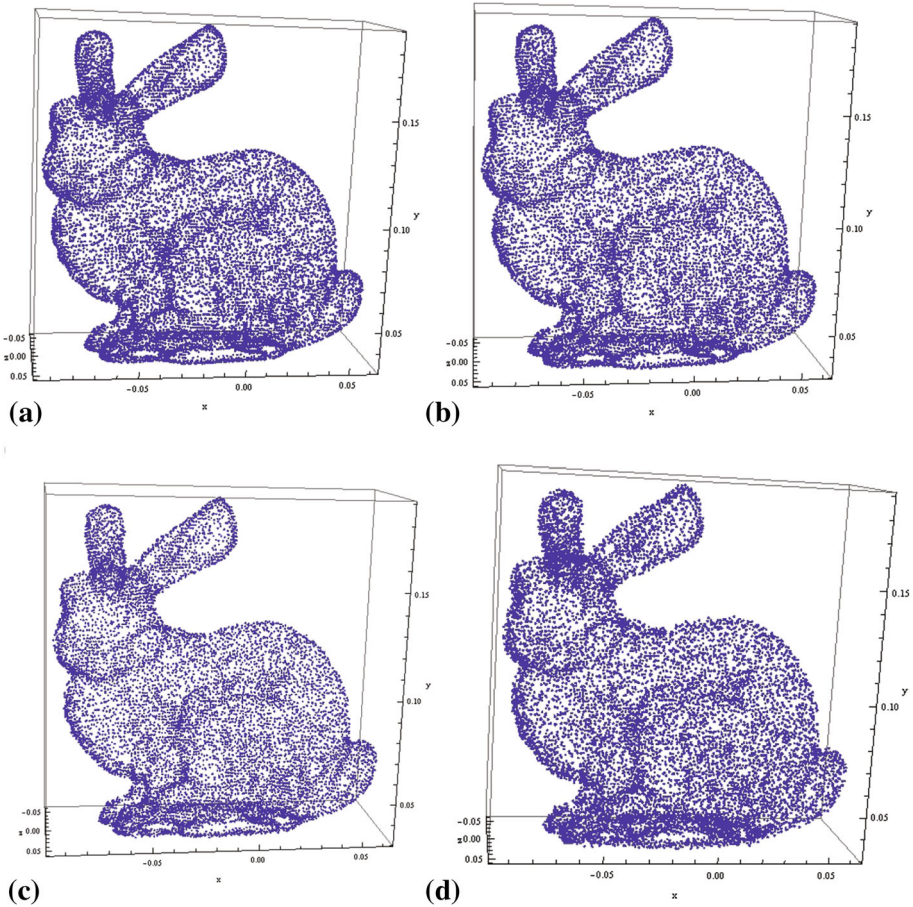
The following results were obtained by testing 200 data points from the Stanford bunny model, with noise levels 0.00, 0.25, 0.50, and 1.00. The point set model that corresponds to the chosen noise levels is shown in Fig. 1.

Before presenting the plots of the training error and bootstrap leave-one-out error against parameter  $\rho$  for the different noise levels, the numerical values of the training error and bootstrap leave-one-out error for different values of  $\rho$  at different noise levels will be tabulated. The  $h$  value will serve as an indicator of the range of  $\rho$  that should be tested. Using the  $h$  values, we can list the choices of  $\rho$  values that fall before and after the  $h$  value. The step size of the  $\rho$  values is a user-defined value. For example, if the  $h$  value is 0.1, then the list of tested  $\rho$  values will be 0, 0.005, 0.01, 0.05,  $\dots$ , 0.5. Higher values of  $\rho$  are tested to observe the implications for both types of error.

We present the plots of the training error and bootstrap leave-one-out error against the value of  $\rho$  for the different noise levels in Fig. 2.

We validate the obtained result in Fig. 2 using the graphical method with a neighbourhood of size 50 with the same centre. The green colour points denote the data points. We choose





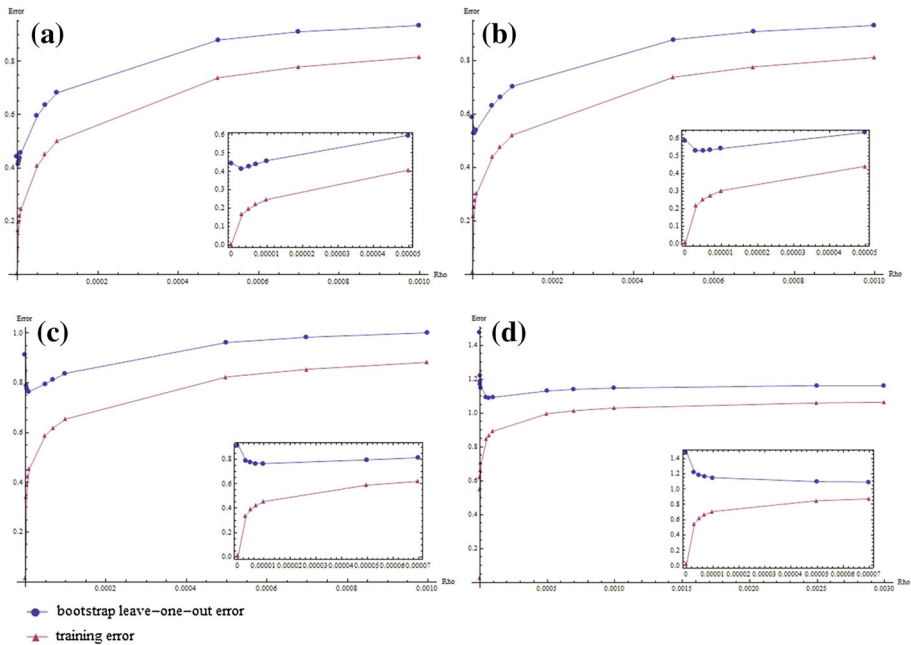
**Fig. 1** The Stanford bunny point set **a–d** that corresponds to the noise levels 0.00, 0.25, 0.50, and 1.00, respectively

a better  $\rho$  value by considering the minimum value of the bootstrap leave-one-out error. We present the results for two different surfaces, which are at data point index 13 and index 213.

### 4 Discussion

The Stanford bunny point set is selected for this study as it contains sharp features. This property is essential for us to have a clearer idea of the fitting using error estimations. From the context of computer graphics, a feature implies a model with a crease, corner, dart, or cusp. To address the computational cost in calculating the average training error and bootstrap leave-one-out error, only 200 data points are selected randomly from the total 11,146 data points available in our point set model. The whole study is conducted using the same 200 data points. Our approach manages to reduce the computational time involved.

From Table 1, the  $h$  value for different noise levels is calculated, and we notice that the values are very small. A small  $h$  value indicates that the data points in the Stanford bunny



**Fig. 2** The plot of the training error and bootstrap leave-one-out error against the value of the parameter  $\rho$  for thin plate spline fitting: **a–d** correspond to the noise levels 0.00, 0.25, 0.50, and 1.00, respectively. The zoom of the area where the minimum of the bootstrap leave-one-out error appears is added inside for each sub-figure

**Table 1** The  $h$ -value that corresponds to the different noise levels for the Stanford bunny point set

Noise level	$h$ value
0.00	0.0015
0.25	0.0016
0.50	0.0017
1.00	0.0019

model are very close together. In addition to indicating the average distance between the points, these  $h$  values are used to determine the values of the parameter  $\rho$ .

Figure 2 shows the plot of the training error and bootstrap leave-one-out error against the parameter  $\rho$  for thin plate spline fitting along the same axis for different noise levels. The  $\rho$  value starts from 0, which means that we start the initial observation from interpolation and follow it by the approximation of the sample of data points. Tables 2, 3, 4 and 5 show the training error and bootstrap leave-one-out error for the different parameter  $\rho$  values at different noise levels. The bold values are the recommended value for smoothing parameter  $\rho$ . For validation purposes, two sets of sample data points with size 50 are selected using the  $k$ -nearest neighbours (kNN) algorithm, as we want to present the surface fitting for different areas of the bunny model for a particular value of the parameter  $\rho$ . The  $k$ -nearest neighbours algorithm searches for the points that are relatively close to a considered point from a set of points in  $n$ -dimensional space.  $k$  is a user-defined value, and one has to choose it carefully.

**Table 2** Training error and bootstrap leave-one-out error for the different values of parameter  $\rho$  at noise level 0.00

$\rho$	Training error	Bootstrap leave-one-out error
0	0.0058	0.4437
<b>0.000003</b>	0.1666	0.4155
0.000005	0.1978	0.4262
0.000007	0.2206	0.4394
0.00001	0.2471	0.4584
0.00005	0.4078	0.5966
0.00007	0.4515	0.6362
0.0001	0.5018	0.6826
0.0005	0.7386	0.8800
0.0007	0.7793	0.9110
0.001	0.8160	0.9346
0.0025	0.8806	0.9766
0.003	0.8890	0.9831
0.005	0.9067	0.9944

**Table 3** Training error and bootstrap leave-one-out error for different values of the parameter  $\rho$  at noise level 0.25

$\rho$	Training error	Bootstrap leave-one-out error
0	0.0093	0.5881
0.000003	0.2168	0.5296
<b>0.000005</b>	0.2525	0.5285
0.000007	0.2763	0.5338
0.00001	0.3022	0.5418
0.00005	0.4409	0.6335
0.00007	0.4781	0.6648
0.0001	0.5219	0.7046
0.0005	0.7382	0.8788
0.0007	0.7768	0.9088
0.001	0.8120	0.9321
0.0025	0.8746	0.9720
0.003	0.8828	0.9763
0.005	0.9004	0.9878

If the  $k$  value is too small, noise will be included in the model, whereas for large  $k$  values, the neighbours may include many points from other classes.

At noise level 0.00, where the data points are smooth without the presence of noisy data, the training error and bootstrap leave-one-out error increase gradually with an increase in parameter  $\rho$ , as shown in Fig. 2a. This indicates that a smaller value of  $\rho$  improves the thin plate spline fitting. It is recommended that a value of parameter  $\rho$  approximately close to 0 is the best choice for noise-free data points. A larger value of the parameter  $\rho$  leads to higher error estimations. This can be validated using the graphical method, as shown in Figs. 3b and 4b, where the shapes look flat.

**Table 4** Training error and bootstrap leave-one-out error for different values of parameter  $\rho$  at noise level 0.50

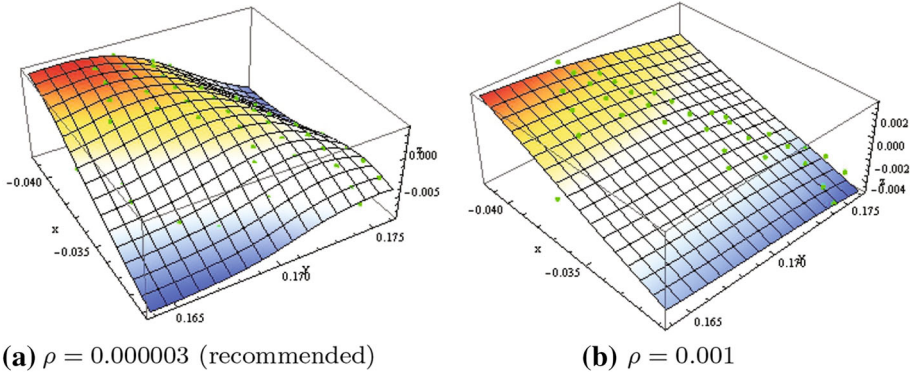
$\rho$	Training error	Bootstrap leave-one-out error
0	0.0168	0.9151
0.000003	0.3403	0.7885
0.000005	0.3913	0.7767
0.000007	0.4232	0.7686
<b>0.00001</b>	0.4554	0.7657
0.00005	0.5895	0.7967
0.00007	0.6199	0.8137
0.0001	0.6546	0.8382
0.0005	0.8237	0.9622
0.0007	0.8544	0.9833
0.001	0.8825	1.0014
0.0025	0.9327	1.0297
0.003	0.9394	1.0325
0.005	0.9534	1.0430

**Table 5** Training error and bootstrap leave-one-out error for different values of parameter  $\rho$  at noise level 1.00

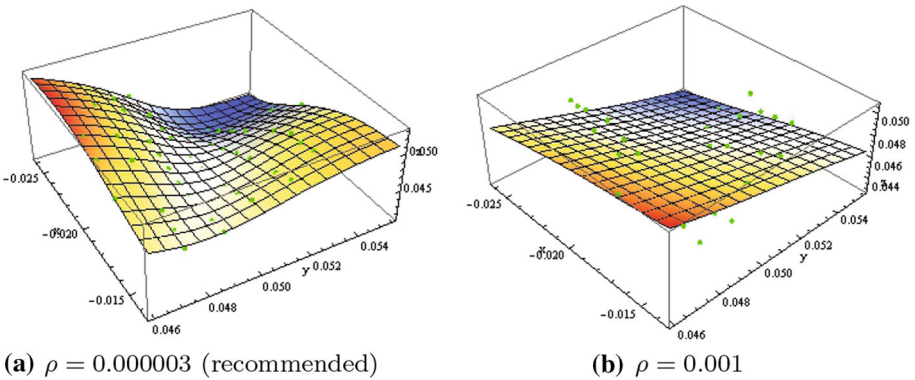
$\rho$	Training error	Bootstrap leave-one-out error
0	0.0258	1.4779
0.000003	0.5476	1.2226
0.000005	0.6205	1.1876
0.000007	0.6639	1.1690
0.00001	0.7058	1.1486
0.00005	0.8470	1.0952
<b>0.00007</b>	0.8702	1.0910
0.0001	0.8936	1.0929
0.0005	0.9957	1.1307
0.0007	1.0135	1.1400
0.001	1.0299	1.1474
0.0025	1.0596	1.1620
0.003	1.0636	1.1623
0.005	1.0720	1.1670

At noise level 0.25, the training error increases gradually as it is no longer closely fitted, but the bootstrap leave-one-out error initially decreases a bit and then increases after a certain  $\rho$  value, as shown in Fig. 2b. The  $\rho$  value that is approximately near the turning point of the plot of the bootstrap leave-one-out error is selected as the recommended parameter value as it yields the minimum bootstrap-leave-one out error. The corresponding validation is shown in Figs. 5 and 6.

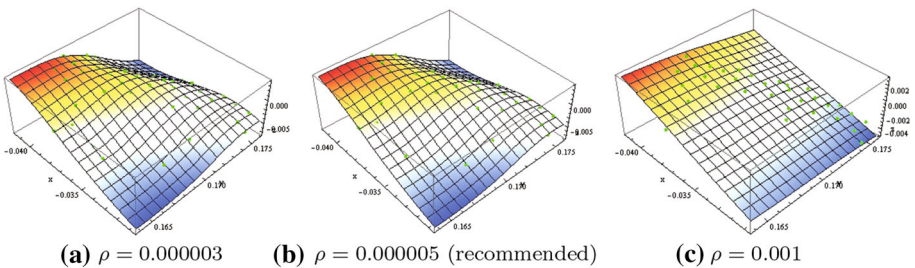
At noise levels 0.50 and 1.00, the plots of the training error and bootstrap leave-one-out error against the parameter  $\rho$  demonstrate a shape similar to that in Fig. 2, also seen in Fig. 2c, d. The graphical validations for noise level 0.50 and 1.00 are shown in Figs. 7, 8, 9, and 10. They show that the parameter  $\rho$  value approximately near 0 is not a good choice for fitting



**Fig. 3** Comparison of the surface fitted using the thin plate spline at noise level 0.00 for the 50-nearest neighbourhood at the centre of data point index 13



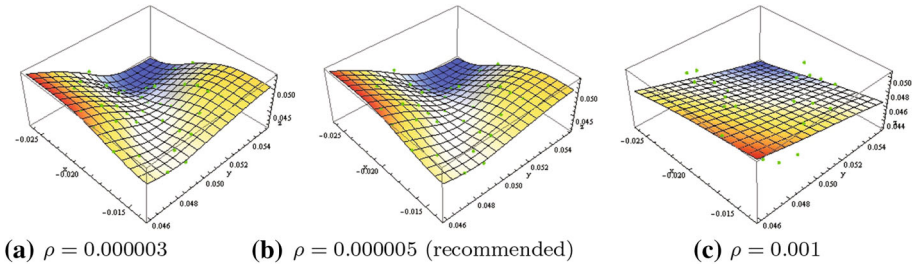
**Fig. 4** Comparison of the surface fitted using the thin plate spline at noise level 0.00 for the 50-nearest neighbourhood at the centre of data point index 213



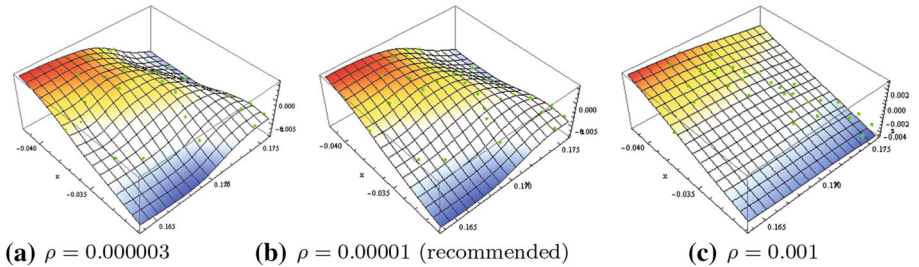
**Fig. 5** Comparison of the surface fitted using the thin plate spline at noise level 0.25 for the 50-nearest neighbourhood at the centre of data point index 13

when the noise level is increasing. With a suitable choice of  $\rho$ , the approximated thin plate spline surface can still maintain the required shape at a higher level of noise.

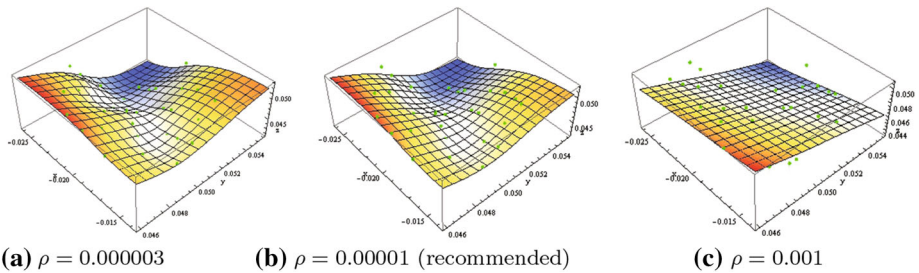
The training error increases gradually and seems to stabilise with small oscillations at the tail of the plot, whereas the bootstrap leave-one-out error decreases gradually and also



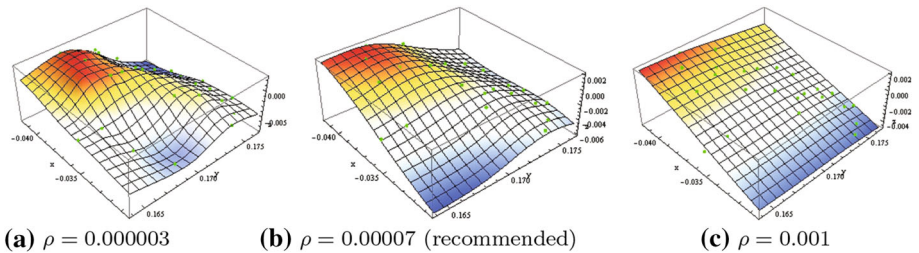
**Fig. 6** Comparison of the surface fitted using the thin plate spline at noise level 0.25 for the 50-nearest neighbourhood at the centre of data point index 213



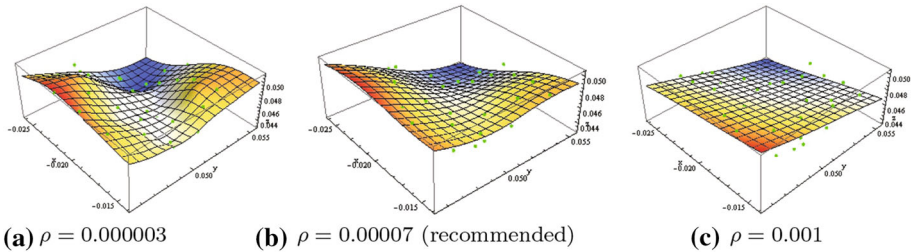
**Fig. 7** Comparison of the surface fitted using the thin plate spline at noise level 0.50 for the 50-nearest neighbourhood at the centre of data point index 13



**Fig. 8** Comparison of the surface fitted using the thin plate spline at noise level 0.50 for the 50-nearest neighbourhood at the centre of data point index 213



**Fig. 9** Comparison of the surface fitted using the thin plate spline at noise level 1.00 for the 50-nearest neighbourhood at the centre of data point index 13



**Fig. 10** Comparison of the surface fitted using the thin plate spline at noise level 1.00 for the 50-nearest neighbourhood at the centre of data point index 213

seems to be constant at the tail of the plot. We can observe that the training error would not provide good estimations for the parameter  $\rho$ , as the error would simply become zero when the function overfits at  $\rho = 0$ . However, bootstrap error estimation is better for finding the optimum value of the parameter.

## 5 Conclusion

This paper shows that the test error estimation method can be used to estimate the parameter  $\rho$  and, therefore, provide a reliable surface approximation from a set of data points with features and different levels of noise. Experimental results show that the bootstrap leave-one-out error calculated from a sample of data points can be applied to search for a better parameter  $\rho$  compared with the training error.

**Acknowledgements** The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. The authors also would like to extend their gratitude to the financial support from USM short term Grant no. 304/PMATHS/6312136 and FRGS Grant no. 203/PMATHS/6711433. The first author also gratefully acknowledges the generous financial support by the Ministry of Education Malaysia under the MyPhD scholarship.

## References

- Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva CT (2003) Computing and rendering point set surfaces. *IEEE Trans Visual Comput Graph* 9(1):3–15
- Amenta N, Bern M, Kamvysselis M (1998) A new voronoi-based surface reconstruction algorithm. In: *SIGGRAPH '98 Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, pp 415–421
- Bennink H, Korbeek J, Janssen B, Romeny BMtH (2007) Warping a neuro-anatomy atlas on 3D MRI data with radial basis functions. In: *Proceedings of the 3rd Kuala Lumpur International Conference on Biomedical Engineering (Biomed) 2006*. New York, NY, USA: Springer-Verlag, Berlin Heidelberg, pp 28–32
- Carr JC, Beatson RK, Cherrie JB, Mitchell TJ, Fright WR, McCallum BC, Evans TR (2001) Reconstruction and representation of 3D objects with radial basis functions. In: *SIGGRAPH '01 Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pp 67–76
- Carr JC, Beatson RK, McCallum BC, Fright WR, McLennan TJ, Mitchell TJ (2003) Smooth surface reconstruction from noisy range data. In: *GRAPHITE '03 Proceedings of the 1st International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, pp 119–216
- Duchon J (1977) Constructive theory of functions of several variables. *Lecture Notes in Mathematics*, Ch. Splines Minimizing Rotation-invariant Semi-norms in Sobolev Spaces. Springer-Verlag, Berlin, pp 85–100
- Efron B (1979) Bootstrap methods: Another look at the jackknife. *Ann Stat* 7(1):126

- Efron B, Tibshirani R (1997) Improvements on cross-validation: the 632+ bootstrap method. *J Am Stat Assoc* 92(438):548–560
- Fong R, Zhang Y (2013) Piecewise bivariate hermite interpolations for large sets of scattered data. *J Appl Math* 10
- Franke R (1982) Scattered data interpolation. *Math Comput* 38:181–200
- Guennebaud G, Gross M (2007) Algebraic point set surfaces. In: *SIGGRAPH '07 Proceedings of the 34th Annual Conference on Computer Graphics and Interactive Techniques*, pp 10
- Guvenir HA, Akkus A (1997) Weighted k nearest neighbor classification on feature projections. In: *Proceedings of the 12-th International Symposium on Computer and Information Sciences*
- Han J, Kamber M (2001) *Data mining: concepts and techniques*. Ch. Classification and Prediction. Elsevier, pp 314–315
- Hardy RL (1971) Multiquadric equations of topography and other irregular surfaces. *J Geophys Res* 76(8):1905–1915
- Hastie T, Tibshirani R, Friedman J (2001) *The elements of statistical learning: data mining, inference, and prediction*. Springer
- Hickernell FJ, Hon YC (1998) Radial basis function approximation of the surface wind field from scattered data. *Int J Appl Sci Comput* 4(3):221–247
- Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W (1992) Surface reconstruction from unorganized points. In: *SIGGRAPH '92 Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*, pp 71–78
- Jenke P, Bokeloh M, Schilling A, Straer W, Wand M (2006) Bayesian point cloud reconstruction. *EURO-GRAPHICS* 25(3):10
- Levin D (1998) The approximation power of moving least-squares. *Math Comput* 67(224):1517–1531
- Qian G, Tong R, Peng W, Dong J (2006) Bayesian mesh reconstruction from noisy point data. *Lect Notes Comput Sci Adv Artifi Real Tele-Existence* 4282:819–829
- Qin H (2014) *Lecture notes on computer graphics (theory, algorithms, and applications)*. Department of Computer Sciences, Stony Brook University
- Ramli A, Ivrişsimtzis I (2009) Bootstrap test error estimations of polynomial fittings in surface reconstruction. In: *Proceedings of the Vision, Modeling, and Visualization Workshop 2009*, pp 101–109
- Ramli A (2012) *Bootstrap based surface reconstruction*. PhD thesis, Durham University, United Kingdom
- Wahba G (1990) Spline models for observational data. Ch. Estimating the Smoothing Parameter. *Society for Industrial and Applied Mathematics, SIAM*, pp 314–315
- Yoon MC, Ivrişsimtzis I, Lee SY (2009) Variational Bayesian noise estimation of point sets. *Comput Graph* 33(3):226–234