

Article

Least Squares Compactly Supported Radial Basis Function for Digital Terrain Model Interpolation from Airborne Lidar Point Clouds

Chuanfa Chen ^{1,2,*} , Yanyan Li ², Na Zhao ³, Bin Guo ²  and Naixia Mou ²

¹ State Key Laboratory of Mining Disaster Prevention and Control Co-founded by Shandong Province and the Ministry of Science and Technology, Shandong University of Science and Technology, Qingdao 266590, China

² Shandong Provincial Key Laboratory of Geomatics and Digital Technology of Shandong Province, Shandong University of Science and Technology, Qingdao 266590, China; yylee@whu.edu.cn (Y.L.); guobin07@mails.ucas.ac.cn (B.G.); mounaixia@163.com (N.M.)

³ State Key Laboratory of Resources and Environment Information System, Institute of Geographical Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing 100101, China; Zhaon@lreis.ac.cn

* Correspondence: chencf@lreis.ac.cn

Received: 10 March 2018; Accepted: 8 April 2018; Published: 10 April 2018



Abstract: To overcome the huge volume problem of light detection and ranging (LiDAR) data for the derivation of digital terrain models (DTMs), a least squares compactly supported radial basis function (CSRBF) interpolation method is proposed in this paper. The proposed method has a limited support radius and fewer RBF centers than the sample points, selected by a newly developed surface variation-based algorithm. Those make the linear system of the proposed method not only much sparser but also efficiently solvable. Tests on a synthetic dataset demonstrate that the proposed method is comparable to the smoothing RBF, and far superior to the exact RBF. Moreover, the first is much faster than the others. The proposed method with the RBF centers selected by the surface variation-based algorithm obviously outperforms that with the random selection of equal number. Real-world examples on one private and ten public datasets show that the surfaces of simple interpolation methods including inverse distance weighting, natural neighbor, linear and bicubic suffer from the problems of roughness, peak-cutting, discontinuity and subtle terrain feature loss, respectively. By contrast, the proposed method produces visually appealing results, keeping a good tradeoff between noise removal and terrain feature preservation. Additionally, the new method compares favorably with ordinary kriging (OK) for the generation of high-resolution DTMs in terms of interpolation accuracy, yet the former is much more robust to spatial resolution variation and terrain characteristics than the latter. More importantly, our method is about 4 times faster than OK. In conclusion, the proposed method has high potential for the interpolation of a large LiDAR dataset, especially when both interpolation accuracy and computational cost are taken into account.

Keywords: interpolation; LiDAR; DTM; noise

1. Introduction

At present, remote-sensing techniques such as photogrammetry, interferometry synthetic aperture radar (InSAR) and light detection and ranging (LiDAR) have become the mainstream tools for collecting elevation dataset over large areas [1–4]. Generally, the remote-sensing-derived datasets have a high spatial resolution less than 1 m for airborne LiDAR data [5,6]. Therefore, remotely sensed large-volume datasets pose great challenges for most current interpolation methods [7], especially when the required digital terrain model (DTM) resolution is smaller than the average distance between points of the original point cloud.

Classical interpolation methods commonly used to generate DTMs from a LiDAR dataset include inverse distance weighting (IDW), natural neighbor (NN), ordinary kriging (OK), linear interpolation and radial basis function (RBF) [1,8–11]. Among the interpolators, RBF is considered as the most promising method for scattered data approximation and interpolation with respect to ease of use, interpolation accuracy and visual quality [12–15]. Nevertheless, the matrix in the linear system of the classical RBF is dense due to its global support. This makes RBF interpolation computationally impossible for large datasets, since solving a dense linear system by direct methods requires $O(n^3)$ operations and $O(n^2)$ memory usage with n representing the number of sample points.

To alleviate the computational burden, various schemes have been proposed, which can be categorized into three main groups, namely, fast solution of the linear system of RBF equations, domain decomposition and compactly supported radial basis functions (CSRBFs) [16]. Here, we mainly focus on CSRBFs and propose a least squares (LS) CSRBF approximation method to produce DTMs from LiDAR dataset. Compared to the classical CSRBFs, the main contributions of the proposed method include: (1) RBF centers are fewer than the sample points and mainly located on local terrain features, which could improve computational efficiency and decrease memory cost; (2) an unconstrained LS method is developed to efficiently solve the objective function of the proposed method and (3) the proposed method is insensitive to noise due to the LS nature. Note that the LS method is highly sensitive to outliers. Here, we assume that the LiDAR ground points are free of outliers, which can be achieved by a careful manual editing in practice [17,18].

The objectives of this paper are as follows: (1) to propose a LS CSRBF interpolation method for the derivation of DTMs from airborne LiDAR point clouds, where all off ground points were completely filtered out; (2) to develop a surface variation-based algorithm to select RBF centers mainly located on terrain features, which could improve the interpolation accuracy of the proposed method; and (3) to assess the performance of the proposed method with those of some well-known interpolation methods on simulated and real-world datasets.

The remainder of this paper is organized as follows. Relevant papers are reviewed in Section 2. In Section 3, the principle of the proposed method is given. Section 4 employs experiments on synthetic and real-world LiDAR datasets to compare the performance of the proposed method and some well-known interpolation methods. Discussion and conclusions are shown in Section 5.

2. Literature Review

The commonly used methods to decrease the computational burden of RBF interpolation or approximation can be generally classified into three groups. The first group is to use fast methods for fitting and evaluating RBFs. For example, Beatson et al. [19] presented a combination of a suitable approximate cardinal function preconditioner, the generalized minimal residual (GMRES) iterative method and existing fast matrix-vector algorithms to lower the computational cost of solving an RBF interpolation problem. Faul et al. [20] used a carefully selected set of q points to construct preconditioners for which they were able to interpolate problems with $d = 40$ dimensions and $N = 10,000$ points in a few iterations. Gumerov and Duraiswami [21] employed the Fast Multipole Method (FMM) to accelerate the matrix-vector product in the method of Faul et al. [20], and used a novel FMM algorithm for the evaluation. Crivellaro et al. [22] reconstructed large 3D scattered data via RBFs by an adaptive least-squares multi-level interpolating approach. However, the above methods do not take into account the huge memory cost of RBFs.

The second group is to apply the function locally based on the fact that the regression function is insensitive to distant points. For example, Hardy [23] and Beatson et al. [24] presented the domain decomposition method. The idea behind this method is to divide the considered domain into a number of subdomains and then try to solve the original problem as a series of sub-problems that interact through the interfaces. Similarly, Wendland [25] combined the theory of RBF with a partition of unity method to solve large-scale, scattered data problems. The idea is to solve a large number of small, local problems instead of one large-scale problem and to put the local solutions together using a partition of

unity method. Deng and Tang [26] developed a moving surface spline interpolation method, where only the nearest neighbor points were chosen for interpolation. Yet, this group of methods still requires the aforementioned fast methods to evaluate and fit RBFs if a high speed is to be reached.

The last group is to adopt CSRBFs [16]. They lead to a sparse interpolation matrix, which can greatly decrease memory and computational costs [27]. However, the classical CSRBFs still use all data points as RBF centers, and are sensitive to noise. Generally, when the density of sample points is high, not all data need to be used as centers to define a representative surface [28–30]. Thus, Franke et al. [31] described a greedy algorithm for RBF center selection from the original sample points. Ohtake et al. [32] developed an adaptive RBF fitting procedure with the RBF centers randomly chosen from the points with the randomness controlled by the point density and surface geometry. Süßmuth et al. [33] optimized the positions and the weights of the RBF centers by minimizing a nonlinear error function. Skala [34] first defined a “virtual mesh” in the study domain and then took the corners of this mesh as the RBF centers. It is well known that a good placement of the RBF centers can improve the approximation of the underlying data, especially for DTM construction [15]. However, the above methods do not consider the local characteristics of RBF centers.

3. Principle of the Proposed Method

In this section, the formulation of the proposed method is first given. Then, a new algorithm based on surface variation is developed to select RBF centers. Finally, the sparseness of the interpolation matrix based on Wendland’s CSRBF is analyzed and a so-called coordinate format is introduced to store the sparse matrix.

3.1. Formulation

Suppose that LiDAR-derived sample points are modeled as

$$z_i = f(\mathbf{x}_i) + e_i, i = 1, 2, \dots, N \quad (1)$$

where $\mathbf{x}_i \in R^2$ and z_i respectively represent the horizontal coordinate and elevation of the i th sample point, N is the total number of sample points, e is sample noise and $f(\mathbf{x})$ is the interpolation function to be estimated.

Similar to RBFs, $f(\mathbf{x})$ by the proposed method can be expressed as

$$f(\mathbf{x}) = \sum_{i=1}^n q_i(\|\mathbf{x} - \mathbf{c}_i\|)\alpha_i + \sum_{j=1}^m p_j(\mathbf{x})\beta_j \quad (2)$$

where q_i is the i th basis function with the center $\mathbf{c}_i$, n is the number of centers, $\|\cdot\|$ is the Euclidean norm, α is the weight of the basis function, $p(\cdot)$ and β are the polynomial basis and coefficient, and $m - 1$ is the degree of the polynomial. It is obvious that the addition of polynomials of total degree at most $m - 1$ guarantees polynomial precision, i.e., if the data come from a polynomial of total degree less than or equal to $m - 1$, they are fitted by that polynomial [27]. Namely, with the polynomial, polynomial precision can be obtained. In the proposed method, RBF centers are fewer than the sample points, i.e., $n < N$.

To perform interpolations using Equation (2), the coefficients α and β must be pre-obtained by solving the following objective function:

$$\begin{cases} \min_{\alpha, \beta} \left\| \begin{matrix} \mathbf{Q} & \mathbf{P} \\ N \times n & m \times 1 \end{matrix} \begin{matrix} \alpha \\ \beta \end{matrix} - \begin{matrix} \mathbf{z} \\ N \times 1 \end{matrix} \right\|_2^2 \\ \text{s.t.} \quad \begin{matrix} \hat{\mathbf{P}}^T & \alpha \\ m \times n & n \times 1 \end{matrix} = \begin{matrix} \mathbf{0} \\ m \times 1 \end{matrix} \end{cases} \quad (3)$$

where $(\mathbf{Q})_{ij} = q(\|x_i - c_j\|)$, $(\mathbf{P})_{ij} = p_j(x_i)$, $(\hat{\mathbf{P}})_{ij} = p_j(c_i)$, $\alpha = [\alpha_1 \ \cdots \ \alpha_n]^T$, $\beta = [\beta_1 \ \cdots \ \beta_m]^T$ and $\mathbf{z} = [z_1 \ \cdots \ z_N]^T$.

Equation (3) can be generally solved using the well-known method of Lagrange multipliers. However, in terms of weighting for equality constrained LS problems [35], Equation (3) can be transformed into an unconstrained problem with the form:

$$\min_{\alpha, \beta} \|\mathbf{B}\mathbf{y} - \mathbf{l}\|_2^2 \quad (4)$$

where $\mathbf{B} = \begin{bmatrix} \mathbf{Q} & \mathbf{P} \\ \lambda \hat{\mathbf{P}}^T & \mathbf{0} \end{bmatrix}$; $\mathbf{y} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$; $\mathbf{l} = \begin{bmatrix} \mathbf{z} \\ \mathbf{0} \end{bmatrix}$; λ is a suitably large number.

The rationale for choosing λ is that if λ is large enough, the residual $\hat{\mathbf{P}}^T \alpha$ will be so small that the constraint is effectively satisfied. Bjorck [35] proved that as λ increases, the solution of an unconstrained LS problem approaches the corresponding constrained LS solution. Due to the LS nature, the proposed method is robust to noise.

Letting the derivatives of Equation (4) with respect to α and β be zero, we can obtain the following system:

$$(\mathbf{B}^T \mathbf{B})\mathbf{y} = \mathbf{B}^T \mathbf{l} \quad (5)$$

Thus, to estimate α and β , solving Equation (5) is more efficient than solving Equation (3).

3.2. Selection of RBF Centers

Since the RBF centers are fewer than the sample points in the proposed algorithm, the key point is how to select RBF centers from the whole sample points. In the context of DTM construction, Hardy [36] indicated that the RBF centers located on significant terrain features could improve interpolation accuracy. Thus, in the proposed method, we try to select local terrain feature points as the RBF centers as many as possible.

Pauly et al. [37] showed that surface variation can quantitatively describe the variation along the surface normal, i.e., estimate how much the points deviate from the tangent plane. Thus, the surface variation is used to detect terrain feature points in our study. The procedure of using surface variation to select RBF centers is as follows:

- (i) For a point $\mathbf{p} = (p_x, p_y, p_z)$ in the given LiDAR data, find its k nearest neighbors $\mathbf{p}_i$ ($i = 1, 2, \dots, k$) using the kd-tree algorithm.
- (ii) Define the covariance matrix $\mathbf{C}$ at the point $\mathbf{p}$ as $\mathbf{C} = \begin{bmatrix} \mathbf{p}_1 - \bar{\mathbf{p}} \\ \vdots \\ \mathbf{p}_k - \bar{\mathbf{p}} \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 - \bar{\mathbf{p}} \\ \vdots \\ \mathbf{p}_k - \bar{\mathbf{p}} \end{bmatrix}^T$, where $\bar{\mathbf{p}} = \frac{1}{k} \sum_{i=1}^k \mathbf{p}_i$.
- (iii) Compute the surface variation as $\sigma(\mathbf{p}) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}$, where $\lambda_0 < \lambda_1 < \lambda_2$ are the three eigenvalues of $\mathbf{C}$.
- (iv) Repeat (i)–(iii) until the surface variations of all points are estimated.
- (v) Cover the study domain using grids with a pre-defined resolution h , which is determined by the required number of RBF centers J . Namely, $h = \sqrt{W \times L / J}$, where W and L respectively represent the width and length of the study domain.
- (vi) Find all points located in the same grid and select the point with the largest surface variation as the potential RBF center. This assures the uniform distribution of RBF centers and improves the probability of the selected RBF center as the significant terrain feature point.

As shown in Figure 1, the RBF centers selected by the surface variation-based algorithm uniformly cover the study area and are mainly located on the terrain features.

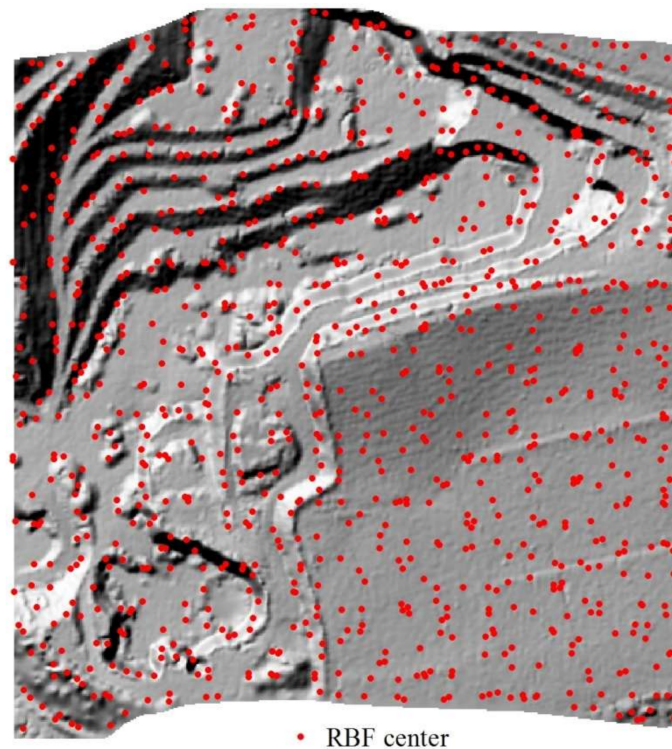


Figure 1. Distribution of RBF centers selected by the surface variation-based algorithm.

3.3. Sparseness of the Interpolation Matrix

In our method, Wendland's CSRBFs are adopted (Table 1), as they are the most commonly used "local" radial basis functions [27].

Here, we define

$$\varphi(r) = (1-r)_+ = \begin{cases} (1-r) & 0 \leq r \leq 1 \\ 0 & r > 1 \end{cases} \quad (6)$$

Equation (6) indicates that when $\bar{r} > R$, the corresponding elements of Q in the interpolation matrix B are zeros (Table 1). Thus, if R was properly selected, Q would become a sparse matrix and have very few nonzero elements, especially for the CSRBFs with fewer RBF centers than the sample points.

Table 1. Wendland's CSRBFs $q_{3,k}(r)$ and $\hat{q}_{3,k}(r) = q_{3,k}(1-r)$ for different degrees of smoothness $2k$ and three variates, where $r = \bar{r}/R$, R is the support radius and $\bar{r}$ is Euclidean distance between the sample point and the RBF center.

k	$q_{3,k}(r)$	$\hat{q}_{3,k}(r)$	Smoothness
0	$(1-r)_+^2$	r_+^2	C^0
1	$(1-r)_+^4(4r+1)$	$r_+^4(5-4r)$	C^2
2	$(1-r)_+^6(35r^2+18r+3)$	$r_+^6(56-88r+35r^2)$	C^4
3	$(1-r)_+^8(32r^3+25r^2+8r+1)$	$r_+^8(66-154r+121r^2-32r^3)$	C^6

Here, we employed the so-called coordinate format to store the sparse matrix [38]. The data structure includes three arrays: (i) a real array containing all the real values of the nonzero elements in each row; (ii) an integer array containing their row indices; and (iii) a second integer array containing their column indices. This data structure greatly saves storage cost. Generally, for an $N \times N$ dense

matrix of the classical RBFs, the storage requirement is $8N^2$ bytes, if we use 8-bytes for real numbers and 4-bytes for integers. Supposing that each center of the classical CSRBFs has approximate k points located in the support radius, the storage requirement based on the coordinate format is $16(N \times k) + 4$ bytes, while for the proposed method with n RBF centers, the storage requirement is $16(n \times k) + 4$ bytes. Thus, the ratio of the memory requirement of the global RBF to that of the classical CSRBF is

$$v_1 = \frac{8N^2}{16(N \cdot k) + 4} \approx \frac{N}{2k}$$

and the ratio of the memory requirement of the global RBF to that of the proposed method is

$$v_2 = \frac{8N^2}{16(n \cdot k) + 4} \approx \frac{N^2}{2(n \cdot k)}$$

Since both n and k are much smaller than N (i.e., $v_2 > v_1$), our method can greatly reduce memory requirements. For example, for interpolating $N = 100,000$ sample points, the memory requirements of the global RBF, the classical CSRBF and the proposed method with $k = 0.1N$ and $n = 0.1N$ are about 74.5 GB, 14.9 GB and 1.5 GB, respectively.

Assume that solving a system of linear equations of the size $N \times N$ requires the time complexity $O(N^3)$. The time complexities of the classical CSRBF and the proposed method are $O(N^3)$ and $O(n^3)$, respectively. Thus, the speed-up of the proposed method to the global RBF or the classical CSRBF is $O\left((N/n)^3\right)$. Therefore, compared to the classical CSRBFs and global RBFs, the proposed method greatly decreases the computational burden with respect to memory and computational costs. The pseudo-code of the proposed approach is shown in Algorithm 1.

Algorithm 1: Proposed method

Input: Training points, number of RBF centers and support radius

Output: a DTM

1. Select the RBF centers with the surface variation-based algorithm.
 2. Obtain α and β by solving the system of linear equations (i.e., Equation (5)).
 3. Cover the study domain using grids with the desired DTM resolution.
 4. Estimate the value of each grid using Equation (2).
-

4. Examples

In this section, experiments on simulated and real-world LiDAR datasets are employed to assess the performance of the proposed algorithm, and its results are compared with some well-known interpolation methods. Note that the implementation of the proposed method was performed in MATLAB (a proprietary product of MathWorks located in Natick, MA, USA) on a PC with the configuration of Intel® Core™ (a product of Intel Corporation located in Santa Clara, CA, USA) i7-4790 CPU @ 3.60 GHz and 16.00 GB RAM. Moreover, the linear system of equations was directly solved with ‘\’ in MATLAB. The versatility of ‘\’ in solving linear systems stems from its ability to take advantage of symmetries in the problem by dispatching to an appropriate solver. This approach aims to minimize computation time. The first distinction the function makes is between full (also called “dense”) and sparse input arrays.

4.1. A Numerical Test

4.1.1. Simulated Dataset

In the numerical test, Gaussian synthetic surface was taken as the test surface (Figure 2), so that the true value can be predetermined to avoid uncertainty caused by uncontrollable data errors. The formulation of the surface is expressed as

$$f(x, y) = 3(1 - x)^2 e^{(-x^2 - (y+1)^2)} - 10 \left(x/5 - x^3 - y^5 \right) e^{(-x^2 - y^2)} - e^{-(x+1)^2 - y^2} / 3$$

where $x \in [-3, 3]$, $y \in [-3, 3]$ and $-6.6 < f(x, y) < 8.1$.

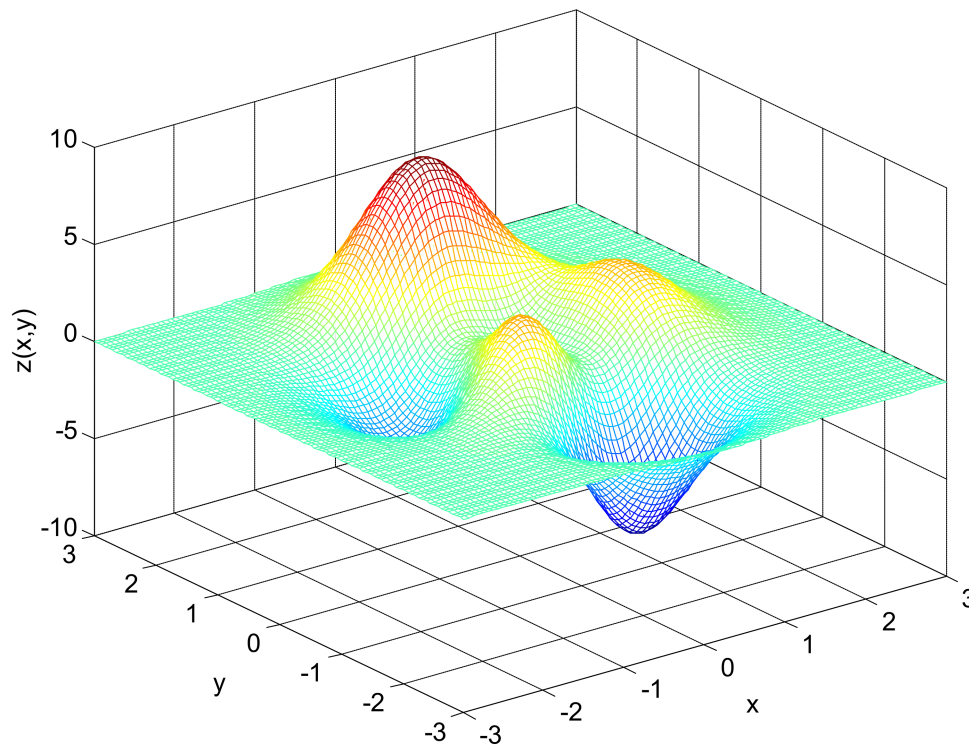


Figure 2. Gaussian synthetic surface.

The computational domain was covered with 100×100 grids with the grid resolution of 0.06. Two thousands points with Halton distribution were randomly selected from the surface to simulate the synthetic surface. Similar to the real-world sample points, the 2000 points were subject to noise, which was drawn from a normal distribution with the mean of zero and standard deviation of σ , i.e., $N(0, \sigma^2)$, where σ is 0.01, 0.02, 0.04, 0.08 or 0.1, respectively.

For the sake of simplicity, the C^0 , C^1 , C^2 or C^3 Wendland's CSRBF (Table 1) is hereafter named as CSRBF0, CSRBF1, CSRBF2 or CSRBF3, respectively. Their simulation results were compared with those of the exact RBF and the smoothing RBF, both of which use a global support. We also assessed the performance of the proposed method with the RBF centers selected by the surface variation-based algorithm and by the random method of equal number (the random method means that the RBF centers are randomly selected). The normal equation of the smoothing RBF is expressed as [17],

$$\begin{bmatrix} \mathbf{Q} + s\mathbf{I} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \mathbf{z} \\ \mathbf{0} \end{bmatrix} \quad (7)$$

where s is the smoothing parameter; $(Q)_{ij} = q(r_{ij})$; $q(r) = e^{-(\rho r)^2}$ is the Gaussian basis function and ρ is a shape parameter. Note that the exact RBF is a special case of the smoothing RBF for $s = 0$.

Root mean square error (RMSE) was used to assess the interpolation accuracy. It is formulated as,

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (f_i - \hat{f}_i)^2}{N}} \quad (8)$$

where f_i and $\hat{f}_i$ respectively represent the i th true and simulated values, and N is the number of evaluation points.

4.1.2. Accuracy Comparison

The optimal parameters of all the interpolation methods, determined by the ten-fold cross validation technique [39], are shown in Table 2. Results indicate that for the proposed method, regardless of noise distribution, CSRBF with a higher degree of smoothness generally requires fewer RBF centers and smaller support radius than that with a lower degree. For example, CSRBF3 has 200 centers and the support radius of 5 for the noise distribution of $N(0,0.01^2)$, while CSRBF0 has 750 centers and the support radius of 6.5. Moreover, with the increase of noise standard deviation, the number of centers decreases. For instance, when the noise standard deviation increases from 0.01 to 0.1, the centers of CSRBF2 decrease from 300 to 150. This demonstrates that the proposed method should adopt fewer centers to resist the effect of noise of a higher level.

Table 2. Optimal parameters of all versions of RBF in the numerical test.

Noise Distribution		CSRBF0	CSRBF1	CSRBF2	CSRBF3		Smoothing	Exact
$N(0,0.01^2)$	N_{ctr}	750	450	300	200	ρ	2	6.5
	R	6.5	4.5	5	5	s	0.01	-
$N(0,0.02^2)$	N_{ctr}	750	400	250	200	ρ	1.5	4.9
	R	6	6.5	5	5	s	0.01	-
$N(0,0.04^2)$	N_{ctr}	750	300	250	150	ρ	1.5	5.6
	R	6.5	4.5	4.5	4	s	0.09	-
$N(0,0.08^2)$	N_{ctr}	550	200	200	150	ρ	1	6.3
	R	6	4.5	5	4	s	0.01	-
$N(0,0.1^2)$	N_{ctr}	450	200	150	150	ρ	1	6.5
	R	6	4.5	5	4	s	0.01	-

Simulation results (Table 3) show that the exact RBF obtains the poorest results for the five groups of points, indicating that it is highly sensitive to sample noise. CSRBF with a higher degree of smoothness always outperforms that with a lower level, mainly due to the high smoothness of the simulated surface. For example, for the noise distribution of $N(0,0.04^2)$, the RMSE of proposed method increases from 0.0317 to 0.0502, as the smoothness degree of CSRBF decreases from 6 to 0. CSRBF3 is always slightly more accurate than the smoothing RBF except for the noise distribution of $N(0,0.1^2)$. On average, CSRBF3 is approximately as accurate as the smoothing RBF. The exact RBF has the worst results, which is mainly due to its high sensitivity to sample noise.

Table 3. RMSEs of all versions of CSRBF in the numerical test, where the RBF centers of the proposed method are selected with the surface variation-based algorithm (Note that bold font represents the best result).

Noise Distribution	CSRBF0	CSRBF1	CSRBF2	CSRBF3	Smoothing RBF	Exact RBF
$N(0,0.01^2)$	0.0248	0.0121	0.0096	0.0087	0.0094	0.0569
$N(0,0.02^2)$	0.0304	0.0210	0.0169	0.0160	0.0165	0.0931
$N(0,0.04^2)$	0.0502	0.0388	0.0335	0.0317	0.0327	0.1508
$N(0,0.08^2)$	0.0841	0.0672	0.0609	0.0514	0.0522	0.2627
$N(0,0.1^2)$	0.1044	0.0815	0.0717	0.0690	0.0678	0.3136
On average	0.0588	0.0441	0.0385	0.0354	0.0357	0.1754

Figure 3 shows that regardless of noise distribution and CSRBF, the proposed method with the RBF centers selected by the surface variation-based algorithm is always more accurate than that by the random method. This further indicates that the distribution of RBF centers plays an important role in the performance of the proposed method.

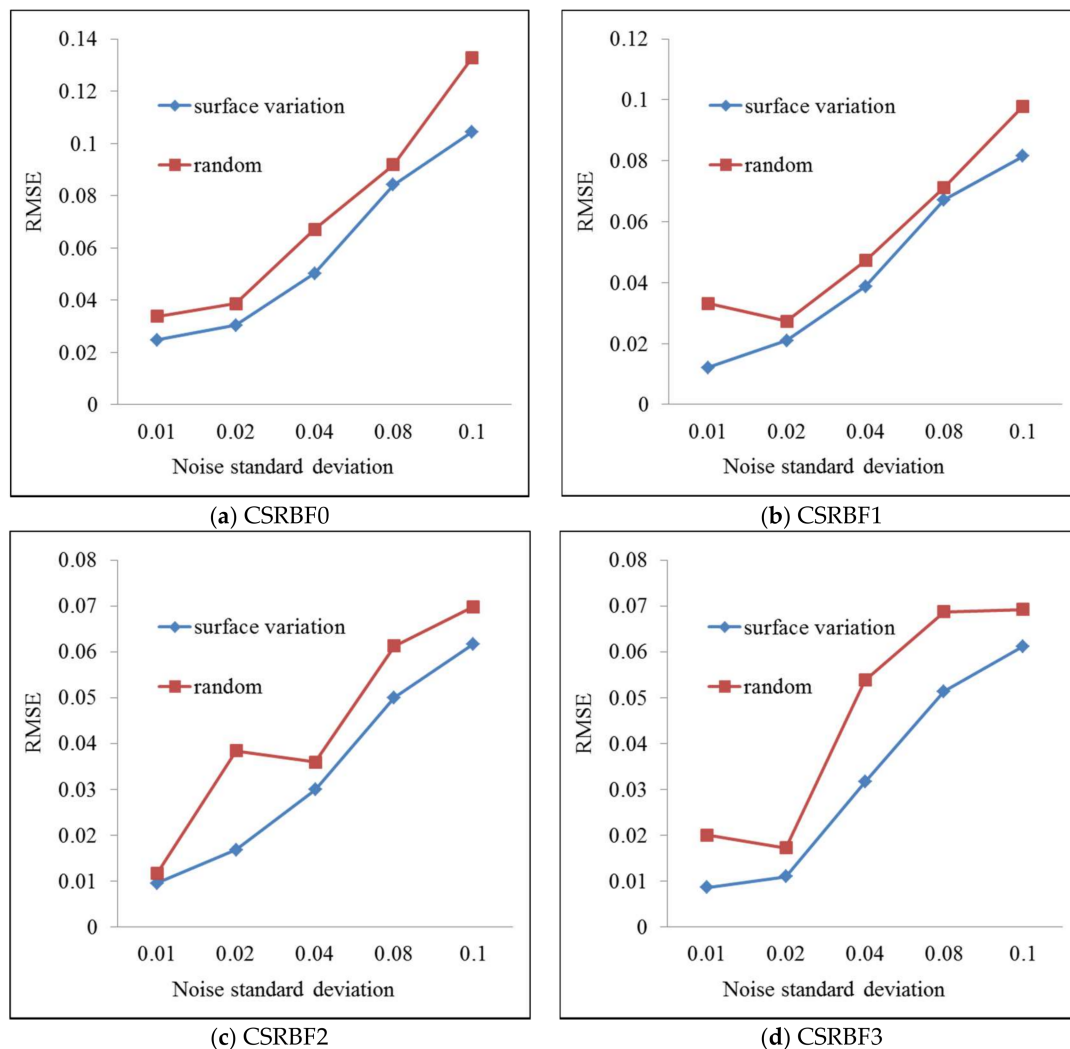


Figure 3. RMSEs of the proposed method with the RBF centers selected by the surface variation-based algorithm and by the random method.

Figure 4 shows that the exact RBF has the coarsest error surface, which indicates it has the worst performance for interpolating noisy sample points. CSRBF3 has a similar appearance to the smoothing RBF, both of which are more accurate than the exact RBF and CSRBF0.

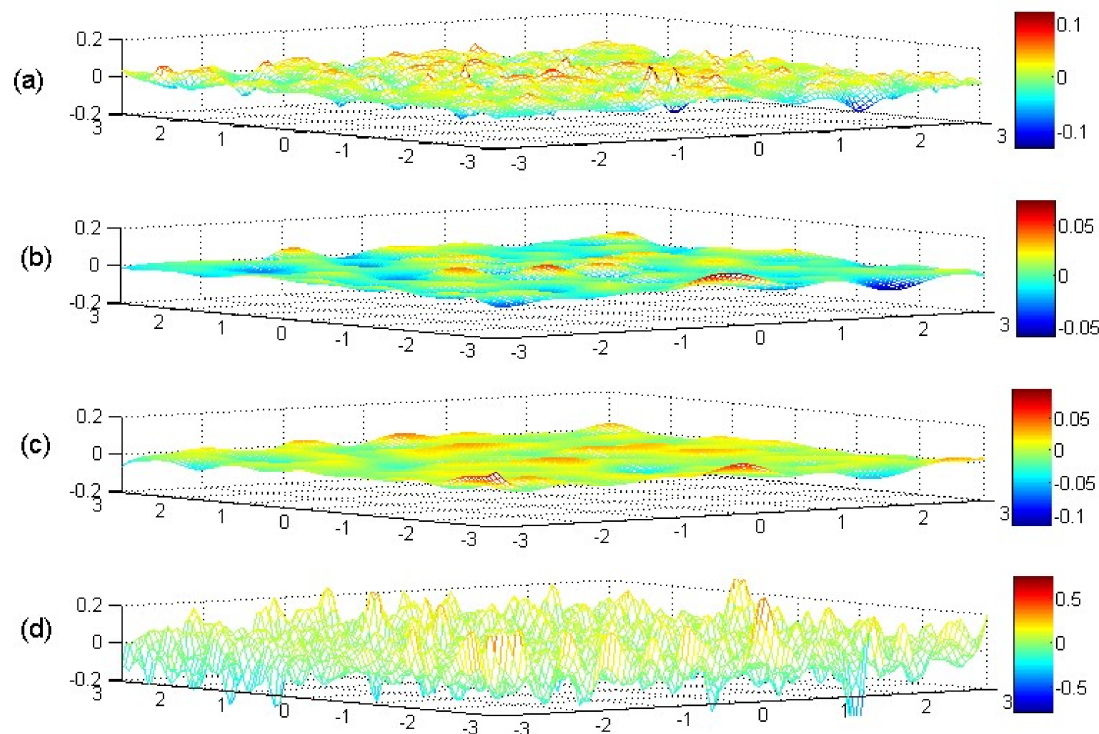


Figure 4. Error surfaces of (a) CSRBF0, (b) CSRBF3, (c) smoothing RBF and (d) exact RBF for the noise distribution of $N(0,0.04^2)$.

Note that the computational requirement of the proposed method includes the costs of RBF center selection using the surface variation-based algorithm and solving the linear system. Figure 5 shows the computational costs of the two steps of CSRBF3. It can be found that compared to the cost of solving the linear system, the cost of RBF center selection can be negligible.

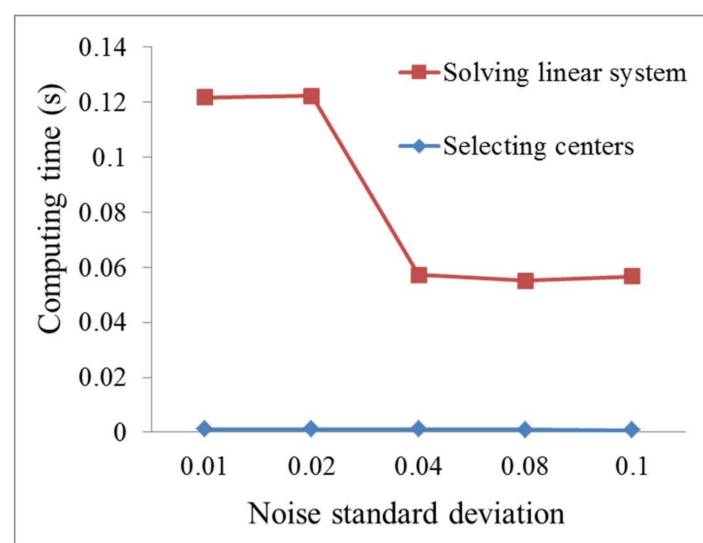


Figure 5. Computing costs of RBF center selection and linear system solution of the proposed method.

Table 4 demonstrates that the computational cost of the smoothing RBF is approximately equal to that of the exact RBF. This result can be expected since the two methods solve a similar linear system (i.e., Equation (7)). The cost of the proposed method is mainly dependent on the number of RBF centers. With the increase of the number of centers, the cost becomes larger. This is because the sparseness level of the interpolation matrix will be reduced when more centers are used. On average, the proposed method is much faster than the smoothing and exact RBFs.

Table 4. Computational costs of all versions of RBF in the numerical test (unit: s).

Noise Distribution	CSRBF0	CSRBF1	CSRBF2	CSRBF3	Smoothing	Exact
$N(0,0.01^2)$	2.0	0.51	0.28	0.14	6.2	6.0
$N(0,0.02^2)$	2.0	0.55	0.18	0.14	6.8	7.1
$N(0,0.04^2)$	1.9	0.24	0.16	0.068	7.3	7.3
$N(0,0.08^2)$	1.1	0.12	0.14	0.064	6.8	6.3
$N(0,0.1^2)$	0.8	0.12	0.089	0.063	7.7	6.4
On average	1.6	0.30	0.17	0.095	7.0	6.6

4.2. Real-World Examples

In this section, the proposed method is respectively assessed on one private and ten publicly-available datasets of airborne lidar surveys. Its results are compared with those of some commonly used interpolators for lidar-derived DTM construction, such as IDW, OK, linear, bicubic and NN interpolation methods.

IDW is a type of deterministic method for multivariate interpolation with a known scattered set of points [40]. To predict a value for any unmeasured location, IDW uses the measured values surrounding the prediction location. The measured values closest to the prediction location have more influence on the predicted value than those farther away [41]. Thus, it gives greater weights to points closest to the prediction location. All points within a specified radius or a specified number of points are used to perform prediction. Here, we used the latter.

Kriging is an advanced geostatistical procedure that generates an estimated surface from a scattered set of points [42]. Kriging is similar to IDW in that it weights the surrounding measured values to derive a prediction for an unmeasured location. In IDW, the weight depends solely on the distance to the prediction location. However, with the kriging method, the weights are based not only on the distance between the measured points and the prediction location, but also on the overall spatial arrangement of the measured points [43]. To use the spatial arrangement in the weights, the spatial autocorrelation must be quantified with semivariogram. Thus, the important step in kriging is to accurately fit the experimental model to a theoretical variogram model. As one type of kriging, OK assumes that the variation in z-values is free of any structural component (drift) [44].

Linear interpolation has been used by the GIS community for many years, which generally uses Triangulated Irregular Network (TIN) to achieve interpolations. Namely, it first triangulates sample points into a temporary TIN with Delaunay triangulation and then raster the TIN to create a DTM with the desired resolution. The main disadvantage of the linear interpolation is that the surfaces are not smooth and may give a jagged appearance [45]. This is caused by discontinuous slopes at the triangle edges and sample data points.

Bicubic interpolation is the lowest order two-dimensional interpolation procedure which maintains the continuity of the function and its first derivatives across cell boundaries [45]. It requires 16 values to derive the coefficients. The 16 values can be obtained from two schemes. The first one uses the elevations at the four vertices of the grid cell, together with three derivatives [46], and the second one uses the elevations of sample points directly, without estimating the partial derivatives [45]. The first scheme is commonly adopted to interpolate a grid-based DTM to higher spatial resolutions, since the three derivatives can be easily computed with the finite difference. However, for interpolating scattered sample points, we can only use the second one due to the difficult computation of the

derivatives. Here, we used the second scheme because of the interpolation of randomly distributed sample points.

NN finds the closest subset of input samples to a query point and applies weights to them based on proportionate areas to interpolate a value [47]. It is also known as Sibson or “area-stealing” interpolation. Its basic properties are that it’s local, using only a subset of samples that surround a query point, and interpolated heights are guaranteed to be within the range of the samples used. It does not infer trends and will not produce peaks, pits, ridges, or valleys that are not already represented by the input samples [48]. The surface passes through the input samples and is smooth everywhere except at locations of the input samples.

In order to give a fair comparison to the proposed method with respect to interpolation accuracy and computing time, all aforementioned interpolation methods were performed in MATLAB, and their parameters were optimized with the ten-fold cross validation technique [39]. For OK, we used the existing ‘kriging’ function to perform interpolation [49], where the calculation of the experimental variogram and the fit of a theoretical variogram to the experimental variogram were respectively conducted with the functions of ‘variogram’ [50] and ‘variogramfit’ [51]. Since variogram fitting is very time-consuming, only five thousand points, randomly selected from the training dataset, were used to fit the variogram. The other interpolation methods can be easily coded in MATLAB.

4.2.1. Private Dataset

The study site with an area of 2 ha and the mean slope of about 20° is located in the valley of Alptal in Switzerland. It is characterized by a strong micro-relief with many ridges and depressions, and covered with trees and vegetation [52]. The airborne LiDAR dataset was collected using a Riegl VZ-1000 mounted on a helicopter with the flight altitude and speed of 500 m above ground and 30 knots, respectively. The average pulse density was 16.9 pulses/m². Since the raw point cloud contains both ground and non-ground points, the lasground function in the software package of LAStools was used to automatically filter the point cloud. After the filtering process, 146,809 ground points with the density of 7.34 points/m² were obtained. Namely, the average point space was 0.37 m. The ground points were randomly separated into training and testing datasets with the proportions of 90% and 10%, which were used to generate DTMs and assess the interpolation accuracy, respectively.

Similar to the numerical test, the proposed method with different CSRBFs were used for DTM construction, and the performances of the proposed method with the RBF centers selected by the surface variation-based algorithm and by the random scheme were compared. Since data density and grid size have significant effects on the quality of DTMs [11,53], 100%, 50%, 30% and 10% of points were randomly extracted from the training dataset, and for each data density, DTMs with five different resolutions were produced by each interpolator, i.e., 0.3, 0.5, 1, 1.5 and 2 m. In total, $4 \times 5 \times 11 = 220$ DTMs were generated. Note that our purpose was not to assess the absolute accuracy of DTMs because the testing points have a similar accuracy level to the training data. Instead, the testing points were employed to evaluate the performance of the interpolators for computing heights at locations without measured points in the training points. The flowchart of accuracy comparison between different interpolation methods is shown in Figure 6.

It can be concluded that with 100% of training data, $146,809 \times 0.9 = 132,128$ points should be used to construct DTMs. Thus, the smoothing and exact RBFs require a memory cost of at least 130.1 GB, while the proposed method requires a cost of 263 MB when 13,128 RBF centers were selected and 1313 sample points were located in the support radius of each center. Due to the large dataset, the exact and smoothing RBFs cannot be carried out with the debug of ‘out of memory’ shown in MATLAB. Thus, the results of the two methods will not be shown in the example.

As shown in Table 5, irrespective of DTM resolution and data density, the proposed method with the RBF centers selected by the surface variation-based algorithm generally outperforms that by the random method, especially for high-resolution DTM generation. For example, for CSRBF1 with data percentage of 100%, the accuracy increasing ratio of surface variation-based method to the

random selection is 65.9% on 0.3-m-resolution, while that is only 4.7% on 2-m-resolution. CSRBF0 seems slightly more accurate than the other CSRBFs. This conclusion is inconsistent with that from the numerical test, which may be attributed to the fact that the surface of the study site is so coarse that the higher degree of smoothness assumed by some CSRBFs cannot be met.

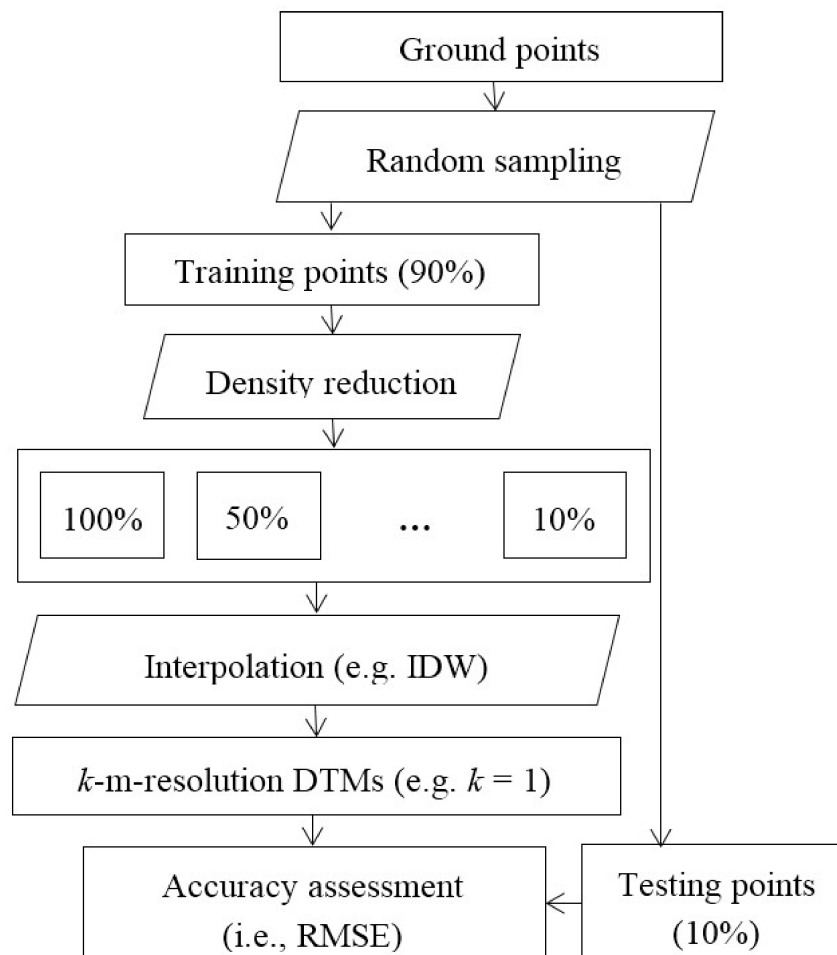


Figure 6. Flowchart of accuracy comparison between different interpolation methods.

Interpolation results (Table 6) indicate that data density and spatial resolution have great influences on DTM accuracy. The accuracy of DTMs consistently increases with finer resolution and denser data points. For example, as the data percentage decreases from 100% to 10% with the resolution of 0.3 m, the RMSE of the proposed method changes from 4.86 to 9.21 cm with the increasing ratio of 89.5%, while the increasing ratios of IDW, OK, linear, NN and bicubic are 117.4%, 82.9%, 101.4%, 103.4% and 121.9%, respectively. Those demonstrate that the proposed method, which is on par with OK, is more stable to data density variation than the other methods for generating high-resolution DTMs. In comparison, the bicubic interpolation is the most sensitive to data density variation, which is closely followed by IDW. As spatial resolution ranges from 0.3 to 2 m with the data percentage of 100%, the RMSE of the proposed method varies from 4.86 to 17.63 m with the increasing ratio of 262.8%, and the increasing ratios of IDW, OK, linear, NN and bicubic are 206.1%, 288.9%, 273.9%, 280.6% and 226.7%, respectively. Thus, our method is more robust to the variable of DTM resolution than OK under high data density.

Table 5. RMSEs of the proposed methods with the RBF centers selected by the surface variation (SV)-based algorithm and by the random scheme (unit: cm) (Note that bold font represents the best result).

Data Percentage (%)	Resolution (m)	CSRBF0		CSRBF1		CSRBF2		CSRBF3	
		SV	Rand	SV	Rand	SV	Rand	SV	Rand
100	0.3	4.86	6.54	5.07	8.41	4.92	7.82	5.10	8.66
	0.5	6.07	7.50	6.22	8.99	6.12	8.51	6.26	9.21
	1	10.09	10.82	10.14	11.61	10.14	11.34	10.19	11.74
	1.5	13.78	14.54	13.80	15.24	13.82	14.92	13.83	15.51
	2	17.63	18.15	17.53	18.35	17.58	18.31	17.64	18.25
50	0.3	5.60	5.80	5.64	6.20	5.64	6.20	5.79	6.35
	0.5	5.85	6.61	6.65	7.23	6.67	7.07	6.78	7.47
	1	10.13	10.43	10.20	10.63	10.21	10.68	10.29	10.91
	1.5	13.99	14.33	13.99	14.58	14.02	14.71	14.07	14.74
	2	17.91	17.92	18.21	19.13	17.51	18.07	17.55	18.17
30	0.3	6.17	8.71	6.14	6.68	6.22	6.92	6.38	7.21
	0.5	7.14	9.44	7.12	7.68	7.20	7.86	7.32	8.07
	1	10.44	12.53	10.50	10.91	10.51	10.93	10.62	11.13
	1.5	13.96	15.57	13.98	14.58	14.01	14.61	14.06	14.71
	2	17.79	19.19	17.74	18.41	17.80	18.57	18.38	19.02
10	0.3	9.21	9.27	8.97	9.79	9.10	10.44	9.42	10.76
	0.5	9.71	9.78	9.50	10.43	9.61	10.70	9.91	11.24
	1	12.07	12.07	11.95	12.50	12.09	12.94	12.32	13.05
	1.5	14.95	15.26	14.89	15.60	15.01	15.94	15.11	16.18
	2	18.24	18.59	18.24	19.13	18.38	19.70	18.45	19.54

Table 6. RMSEs of the well-known interpolation methods and the proposed method with CSRBF0 and RBF centers selected by the surface variation-based algorithm (unit: cm) (Note that bold font represents the best result).

Data Percentage (%)	Resolution (m)	IDW	OK	Linear	NN	Bicubic	Proposed
100	0.3	6.04	4.86	5.05	4.95	5.66	4.86
	0.5	7.26	6.28	6.39	6.33	6.74	6.07
	1	10.98	10.74	10.75	10.69	10.59	10.09
	1.5	14.68	14.82	14.83	14.78	14.51	13.78
	2	18.49	18.90	18.88	18.84	18.49	17.63
50	0.3	7.25	5.51	5.82	5.77	6.83	5.60
	0.5	8.07	6.59	6.86	6.78	7.62	5.85
	1	11.10	10.50	10.61	10.53	10.78	10.13
	1.5	14.83	14.69	14.81	14.74	14.55	13.99
	2	18.36	18.64	18.57	18.51	18.16	17.91
30	0.3	8.51	6.00	6.54	6.46	8.21	6.17
	0.5	9.28	7.24	7.76	7.64	8.90	7.14
	1	11.88	10.77	11.02	10.94	11.58	10.44
	1.5	15.19	14.76	14.88	14.81	14.96	13.96
	2	18.67	18.72	18.85	18.75	18.54	17.79
10	0.3	13.13	8.89	10.17	10.07	12.56	9.21
	0.5	13.50	9.51	10.75	10.63	12.87	9.71
	1	15.16	12.10	13.13	13.03	14.59	12.07
	1.5	17.56	15.39	16.09	16.04	17.07	14.95
	2	20.25	18.86	19.37	19.25	19.75	18.24

Figure 7 shows that the proposed method is obviously superior to the other methods with respect to average RMSE on the five resolutions, especially for the high data density, and IDW always has the lowest performance. This further indicates the high robustness of the proposed method to spatial resolution change.

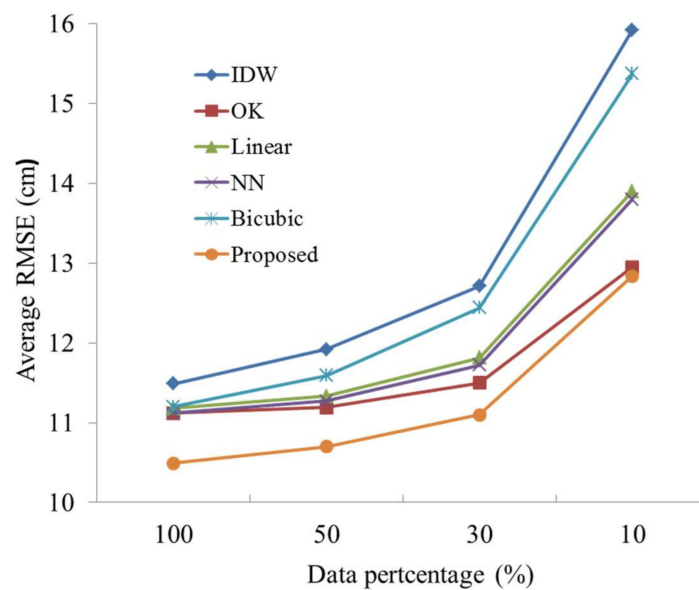


Figure 7. Average RMSEs of the proposed method and the well-known interpolation methods under different resolutions (unit: cm).

RMSE as a global accuracy measure may give a biased assessment to the interpolation methods [54]. Thus, shaded relief maps of all methods were generated. Results demonstrate that the surface of IDW is very coarse and has some fake depressions, such as those marked in the circle (Figure 8a). OK keeps the terrain features very well (Figure 8b), but has a relatively coarse appearance. Both linear (Figure 8c) and NN (Figure 8d) have a peak-cutting problem, producing many flat areas with an abnormal appearance. The bicubic surface is fragmented on some terrain features (Figure 8e). By contrast, the proposed method obtains a good tradeoff between noise removal and subtle feature preservation (Figure 8f).

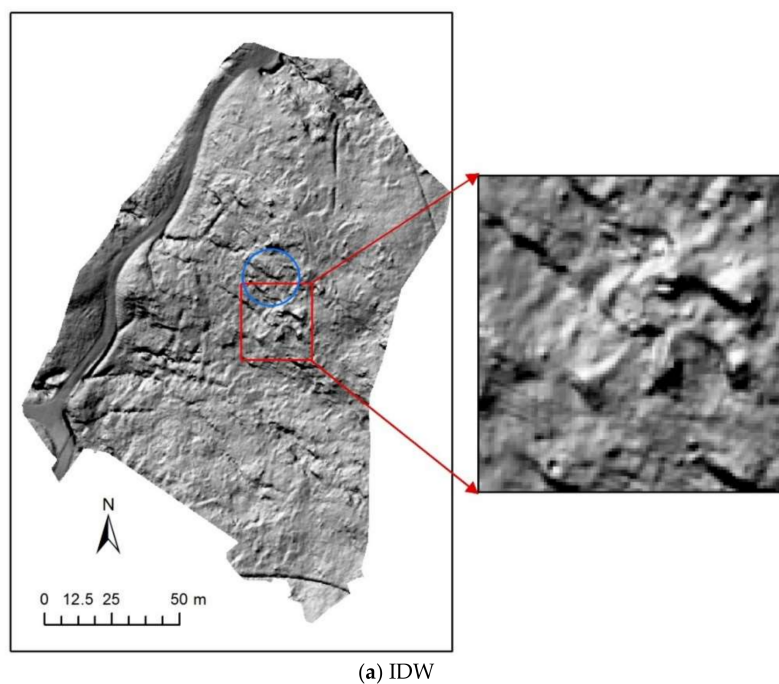
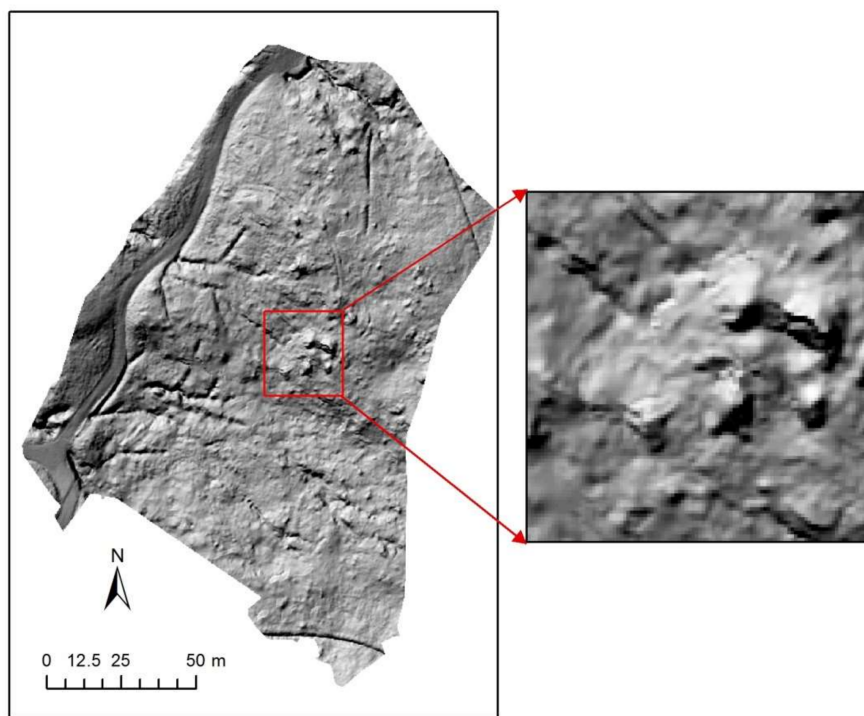
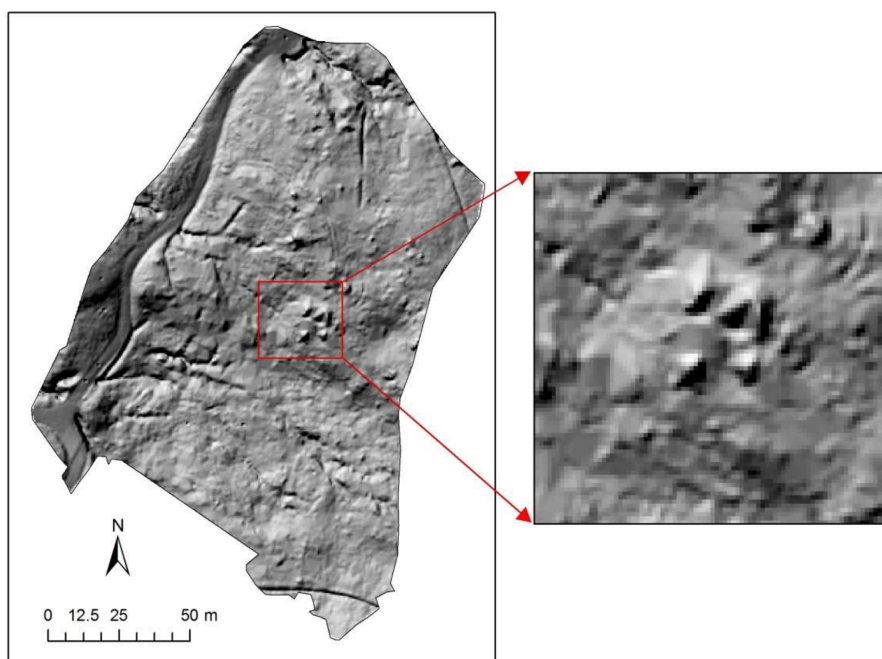


Figure 8. Cont.

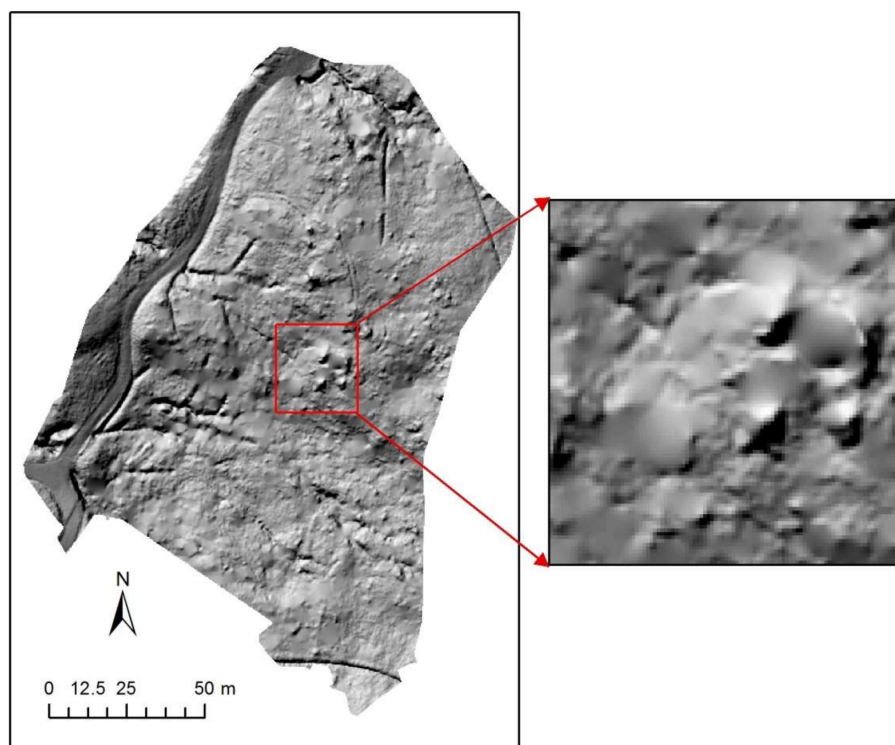


(b) OK

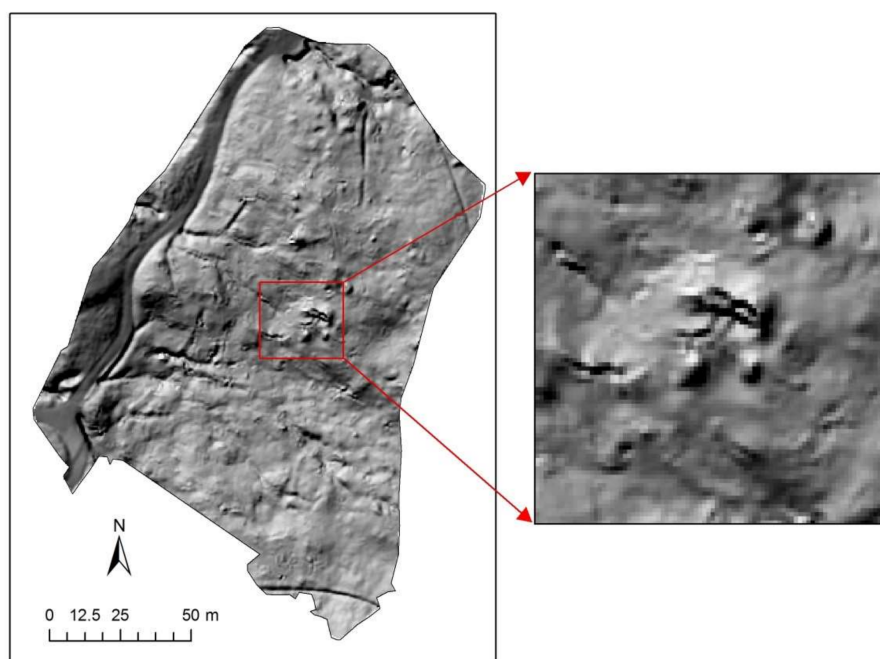


(c) Linear

Figure 8. *Cont.*



(d) NN



(e) Bicubic

Figure 8. *Cont.*

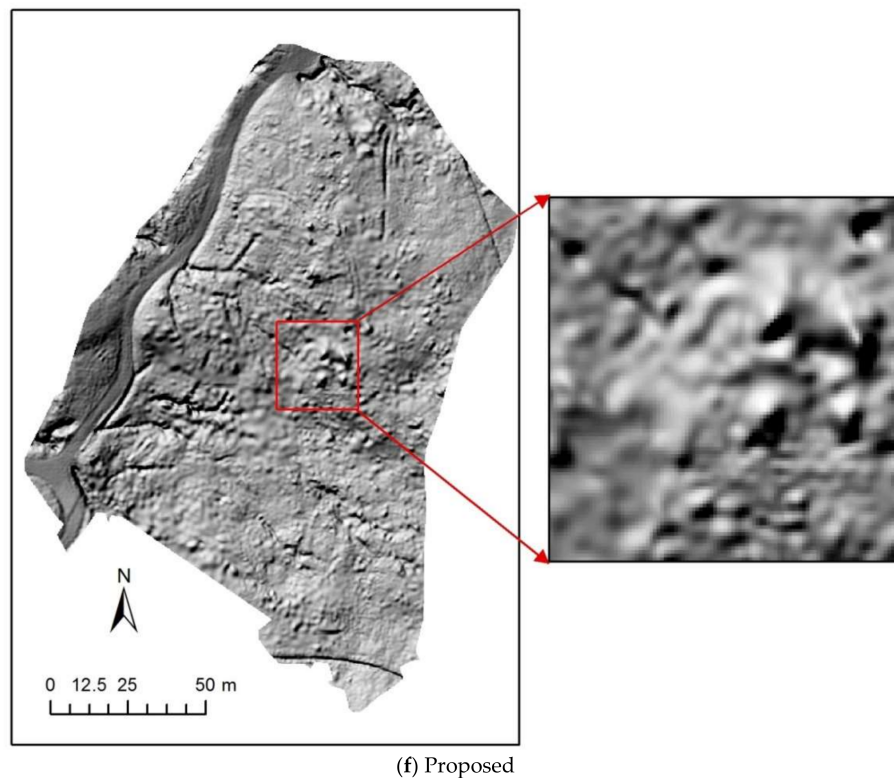


Figure 8. Shaded relief maps of the proposed method and the well-known interpolation methods for the data percentage of 100%.

The computational costs (Table 7) show that IDW is the fastest method. This result can be expected, as IDW does not require the solution of a linear system. Linear, NN and Bicubic have similar computational costs. OK has the largest computational cost, because it requires the fitting of semivariogram and the solution of a linear system with a dense matrix. In comparison, the proposed method is faster than OK, and slower than the other methods. The superior performance of the proposed method to OK is attributed to its sparse interpolation matrix and fewer RBF centers than the sample points. On average, the proposed method is about 4 times faster than OK.

Table 7. Computational costs of the proposed method and the well-known interpolation methods for the data percentage of 100% (unit: s).

Resolution (m)	IDW	OK	Linear	NN	Bicubic	Proposed
0.3	7.09	1464.6	36.47	36.82	39.75	160
0.5	2.93	672.1	13.61	13.67	14.22	130
1	1.13	355.8	3.88	3.84	3.78	127
1.5	0.79	296.0	2.02	2.02	1.84	124
2	0.67	277.5	1.38	1.39	1.17	120
On average	2.5	613.2	11.5	11.6	12.2	132.2

4.2.2. Public Datasets

Ten benchmark datasets with different terrain characteristics, provided by ISPRS Commission III [55] were further employed to assess the performance of the proposed method and the aforementioned interpolation methods. Note that the proposed method with CSRBF0 and RBF centers selected by the surface variation-based algorithm was adopted due to the high interpolation accuracy shown on both the simulated and private datasets. Since the label of each point (i.e., ground

or nonground) in the benchmark datasets are known, only the ground points were selected to generate DTMs. The flowchart in this example was similar to Figure 6. The differences were as follows: (1) the sample points were not reduced due to the low sample density and (2) the spatial resolution was set to be 0.1 m to reduce the level of location error on DTM accuracy [56]. The numbers of training points (#training) and testing points (#testing), area and data density for each sample were listed in Table 8. Moreover, the standard deviation (STD) of elevations for describing terrain characteristic was also provided. We can see that the 10 samples with different areas have various terrain complexities.

Table 8. Data description for each sample.

Sample	#Training	#Testing	Area (ha)	Density (point/m ²)	STD (m)
Samp11	19,607	2179	4.05	0.53	29.2
Samp12	24,021	2670	5.40	0.49	3.1
Samp21	9076	1009	1.43	0.71	0.6
Samp22	20,253	2251	3.40	0.66	2.8
Samp31	14,000	1556	2.82	0.55	0.9
Samp41	5041	561	1.75	0.32	3.5
Samp51	12,555	1395	9.99	0.14	15.1
Samp52	18,100	2012	13.55	0.15	28.8
Samp61	30,468	3386	22.39	0.15	7.1
Samp71	12,487	1388	8.73	0.16	3.0

Figure 9 demonstrates that irrespective of interpolation methods, terrain characteristics have a significant effect on the accuracy of DTMs. More specifically, as the complexity of the terrain increases, the quality of DTMs decreases. For example, regardless of interpolation methods, samp21 and samp31, less complex than the other samples, obtains much better results. Samp11 with the greatest terrain complexity has the worst result, which is followed by samp52. The proposed method consistently performs better than the other methods on all the samples, especially for the complex terrains, such as samp11, 41, 52 and 61. This indicates that the new method has the most robustness to terrain characteristics, which is followed by OK. IDW seems to rank lowest, as it has four samples with the lowest accuracy (i.e., samp41, 51, 52 and 71). Linear and bicubic interpolators rank second lowest, both of which have three samples (i.e., samp11, 12 and 31 for linear, and samp21, 22 and 61 for bicubic).

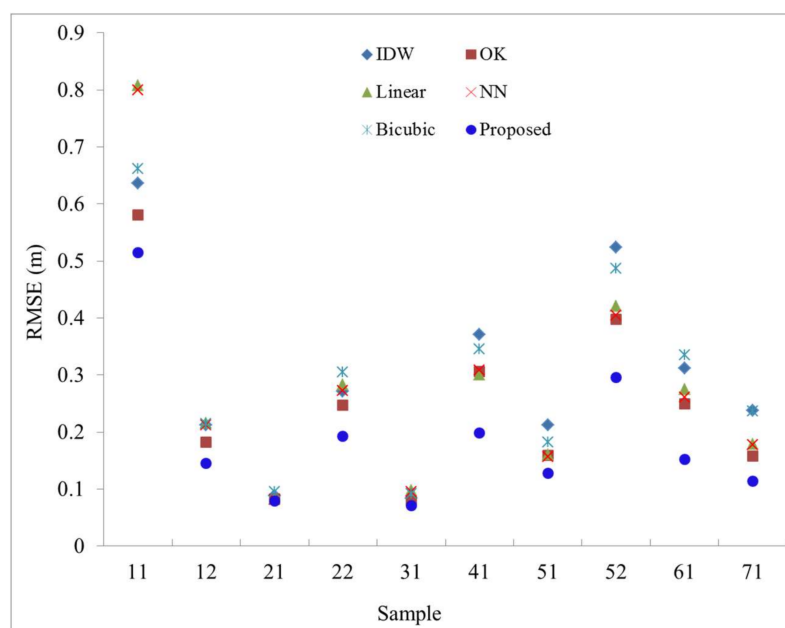


Figure 9. RMSEs of the proposed method and the well-known interpolation methods for each sample.

The shaded relief maps of all the interpolation methods on samp51 were generated (Figure 10). The selection of samp51 is mainly due to its complex terrain characteristics with obvious breaklines and subtle terrain features. Results indicate that the IDW method has the coarsest surface, which is full of bumps, especially on the ramp (Figure 10a). The linear method has obvious discontinuities due to the variable distribution of sample points (Figure 10c). NN is much better than linear, yet the former suffers from the peak-cutting problem (Figure 10d). The surface of bicubic interpolation is so smooth that almost all subtle terrain features are lost (Figure 10e). OK seems to have a much better surface than the classical interpolation methods (Figure 10b). However, it is subject to tree root-like discontinuities in the right-middle area. In comparison, the proposed method produces a visually appealing surface (Figure 10f). Nevertheless, some subtle terrain features are blurred.

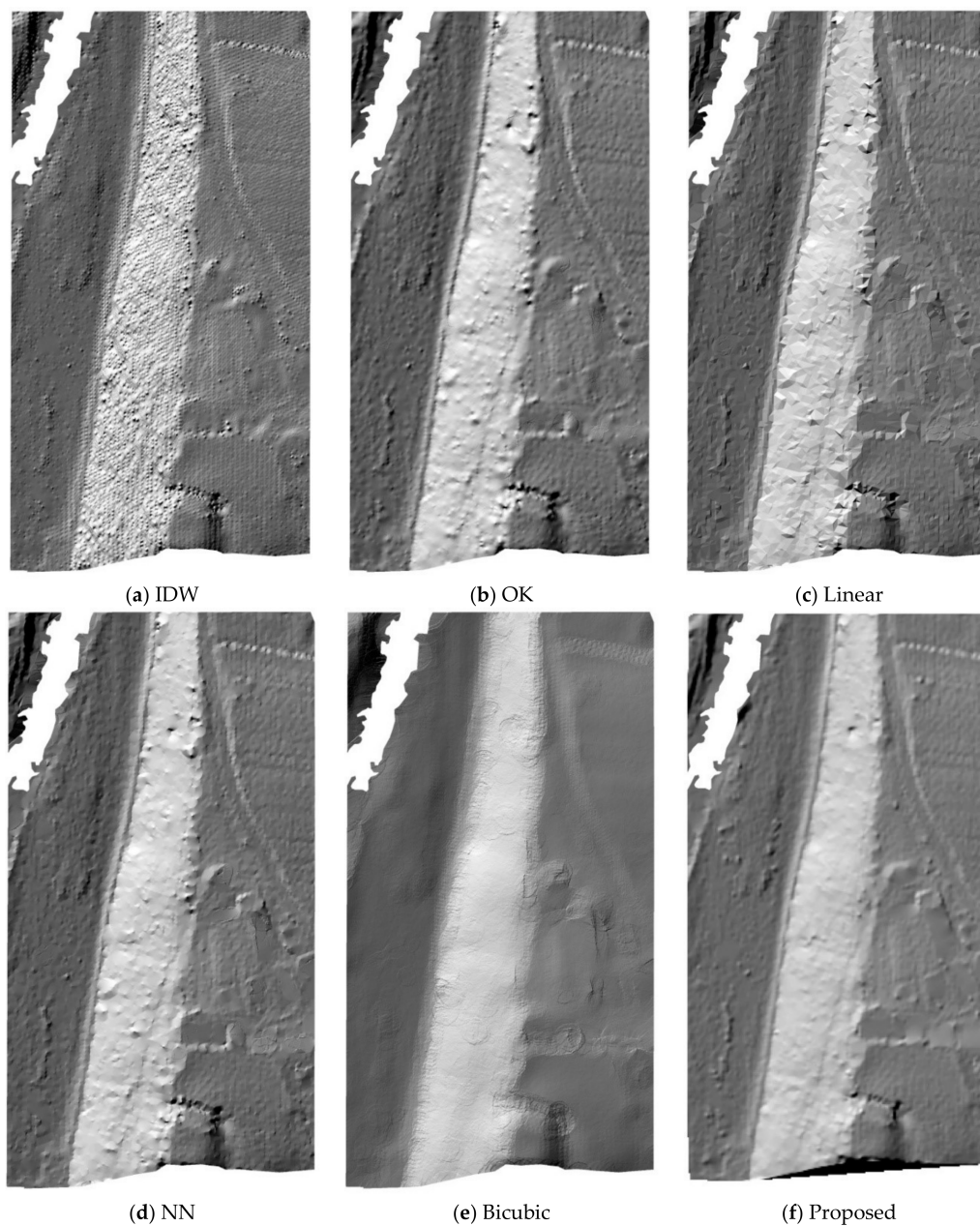


Figure 10. Shaded relief maps of the proposed method and the well-known interpolation methods on samp51.

The computing costs of all the interpolation methods on the ten public datasets are much larger than those on the private dataset (Tables 7 and 9). This is because the number of DTM grids on the former is much larger than that on the latter. For example, among the ten public datasets, samp21 with the smallest area has 1152×1238 DTM grids with the 0.1-m-resolution, while the private dataset with the area of 2 ha has only 723×535 grids with the resolution of 0.3 m. However, the ranks of the interpolators with respect to computing cost are unchanged. Namely, IDW is the fastest method, which is followed by NN, linear and bicubic. OK has the largest computing time. The proposed method is faster than OK, yet slower than the other methods. On average, the speed of the proposed method is about 4 times as high as that of OK.

Table 9. Computational costs of the proposed method and the well-known interpolation methods on the public datasets (unit: s).

Sample	IDW	OK	Linear	NN	Bicubic	Proposed
Samp11	71.1	2393.7	431.5	429.4	531.8	802.7
Samp12	93.0	3157.4	604.7	593.5	699.4	839.5
Samp21	23.6	821.3	150.4	147.3	187.5	247.4
Samp22	56.8	1861.5	368.1	359.9	448.7	671.7
Samp31	47.2	1590.5	310.0	302.8	370.4	492.0
Samp41	29.6	943.9	225.4	223.4	227.8	387.6
Samp51	167.6	6311.3	1015.4	995.7	1303.6	1072.9
Samp52	222.7	7542.9	1366.2	1355.9	1782.4	2178.7
Samp61	1205.6	12,687.0	2386.8	2403.4	2971.6	3111.3
Samp71	148.6	4883.0	884.8	878.5	1132.0	1433.0
On average	206.6	4219.3	774.3	769.0	965.5	1123.7

5. Discussion and Conclusions

5.1. Discussion

Nowadays, LiDAR is considered as one of the most promising techniques for the derivation of DTMs. Due to the huge volume of LiDAR data, simple interpolation methods were commonly recommended. For example, Anderson et al. [8] indicated that IDW could be sufficient for interpolating irregularly spaced LiDAR datasets. Bater and Coops [9] showed that NN provided the best results with a minimum of effort. Guo et al. [11] demonstrated that simple interpolation methods, such as IDW and NN, were more efficient at generating DTMs from LiDAR data. Montealegre et al. [1] identified that IDW was the best method for LiDAR-derived DTMs in the validation with GNSS-captured check points. However, our results show that the simple interpolation methods are highly sensitive to data density and variation in terrain characteristics. As demonstrated in Table 5, when the data percentage decreases from 100% to 10%, the RMSE increasing ratio of IDW is 117.4% while that of the proposed method is only 89%. Moreover, the simple interpolators always produce visually unsatisfactory surfaces. As shown in Figures 8 and 10, the surface of IDW is very coarse, making some terrain features unrecognizable. Linear and NN show obvious artifacts like surface discontinuity and peak-cutting problems due to the variable of data density. Bicubic interpolation loses almost all subtle terrain features. Comparatively, the proposed method obtains a good tradeoff between noise removal and terrain feature preservation.

When the required DTM resolution is higher than the average sample interval of LiDAR points, sophisticated interpolators are more helpful. Chu et al. [10] pointed out that in sparse LiDAR samples, landslide scarp identification is sensitive to interpolation methods in the dense forest, and the DTMs from kriging are more useful than those of IDW for landslide scarp detection. Guo et al. [11] regarded that kriging-based methods are more reliable if accuracy is the most important consideration. Nevertheless, kriging has a huge computational cost. As listed in Tables 6, 7 and 9, although OK has a relatively high performance, it is much slower than the proposed method. In comparison, the proposed method has a comparative accuracy to OK, but the former is more efficient than the latter. Moreover,

the proposed method is less sensitive to the variable of spatial resolution and terrain characteristics, as shown in Figures 7 and 9.

5.2. Conclusions

To overcome the huge volume and noise problems of LiDAR data, we proposed a least squares CSRBF interpolation method for the derivation of DTMs. The proposed method was based on the solution of a sparse linear system with fewer RBF centers than the sample points and a limited support radius. Thanks to the least squares nature and the sparse interpolation matrix, the proposed method was not only fast but also robust with respect to noise. The test on the synthetic dataset demonstrated that compared with the smoothing RBF and the exact RBF, our method has a comparable performance to the smoothing RBF, yet the former has a higher speed than the latter. Moreover, the proposed method with the RBF centers selected by the surface variation-based algorithm is more accurate than that by the random method. The experiments on LiDAR datasets showed that compared to the simple interpolation methods like IDW and NN, the proposed method produced a visually satisfactory surface. Compared to OK, the new method has a higher computing speed, and is less sensitive to DTM resolution variation and terrain complexity. Namely, our method maintains a good balance between interpolation accuracy and computational cost. Note that the input of the proposed method is the randomly distributed LiDAR ground points, i.e., all off ground points were completely filtered out. However, if the filtered ground points contained non-ground points, advanced techniques, such as outlier detection [57,58] and robust interpolations [17,59], should be adopted, as non-ground points could be considered as outliers in the context of DTM construction and our method is sensitive to outliers due to the LS nature.

Acknowledgments: The authors wish to express their sincere gratitude to the three anonymous reviewers for their assistance, comments and suggestions. This work was supported by the National Natural Science Foundation of China (Grant No. 41371367), the SDUST Research Fund and the Joint Innovative Center for Safe and Effective Mining Technology and Equipment of Coal Resources. The filtered private LiDAR dataset can be downloaded from the website (<https://drive.switch.ch/index.php/s/kd34pMRY5mH4ytV>).

Author Contributions: C.C. and Y.L. conceived and designed the experiments; C.C. performed the experiments; C.C. and N.Z. analyzed the data; B.G. and N.M. contributed analysis tools; C.C. wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest and the founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Montealegre, A.; Lamelas, M.; Riva, J. Interpolation routines assessment in ALS-derived digital elevation models for forestry applications. *Remote Sens.* **2015**, *7*, 8631–8654. [CrossRef]
2. Xiong, L.; Wang, G.; Wessel, P. Anti-aliasing filters for deriving high-accuracy DEMs from TLS data: A case study from Freeport, Texas. *Comput. Geosci.* **2017**, *100*, 125–134. [CrossRef]
3. Wilson, J.P. Digital terrain modeling. *Geomorphology* **2012**, *137*, 107–121. [CrossRef]
4. Lin, X.; Zhang, J. Segmentation-based filtering of airborne LiDAR point clouds by progressive densification of terrain segments. *Remote Sens.* **2014**, *6*, 1294–1326. [CrossRef]
5. Tarolli, P. High-resolution topography for understanding Earth surface processes: Opportunities and challenges. *Geomorphology* **2014**, *216*, 295–312. [CrossRef]
6. Razak, K.A.; Straatsma, M.W.; van Westen, C.J.; Malet, J.P.; de Jong, S.M. Airborne laser scanning of forested landslides characterization: Terrain model quality and visualization. *Geomorphology* **2011**, *126*, 186–200. [CrossRef]
7. Shi, W.; Zheng, S.; Tian, Y. Adaptive mapped least squares SVM-based smooth fitting method for DSM generation of LIDAR data. *Int. J. Remote Sens.* **2009**, *30*, 5669–5683. [CrossRef]
8. Anderson, E.S.; Thompson, J.A.; Austin, R.E. LIDAR density and linear interpolator effects on elevation estimates. *Int. J. Remote Sens.* **2005**, *26*, 3889–3900. [CrossRef]
9. Bater, C.W.; Coops, N.C. Evaluating error associated with lidar-derived DEM interpolation. *Comput. Geosci.* **2009**, *35*, 289–300. [CrossRef]

10. Chu, H.; Wang, C.; Huang, M.; Lee, C.; Liu, C.; Lin, C. Effect of point density and interpolation of LiDAR-derived high-resolution DEMs on landscape scarp identification. *GISci. Remote Sens.* **2014**, *51*, 731–747. [[CrossRef](#)]
11. Guo, Q.; Li, W.; Yu, H.; Alvarez, O. Effects of topographic variability and Lidar sampling density on several DEM interpolation methods. *Photogramm. Eng. Remote Sens.* **2010**, *76*, 701–712. [[CrossRef](#)]
12. Mitasova, H.; Mitas, L.; Harmon, R.S. Simultaneous spline approximation and topographic analysis for lidar elevation data in open-source GIS. *IEEE Geosci. Remote Sens. Lett.* **2005**, *2*, 375–379. [[CrossRef](#)]
13. Hofierka, J.; Cebecauer, T. Spatial interpolation of elevation data with variable density: A new methodology to derive quality DEMs. *IEEE Geosci. Remote Sens. Lett.* **2007**, *4*, 117–121. [[CrossRef](#)]
14. Aguilar, F.J.; Agüera, F.; Aguilar, M.A.; Carvajal, F. Effects of terrain morphology, sampling density, and interpolation methods on grid DEM accuracy. *Photogramm. Eng. Remote Sens.* **2005**, *71*, 805–816. [[CrossRef](#)]
15. Majdisova, Z.; Skala, V. Big geo data surface approximation using radial basis functions: A comparative study. *Comput. Geosci.* **2017**, *109*, 51–58. [[CrossRef](#)]
16. Wendland, H. *Scattered Data Approximation*; Cambridge University Press: Cambridge, UK, 2004; Volume 17.
17. Chen, C.F.; Liu, F.Y.; Li, Y.Y.; Yan, C.Q.; Liu, G.L. A robust interpolation method for constructing digital elevation models from remote sensing data. *Geomorphology* **2016**, *268*, 275–287. [[CrossRef](#)]
18. Chen, C.F.; Li, Y.Y.; Zhao, N.; Yan, C.Q. Robust interpolation of DEMs from Lidar-derived elevation data. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 1059–1068. [[CrossRef](#)]
19. Beatson, R.K.; Cherrie, J.B.; Mouat, C.T. Fast fitting of radial basis functions: Methods based on preconditioned GMRES iteration. *Adv. Comput. Math.* **1999**, *11*, 253–270. [[CrossRef](#)]
20. Faul, A.C.; Goodsell, G.; Powell, M.J.D. A Krylov subspace algorithm for multiquadric interpolation in many dimensions. *IMA J. Numer. Anal.* **2005**, *25*, 1–24. [[CrossRef](#)]
21. Gumerov, N.A.; Duraiswami, R. Fast radial basis function interpolation via preconditioned Krylov iteration. *SIAM J. Sci. Comput.* **2007**, *29*, 1876–1899. [[CrossRef](#)]
22. Crivellaro, A.; Perotto, S.; Zonca, S. Reconstruction of 3D scattered data via radial basis functions by efficient and robust techniques. *Appl. Numer. Math.* **2017**, *113*, 93–108. [[CrossRef](#)]
23. Hardy, R. Theory and applications of the multiquadric-biharmonic method. 20 years of discovery 1968–1988. *Comput. Math. Appl.* **1990**, *19*, 163–208. [[CrossRef](#)]
24. Beatson, R.; Light, W.; Billings, S. Fast solution of the radial basis function interpolation equations: Domain decomposition methods. *SIAM J. Sci. Comput.* **2001**, *22*, 1717–1740. [[CrossRef](#)]
25. Wendland, H. Fast evaluation of radial basis functions: Methods based on partition of unity. In *Approximation Theory X: Wavelets, Splines, and Applications*; Chui, C.K., Schumaker, L.L., Stöckler, J., Eds.; Vanderbilt University Press: Nashville, TN, USA, 2002; pp. 473–483.
26. Deng, X.S.; Tang, Z.A. Moving surface spline interpolation based on Green's function. *Math. Geosci.* **2011**, *43*, 663–680. [[CrossRef](#)]
27. Fasshauer, G.E. *Meshfree Approximation Methods with MATLAB*; World Scientific Publishing: Singapore, 2007.
28. Billings, S.D.; Newsam, G.N.; Beatson, R.K. Smooth fitting of geophysical data using continuous global surfaces. *Geophysics* **2002**, *67*, 1823–1834. [[CrossRef](#)]
29. Carr, J.C.; Beatson, R.K.; Cherrie, J.B.; Mitchell, T.J.; Fright, W.R.; McCallum, B.C.; Evans, T.R. *Reconstruction and Representation of 3D Objects with Radial Basis Functions*; Siggraph/ACM: Los Angeles, CA, USA, 2001; pp. 67–76.
30. Chen, S.; Cowan, C.; Grant, P. Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Trans. Neural Netw.* **1991**, *2*, 302–309. [[CrossRef](#)] [[PubMed](#)]
31. Franke, R.; Hagen, H.; Nielson, G. Least squares surface approximation to scattered data using multiquadratic functions. *Adv. Comput. Math.* **1994**, *2*, 81–99. [[CrossRef](#)]
32. Ohtake, Y.; Belyaev, A.; Seidel, H.P. 3D scattered data approximation with adaptive compactly supported radial basis functions. In *Proceedings of the Shape Modeling Applications*, Genova, Italy, 7–9 June 2004; pp. 31–39.
33. Süßmuth, J.; Meyer, Q.; Greiner, G. Surface reconstruction based on hierarchical floating radial basis functions. *Comput. Gr. Forum* **2010**, *29*, 1854–1864. [[CrossRef](#)]
34. Skala, V. Fast interpolation and approximation of scattered multidimensional and dynamic data using radial basis functions. *WSEAS Trans. Math.* **2013**, *12*, 501–511.
35. Bjorck, A. *Numerical Methods for Least Squares Problems*; SIAM: Philadelphia, PA, USA, 1996.

36. Hardy, R.L. Multiquadric equations of topography and other irregular surfaces. *J. Geophys. Res.* **1971**, *76*, 1905–1915. [CrossRef]
37. Pauly, M.; Gross, M.; Kobbelt, L.P. Efficient simplification of point-sampled surfaces. In Proceedings of the Conference on Visualization'02, Boston, MA, USA, 27 October–1 November 2002; IEEE Computer Society: Washington, DC, USA, 2002; pp. 163–170.
38. Saad, Y. *Iterative Methods for Sparse Linear Systems*; SIAM: Philadelphia, PA, USA, 2003.
39. Hofierka, J.; Cebecauer, T.; Šúri, M. Optimisation of interpolation parameters using cross-validation. In *Digital Terrain Modelling*; Peckham, R.J., Jordan, G., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 67–82.
40. Chen, C.F.; Zhao, N.; Yue, T.; Guo, J.Y. A generalization of inverse distance weighting method via kernel regression and its application to surface modeling. *Arabian J. Geosci.* **2015**, *8*, 6623–6633. [CrossRef]
41. Lu, Y.G.; Wong, D.W. An adaptive inverse-distance weighting spatial interpolation technique. *Comput. Geosci.* **2008**, *34*, 1044–1055. [CrossRef]
42. Oliver, M.A.; Webster, R. Kriging: A method of interpolation for geographical information systems. *Int. J. Geogr. Inf. Sci.* **1990**, *4*, 313–332. [CrossRef]
43. Goovaerts, P. *Geostatistics for Natural Resources Evaluation*; Oxford University Press: Oxford, UK, 1997.
44. Chiles, J.P.; Delfiner, P. *Geostatistics: Modeling Spatial Uncertainty*; Wiley: New York, NY, USA, 1999.
45. Kidner, D.B. Higher-order interpolation of regular grid digital elevation models. *Int. J. Remote Sens.* **2003**, *24*, 2981–2987. [CrossRef]
46. Shi, W.Z.; Li, Q.Q.; Zhu, C.Q. Estimating the propagation error of DEM from higher-order interpolation algorithms. *Int. J. Remote Sens.* **2005**, *26*, 3069–3084. [CrossRef]
47. Sibson, R. *A Brief Description of Natural Neighbour Interpolation*; Interpreting Multivariate Data; John Wiley & Sons: Hoboken, NJ, USA, 1981; pp. 21–36.
48. Ledoux, H.; Gold, C. An efficient natural neighbour interpolation algorithm for geoscientific modelling. In *Developments in Spatial Data Handling*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 97–108.
49. Ordinary Kriging-File Exchange. 2010. Available online: <https://cn.mathworks.com/matlabcentral/fileexchange/29025-ordinary-kriging> (accessed on 10 February 2018).
50. Experimental (Semi-) Variogram-File Exchange. 2013. Available online: <https://cn.mathworks.com/matlabcentral/fileexchange/20355-experimental-semi---variogram> (accessed on 10 February 2018).
51. variogramfit-File Exchange. 2010. Available online: <https://cn.mathworks.com/matlabcentral/fileexchange/25948-variogramfit> (accessed on 10 February 2018).
52. Baltensweiler, A.; Walthert, L.; Ginzler, C.; Sutter, F.; Purves, R.S.; Hanewinkel, M. Terrestrial laser scanning improves digital elevation models and topsoil pH modelling in regions with complex topography and dense vegetation. *Environ. Model. Softw.* **2017**, *95*, 13–21. [CrossRef]
53. Erdogan, S. A comparison of interpolation methods for producing digital elevation models at the field scale. *Earth Surf. Process. Landf.* **2009**, *34*, 366–376. [CrossRef]
54. Carlisle, B.H. Modelling the spatial distribution of DEM error. *Trans. GIS* **2005**, *9*, 521–540. [CrossRef]
55. Sithole, G.; Vosselman, G. Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds. *ISPRS J. Photogramm. Remote Sens.* **2004**, *59*, 85–101. [CrossRef]
56. Yue, T.X.; Du, Z.P.; Song, D.J. A new method of surface modelling and its application to DEM construction. *Geomorphology* **2007**, *91*, 161–172. [CrossRef]
57. Liu, H.; Jezek, K.C.; O'Kelly, M.E. Detecting outliers in irregularly distributed spatial data sets by locally adaptive and robust statistical analysis and GIS. *Int. J. Geogr. Inf. Sci.* **2001**, *15*, 721–741. [CrossRef]
58. Nurunnabi, A.; West, G.; Belton, D. Outlier detection and robust normal-curvature estimation in mobile laser scanning 3D point cloud data. *Pattern Recogn.* **2015**, *48*, 1400–1415. [CrossRef]
59. Chen, C.F.; Li, Y.Y.; Yan, C.Q.; Dai, H.L.; Liu, G.L. A robust algorithm of multiquadric method based on an improved huber loss function for interpolating remote-sensing-derived elevation data sets. *Remote Sens.* **2015**, *7*, 3347–3371. [CrossRef]

