A Closed-Form Formulation of HRBF-Based Surface Reconstruction by Approximate Solution

Shengjun Liu[1,2], Charlie C. L. Wang*[3], Guido Brunnett[4], and Jun Wang[5]

[1]*State Key Laboratory of High Performance Manufacturing Complex, Central South University, Changsha, Hunan 410083, China*

[2]*Institute of Engineering Modeling and Scientific Computing, Central South University, Changsha, Hunan 410083,China*

[3]*Department of Design Engineering, Delft University of Technology, The Netherlands*

[4]*Department of Computer Science, Chemnitz University of Technology, Chemnitz 09126, Germany*

[5]*College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China*

*(*Corresponding Author. Tel: +31 1 527 88406. Email: c.c.wang@tudelft.nl)*

Abstract

The *Hermite radial basis functions* (HRBFs) implicits have been used to reconstruct surfaces from scattered Hermite data points. In this work, we propose a closed-form formulation to construct HRBF-based implicits by a quasi-solution to approximate the exact one. A scheme is developed to automatically adjust the support sizes of basis functions to hold the error bound of a quasi-solution. Our method can generate an implicit function from positions and normals of scattered points without taking any global operation. Robust and efficient reconstructions are observed in our experimental tests on real data captured from a variety of scenes.

Keywords: Hermite Radial Basis Functions, Quasi-solution, Closed-Form, Surface Reconstruction

1. Introduction

Reconstructing surface from a set of unorganized points equipped with normal vectors is an important topic in various fields such as computer graphics, reverse engineering, image processing, mathematics, robotics, computer-aided design and manufacturing. A lot of research approaches have been devoted to developing surface reconstruction methods, in which implicit surface fitting based on *Radial Basis Functions* (RBF) is successful in dealing with noisy and incomplete data (e.g., [1, 2, 3]).

Recently, implicits based on *Hermite Radial Basis Functions* (HRBF) were presented to interpolate data points to the first order in [4]. It is robust and effective to deal with coarse and non-uniformly sampled points, close surface sheets, and able to produce surface reconstruction with details. However, interpolating both positions and normals of points leads to the computation of solving a $4n\times4n$ linear system for an input with n points. It is impractical due to the expensive computation. The system becomes sparse when the *Compactly Supported Radial Basis Functions* (CSRBF) are used as the kernel functions. However, this attempt on improving the efficiency can also bring in a more challenging problem of numerical stability. A closed-form formulation is presented in this paper to solve the computational problem of HRBF-based reconstruction by quasi-interpolation. Our formulation results in a kind of approximate interpolants that fit implicits by weighted averages of the values at given points. The most attractive property of this approach is able to robustly construct a surface from a set of points without solving linear systems – i.e., with a closed-form formulation (see Section 3.1). This can make the computation of HRBF-based surface reconstruction stable and efficient. As shown in Fig.1, a mesh surface with $331k$ triangles can be efficiently reconstructed from an input set with $922k$ points by our method in 5.5 seconds. Comparing to the recent *Floating Scale Surface Reconstruction* (FSSR) [5] that also avoids applying global operations, our method is more than 17.9× faster. Moreover, we have analyzed the error-bound between our closed-form solution and the solution obtained by solving linear systems (see Section 3.2). Specifically, the error-bound exists when the number of points covered by every support of all CSRBFs is capped by a fixed number, m. Experimental tests have been conducted to verify the efficiency and robustness of our approach.

1.1. Contributions

In this paper, we propose a closed-form formulation for computing the approximate solution of HRBF-based surface reconstruction from scattered data points.

- Our method can construct a signed scalar function by directly blending the positions and normals of points without any global operation. The computation based on compact support is local and numerically stable.
- Errors between the quasi-solution and the exact one are bounded when applying an automatic scheme to adjust the support sizes of basis functions.

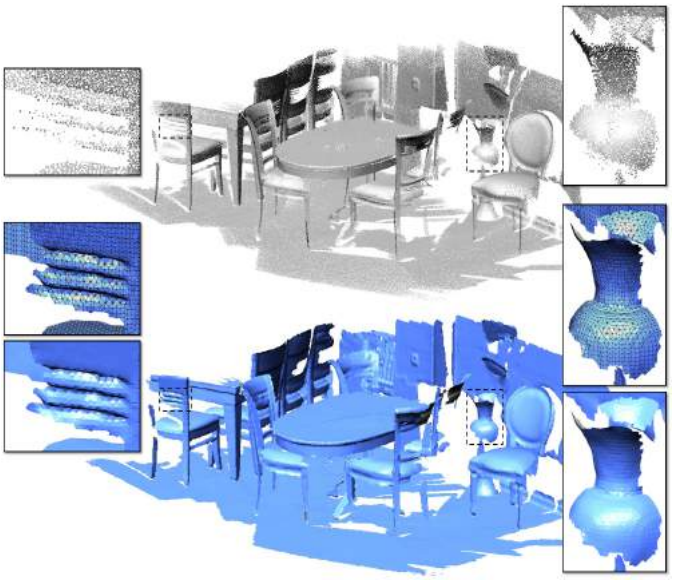

Figure 1: The method proposed in this paper can efficiently reconstruct a surface from a set of noisy and incomplete points – e.g., the indoor scene shown here with 922k points. Our reconstruction takes only 5.5 seconds to generate a mesh surface with 313k triangles have the similar quality as the state-of-the-art [5] but our approach is 17.9x faster. Note that the scale factor used in FSSR [5] is generated as 3× of support size used in our approach so that similar number of triangles are generated, and other default parameters are used for FSSR.

- Our formulation to find the approximate solution of HRBF-based surface reconstruction is robust. When compared with other approaches, this method can still successfully reconstruct a satisfactory surface on highly noisy points with up to 60% Gaussian noise.
- As a local approach, our method is efficient and scalable. This is well-suited for highly parallel implementation as well as distributed/progressive reconstruction.

Note that, the compactly-supported basis functions results in open meshes and leaves holes if the region does not have enough number of points, which actually fits the application of reconstructing outdoor scenes very well. Such holes also provide cues for the reliability of a reconstruction that can be used to supervise the active scanning and reconstruction (ref. [6]).

1.2. Related work

The problem of surface reconstruction from point cloud has been studied in literature for more than two decades. After using signed distance field in [7] to reconstruct mesh surface from point clouds, implicit functions have gained a lot of attention in surface reconstruction because of its ability to handle topological change and filling holes. Example approaches include RBF-based methods [4, 1, 2, 8, 9, 10, 11, 12, 13, 14, 15, 16], Poisson surface reconstructions [17, 18], smooth signed distance method [19], and *Partition-Of-Unity* (POU) based methods [20, 21]. A comprehensive review of all these works has beyond the scope of this paper. More discussion and comparison on different surface reconstruction methods can be found in [22] and the recent survey paper [23]. Here we only review methods that are closely related to our approach.

The methods based on RBF implicits are popular for their capability of handling sparse point clouds. Generally, RBF-based methods transform the reconstruction into a multi-variable optimization problem, where enforcing the interpolation constraints results in a linear system. Solving the linear system is an important but time-consuming step for the RBF-based reconstruction. To obtain a non-trivial solution, RBF-based methods usually require the provision of extra offset-points (ref. [1, 2]) that can be obtained by shifting data points along their normal directions. However, it is not easy to find an optimal offset distance. The positions of these offset points is also difficult to determine (especially when the scanned model has small features). To avoid generating offset-points, Ohtake et al. [9, 13] used a signed function which includes basic approximations and local details. Some prior works [10, 15], deduced from the statistical-learning perspective, avoid generating offset-points in surface reconstruction, where normals were directly used in a variational formulation. Recently, Macedo et al. [4] derive an implicit function from the Hermite-Birkhoff interpolation with RBFs. They enhance the flexibility of HRBF reconstruction by ensuring well-posedness of an interesting approach combining regularization. However, given a set with n Hermite points, these methods result in a $4n \times 4n$ linear system to be solved, which limits the number of points in reconstruction.

Quasi-interpolation is a simple, efficient, and computational stable method in the field of function approximation. In the early work of quasi-interpolation [24], a function approximating a given data set is defined by a weighted average of the values at the data points. The quasi-interpolation with RBF kernels has been studied in [25, 26], which is later employed for surface reconstruction [27, 28]. However, when the problem extended to HRBF based surface reconstruction, these quasi-interpolation techniques cannot be directly applied as the implicit function of HRBF does not follow the required form of quasi-interpolation. Our work presented in this paper overcome this difficulty. To the best of our knowledge, this is the first approach that provides a closed-form solution for HRBF-based surface reconstruction.

General speaking, *Moving Least-Squares* (MLS) based methods [29, 30, 31, 32, 33, 34], also construct a semi-implicit function by weighted sum of data points. Normal constraints are enforced by either approximation or interpolation in a few MLS approaches (ref. [32, 30, 34]). The method of Shen et al. [32] is mainly for polygonal meshes. Therefore, we only compare our results with *Algebraic Point Set Surfaces* (APSS) [30] and *Hermite Point Set Surfaces* [34] in the experimental tests and find that our method is more robust on noisy input. In the recent work named as FSSR [5], weighted average is also employed to fit implicit functions to the input set of points. The final implicit functions are formed by locally supported basis functions satisfying the property of POU. However, FSSR is 8.5× to 36.4× slower than our method when generating a reconstruction with similar number of triangles. More details can be found in Section 5.

2. HRBF Implicits

The HRBF implicits [4] are built upon the theory of Hermite-Birkhoff interpolation with radial basis functions [3]. In this section, we briefly describe how to use HRBF implicits to solve the problem of surface reconstruction from scattered points.

Definition 1 Given a set of data $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n\}$ with unit normals $\mathcal{N} = \{\mathbf{n}_1, \mathbf{n}_2, \cdots, \mathbf{n}_n\}$, the HRBF implicits give a function f interpolating both the points and the normal vectors as

$$f(\mathbf{x}) = \sum_{j=1}^{n} \{a_j \varphi(\mathbf{x} - \mathbf{p}_j) - \langle \mathbf{b}_j, \nabla\varphi(\mathbf{x} - \mathbf{p}_j)\rangle\}, \tag{1}$$

where $\varphi : \Re^3 \mapsto \Re$ is defined by a radial basis function $\varphi(\mathbf{x}) = \phi_\rho(\|\mathbf{x}\|)$, $\langle\cdot,\cdot\rangle$ denotes the dot-product of two vectors, and ∇ is the gradient operator.

Both the scalar coefficients, $a_j \in \Re$, and the vector coefficients, $\mathbf{b}_j \in \Re^3$, can be computed by the constraints of interpolation as

$$f(\mathbf{p}_i) = 0 \;\text{ and }\; \nabla f(\mathbf{p}_i) = \mathbf{n}_i, \quad (i = 1, 2, \cdots, n). \tag{2}$$

Applying these constraints to Eq.(1), we obtain a linear system with equations

$$\begin{array}{l} \sum_{j=1}^{n} \{a_j \varphi(\mathbf{p}_i - \mathbf{p}_j) - \langle \mathbf{b}_j, \nabla\varphi(\mathbf{p}_i - \mathbf{p}_j)\rangle\} = c, \\ \sum_{j=1}^{n} \{a_j \nabla\varphi(\mathbf{p}_i - \mathbf{p}_j) - \mathbf{b}_j \mathbf{H}\varphi(\mathbf{p}_i - \mathbf{p}_j)\} = \mathbf{n}_i, \end{array} \tag{3}$$

where $i = 1, 2, \cdots, n$ and $\mathbf{H}$ is the Hessian operator applied on $\varphi(\cdot)$. The linear system can be rewritten in a matrix form as

$$\mathbf{A}\boldsymbol{\lambda} = \mathbf{y}, \tag{4}$$

where $\boldsymbol{\lambda}$ and $\mathbf{y}$ are $4n$ vectors with the i-th blocks being $[a_i, \mathbf{b}_i]^T$ and $[c, \mathbf{n}_i]^T$ respectively. Here, $\mathbf{A}$ is a $4n\times4n$ coefficients matrix which are assembled from $n\times n$ blocks. Each block $\mathbf{A}_{i,j}$ is a 4×4 sub-matrix corresponding to a pair of RBF centers $(\mathbf{p}_i, \mathbf{p}_j)$.

$$\begin{array}{l} \mathbf{A} = (\mathbf{A}_{i,j})_{n\times n}, \\ \mathbf{A}_{i,j} = \begin{pmatrix} \varphi(\mathbf{p}_i - \mathbf{p}_j) & -(\nabla\varphi(\mathbf{p}_i - \mathbf{p}_j))^T \\ \nabla\varphi(\mathbf{p}_i - \mathbf{p}_j) & -\mathbf{H}\varphi(\mathbf{p}_i - \mathbf{p}_j) \end{pmatrix} \in \Re^{4\times4}. \end{array} \tag{5}$$

In this paper, we use a Wendland's CSRBF [35] as the kernel function because its nice properties in numerical stability and easy-to-implementation.

$$\begin{array}{l} \phi_\rho(r) = \phi(r/\rho) \\ \phi(t) = \begin{cases} (1-t)^4(4t+1), & t \in [0,1], \\ 0, & \text{otherwise}, \end{cases} \end{array} \tag{6}$$

where ρ is the support size, and r is the Euclidean distance between a query point and the center of a RBF. Note that different support sizes can be used at different centers. Solving Eq.(4), an implicit function $f(\mathbf{x})$ can be determined at the space spanned by the supports of centers $\{\mathbf{p}_i\}$. To make the system matrix of RBF interpolation better conditioned, a regularization term with coefficient η is usually added when using RBF interpolants to solve the surface reconstruction problem. That is,

$$(\mathbf{A} + \eta\mathbf{I})\boldsymbol{\lambda} = \mathbf{y}. \tag{7}$$

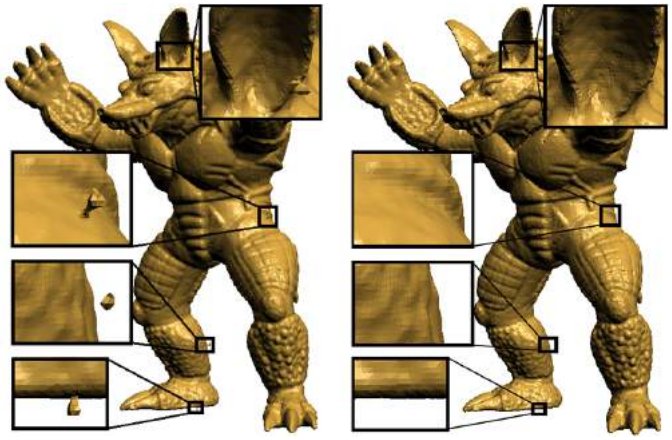

Figure 2: Surface reconstructions by the HRBF implicits without (left) and with (right) the regularization term (η = 1923.1). Artifacts will be produced without the regularization.

An example is shown in Fig.2 to demonstrate the effectiveness of regularization. Benefit about how and why a large value of η can improve the reconstruction can be found in [36].

3. Formulation

This section provides a closed-form formulation for solving the problem of HRBF-based surface reconstruction via quasi-solution. Error-bound of the approximation in quasi-solution is also derived.

3.1. Quasi-solution in closed-form

When increasing the number of centers HRBF-based surface reconstruction, both the numerical instability and the computational cost become intensively remarkable when solving the linear system in Eq.(7). Here, we investigate a closed-form formulation inspired by the theory of quasi-interpolation. Different from the classical quasi-interpolation technique, our method is based on the pattern analysis of matrices. With the help of our formulation, surfaces can be reconstructed in a more stable and efficient way.

Quasi-interpolation techniques can reconstruct a function approximately interpolating a given data set by computing weighted averages of the values at the data points [24]. Specifically, considering an interpolant

$$g(\mathbf{x}) = \textstyle\sum_i \lambda_i \psi_i(\mathbf{x})$$

with constraints on the function values as

$$g(\mathbf{x}_i) = f_i,$$

the function $g(\mathbf{x})$ can be well approximated by letting $\lambda_i \equiv f_i$, where f_i is the function value to be interpolated at $\mathbf{x}_i$. This

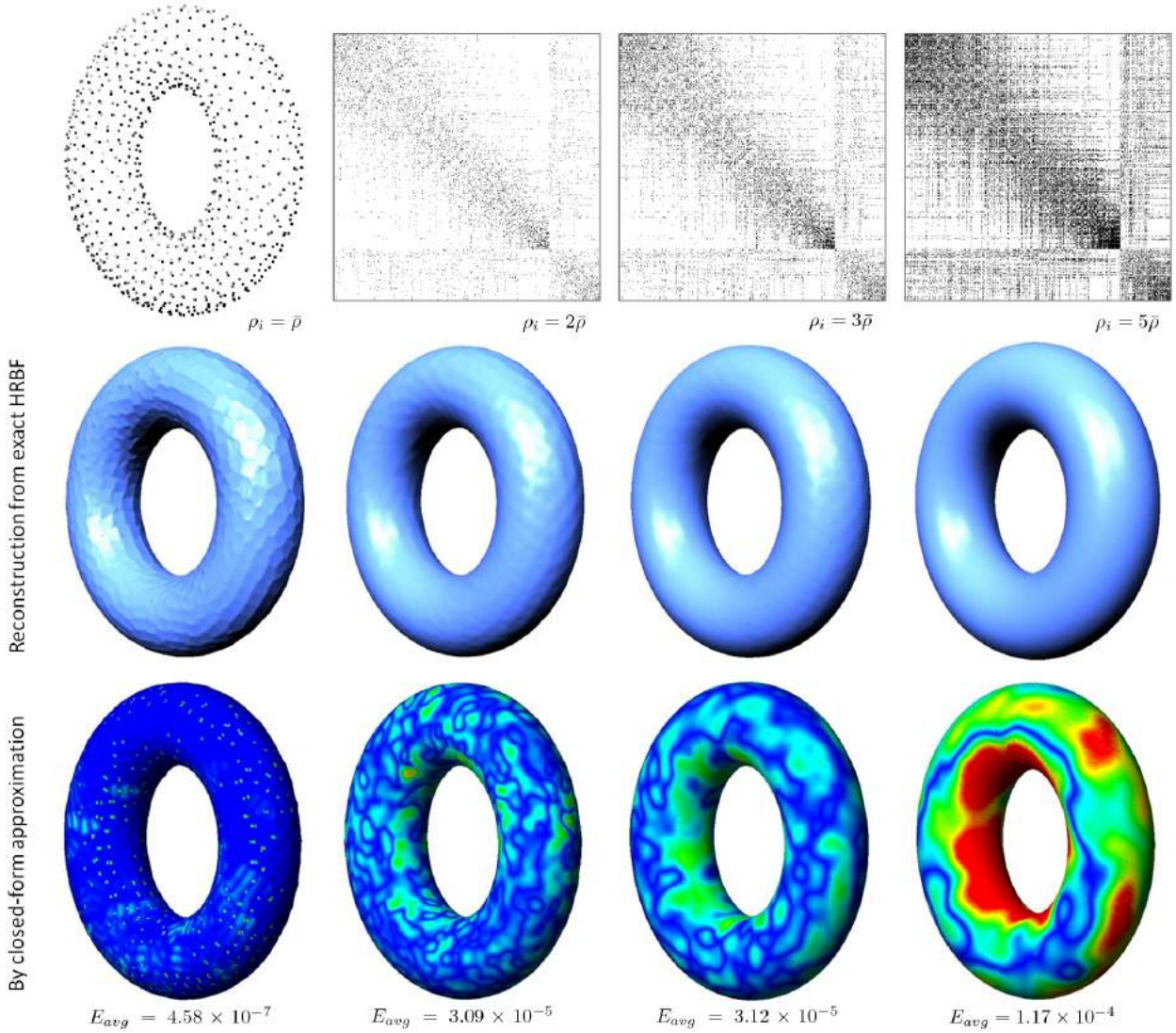

Figure 3: Study on the errors between the coefficient matrix **A** of HRBF implicit and its degenerate diagonal matrix **D**: (top row) the patterns of absolute differences between **A** and **D** when increasing the support sizes – $\bar{\rho} = 0.08$ and only the element with absolute value greater than 10^{-3} are displayed as black dots, (middle row) the results of surface reconstruction by directly solving Eq.(7) with exact **A** and $\eta = 50$, and (bottom row) the reconstruction error by using approximate solution with $\tilde{\boldsymbol{\lambda}}$ (i.e., Eq.(9)). Note that, the reconstruction results generated by our method are visually similar to the results shown in the middle row. Therefore, color maps are employed in the bottom row to illustrate the geometric errors between the exact solution and the approximate solution on the models. The average shape approximation errors are also reported as E_{avg}, which is evaluated by the publicly available Metro tool [37].

is called quasi-interpolation, which can lead to accurate approximation when the trial functions $\psi_i(\mathbf{x})$ satisfy the properties of non-negativity, partition-of-unity and locality/sparsity (ref. [28]). An approximation $\tilde{g}(\mathbf{x})$ of $g(\mathbf{x})$ can be obtained as

$$\tilde{g}(\mathbf{x}) = \textstyle\sum_i f_i \psi_i(\mathbf{x}).$$

More details of approximate date interpolation can be found in prior work (ref. [24, 25, 26, 27, 28]). However, the quasi-interpolation technique cannot be directly applied here as our interpolation constraints consist of both the values and the gradients of functions (see Eq.(2)).

Basically, we need a closed-form formulation to approximate the solution of Eq.(7). Considering from the aspect of matrix computation, the quasi-interpolant with $\lambda_i \equiv f_i$ is actually a solution when the coefficient matrix is approximated by an identity matrix **I**. Inspired by this observation, we further analyze the pattern of matrices generated by the HRBF implicits. Specifically, we study the similarity of the coefficient matrix **A** and its degenerate diagonal matrix. For a CSRBF $\varphi_i(\cdots)$, when there is no other center falling into the space spanned by its support ρ_i, the coefficient matrix is degenerated from the general form $\mathbf{A}_{i,j}$ of Eq.(5) into

$$\mathbf{D}_{i,j} = \begin{cases} diag(1, \frac{20}{\rho_i^2}, \frac{20}{\rho_i^2}, \frac{20}{\rho_i^2}), & (i = j) \\ 0. & (i \neq j) \end{cases} \tag{8}$$

If the scenario of not containing other centers happens at all CSRBF kernels, the linear system in Eq.(7) degenerates into $(\mathbf{D} + \eta\mathbf{I})\boldsymbol{\lambda} = \mathbf{y}$ with $\mathbf{D} = (\mathbf{D}_{i,j})_{n\times n}$. As the coefficient matrix becoming a diagonal matrix, this results in a direct solution of

Eq.(7)

$$\begin{aligned}\tilde{\boldsymbol{\lambda}} &= (\mathbf{D}+\eta\mathbf{I})^{-1}\mathbf{y} \\ &= \{\tfrac{c}{1+\eta}, \tfrac{\rho_1^2\mathbf{n}_1}{20+\eta\rho_1^2}, \cdots, \tfrac{c}{1+\eta}, \tfrac{\rho_n^2\mathbf{n}_n}{20+\eta\rho_n^2}\}.\end{aligned} \tag{9}$$

The criterion of not containing other centers happens at all CSRBF kernels is easy to be satisfied by using a very small support size on all kernels (i.e., $\rho_i = \frac{1}{2}\bar{\rho}$). However, as the supports of kernels are separated from each other in such a case, the implicit surface represented by $f(\mathbf{x}) = 0$ consists of many isolated fragments. In practice, we need to enlarge support sizes at kernels to make them connected so that continuous surfaces can be reconstructed.

When enlarging the support sizes of kernels, the difference between $\mathbf{A}$ and $\mathbf{D}$ increases progressively. However, as shown in Fig. 3, approximating $\mathbf{A}$ by $\mathbf{D}$ can still generate good reconstructions. We also study the patterns of absolute difference between $\mathbf{A}$ and $\mathbf{D}$ in Fig. 3. The errors increase in very small values – see both the reported average error E_{avg} and the color maps that reflect the distribution of geometric errors. Basically, no visual difference can be found between the exact and the approximate solutions. And the average value of shape approximation error, E_{avg}, is also very small. Moreover, when a larger regularization coefficient is employed, the geometric approximation error can be further reduced as the difference between $\mathbf{A}$ and $\mathbf{D}$ becomes more trivial in this case. For example, the values of E_{avg} on all cases shown in Fig. 3 drop below 10^{-6} when the value of η is changed from 50 to 1000. The error of $\boldsymbol{\lambda}$ caused by using $\mathbf{D}$ to replace $\mathbf{A}$ will be analyzed in the section below. The approximate error can be bounded as long as the number of centers falling in the support of each kernel is bounded.

Using $\mathbf{D}$ to approximate $\mathbf{A}$ amounts to letting $\boldsymbol{\lambda} \simeq \tilde{\boldsymbol{\lambda}}$. Then, a closed-form function that approximates $f(\mathbf{x})$ can be obtained. Substituting $\tilde{\boldsymbol{\lambda}}$ of Eq.(9) as $\boldsymbol{\lambda}$ into Eq.(7), the coefficients of the i-th basis function can be approximated by

$$a_i \approx 0 \quad \text{and} \quad \mathbf{b}_i \approx \frac{\rho_i^2}{20+\eta\rho_i^2}\mathbf{n}_i,$$

which give an approximate function of $f(\mathbf{x})$ as

$$\tilde{f}(\mathbf{x}) = -\sum_{j=1}^{n}\langle\frac{\rho_j^2}{20+\eta\rho_j^2}\mathbf{n}_j, \nabla\varphi(\mathbf{x}-\mathbf{p}_j)\rangle. \tag{10}$$

By this implicit function, we can tessellate the isosurface of $\tilde{f}(\mathbf{x}) = 0$ into a polygonal mesh by a variant of *Dual-Contouring* (DC) algorithm [38].

3.2. Error-bound analysis

Error between the quasi-solution $\tilde{\boldsymbol{\lambda}}$ and the exact solution $\boldsymbol{\lambda}$ of Eq.(7) must be bounded to make the closed-form formulation useful. The following lemmas are derived, the proofs of which can be found in [39].

Lemma 1 Defining $\Delta\mathbf{A} = (\mathbf{A}+\eta\mathbf{I}) - \mathbf{D}$ and $\Delta\boldsymbol{\lambda} = \boldsymbol{\lambda} - \tilde{\boldsymbol{\lambda}}$, the error of approximation is bounded as

$$\|\Delta\lambda\|_\infty \le \frac{\|\mathbf{D}^{-1}\|_\infty\|\Delta\mathbf{A}\|_\infty}{1-\|\mathbf{D}^{-1}\|_\infty\|\Delta\mathbf{A}\|_\infty}\|\mathbf{D}^{-1}\mathbf{y}\|_\infty \tag{11}$$

when

$$\|\mathbf{D}^{-1}\|_\infty\|\Delta\mathbf{A}\|_\infty < 1. \tag{12}$$

End users may not know about how to make $\|\mathbf{D}^{-1}\|_\infty\|\Delta\mathbf{A}\|_\infty < 1$ to achieve bounded errors. Thus, the following lemma is derived. Without loss of generality, we can assume that there are at most m other centers falling in the support region for each kernel. Then, the error-bound of our quasi-solution can be achieved on the Wendland's CSRBFs when the values of minimal support size $\rho_{\min}$, m and η satisfy the condition given below.

Lemma 2 When Wendland's CSRBFs are used, if their support sizes $\rho_i \in [\rho_{\min}, \sqrt{20}]$ and each support region contains at most m centers of other CSRBFs, the value of $\|\Delta\lambda\|_\infty$ is bounded by a constant when

$$1+\eta > m(\frac{5}{4\rho_{\min}} + \frac{35}{\rho_{\min}^2}). \tag{13}$$

Here we also limit the value of maximal support size (i.e., $\rho_i \le \sqrt{20}$) to obtain a compact formula in Eq.(13). Otherwise, different formulas for the right part of Eq.(13) will be obtained for ρ_i falling into different range of values.

Remark. The requirements of,

1. all the CSRBFs have their support sizes within the interval $[\rho_{\min}, \sqrt{20}]$,
2. there are at most m centers falling in the support of any others CSRBF, and
3. the values of η, m and $\rho_{\min}$ satisfy Eq.(13),

can be achieved by a carefully designed support size tuning algorithm. After giving the weight of regularization η, the values of ρ_is and m are adjusted to satisfy Eq.(13) (details can be found in Section 4.1). By scaling all models into a bounding box of $[-1,1]^3 \in \Re^3$, the algorithm can also ensure $\max\{\rho_i\} < \sqrt{20}$. Approximation errors on different examples can be found in Fig.4 and will also be discussed in Section 5.2.

4. Reconstruction Algorithm

By using the implicit function $\tilde{f}(\mathbf{x})$ defined in Eq.(10), surface can be efficiently reconstructed from a set of points $\mathcal{P}$ and their normals $\mathcal{N}$. Our algorithm consists of two major steps:

1. *Support size tuning* – a scheme is investigated to determine the support sizes of kernel functions according to *Lemma 2* to guarantee the existence of error-bound.
2. *Mesh extraction* – a polygonal mesh will be extracted from the zero isosurface as the result of surface reconstruction.

The second step takes the majority of computation time in our algorithm. We develop an efficient method that fits well with the architecture of parallel computing to efficiently run on computers with multi-cores.

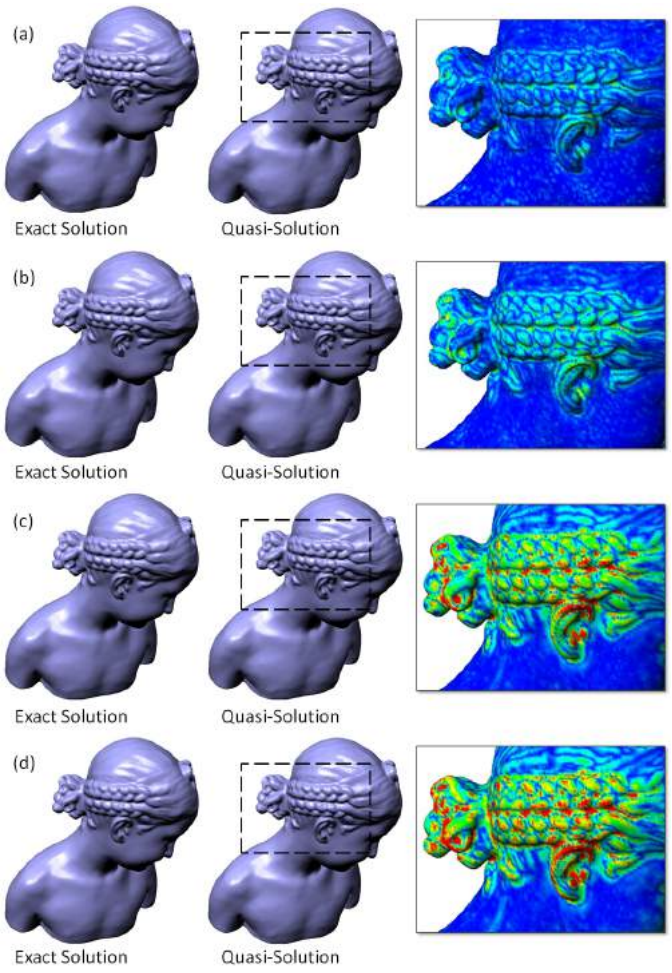

Figure 4: Examples with (a & b) bounded reconstruction vs. (c & d) unbounded reconstructions, where the value of η is scaled from a value suggested in Eq.(15), η_0. The values of η/η_0 are (a) 1.0, (b) 0.5, (c) 0.1 and (d) 0.01. Color maps illustrate the geometric error between the exact and the quasi solutions measured by the Metro tool [37]. Statistics can be found in Table 1. In all cases, $m = 56$ and $\rho_{\min} = 0.75\bar{d} = 0.0309$ are applied.

4.1. Support size tuning

In *Lemma 2*, the formula of inequality consists of three parameters, where η is a user input coefficient to specify the level of regularization. Usually, we do not modify the value of η. According to the properties of quadratic equation, Eq.(13) actually gives the bound of $\rho_{\min}$'s value as

$$\rho_{\min} > \frac{5m + \sqrt{25m^2 + 2240(1+\eta)}}{8(1+\eta)}. \tag{14}$$

The formula is derived based on the assumption that $m, \eta, \rho_{\min} > 0$. When the value of m is known, this gives the minimal value of support size on each kernel. However, a large support size may let a kernel cover more than m centers of other RBFs. Basically, each support must cover enough number of data points to generate a span of the local shape; meanwhile, it cannot be too large.

To resolve this contradiction, we develop an iterative algorithm to determine the support sizes at all CSRBF kernels are determined by three major steps.

Table 1: Errors of Bounded vs. Unbounded Reconstruction

Fig.	η/η_0	Right Side of Eq.(14)	Geometric Error	
			Maximal	Average
(a)	1	0.0186	1.15×10^{-3}	5.47×10^{-5}
(b)	0.5	0.0265	1.77×10^{-3}	7.54×10^{-5}
(c)	0.1	0.0887	3.07×10^{-2}	1.64×10^{-4}
(d)	0.01	0.219	3.08×10^{-2}	1.98×10^{-4}

1. The algorithm starts from analyzing the density of points to determine the initial value of support sizes. An octree is constructed to split the input points into different nodes by keeping similar number of points in each leaf-nodes. Denoting the average diagonal length of the leaf-nodes as $\bar{d}$, the support sizes are temporally set as $\frac{3}{4}s\bar{d}$ with s being an amplifier – $s = 1.0$ is employed for noise-free data sets.
2. We then count the number of points covered by the supporting region of each CSRBF, $\phi_{\rho_j}(\|\mathbf{x}-\mathbf{p}_j\|)$. The maximal number is assigned as m.
3. After knowing m, the temporal support size of each CSRBF kernel is enlarged by an incremental procedure until it covers m centers of other kernels. Among all the support sizes determined in this way, the minimal one is selected as the final support size applied to all kernels (i.e., uniform support size $\rho_{\min}$ is adopted).
4. When the value of $\rho_{\min}$ does not satisfy the inequality of Eq.(14), we go back to step 3 with $m = m - 1$ to run the support size tuning again.

For users have no special demand on the degree of regularization, we suggest to use

$$\eta = \frac{100}{(0.75\bar{d})^2}, \tag{15}$$

which empirically balances the robustness and the quality of reconstruction. According to our experimental tests, Eq.(14) can always be satisfied when the value of η is chosen in this way. However, if an inappropriate value is used for η – e.g., a very small η used in Fig.4, our algorithm cannot find a value of $\rho_{\min}$ satisfying Eq.(14) even when $m = 1$. In such a scenario, we choose m and $\rho_{\min}$ as the values determined by the first iteration of our algorithm. This usually leads to a reconstruction with larger approximation error (see the color map in Fig.4).

Support sizes generated by the above method work well on clean data but may fail in highly noisy input. In such a scenario, we enlarge the temporal support sizes by using $s > 1.0$ to enhance the effectiveness of denoising. This results in a larger m. More geometric details can be preserved with a smaller support size while a larger support size leads to a smoother reconstruction. Examples using different values in the amplifier s can be found in Fig.5.

4.2. Efficient mesh generation

To extract the isosurface $\tilde{f}(\mathbf{x}) = 0$ of an implicit function $\tilde{f}(\mathbf{x})$, the most popular algorithms in literature are *Marching*

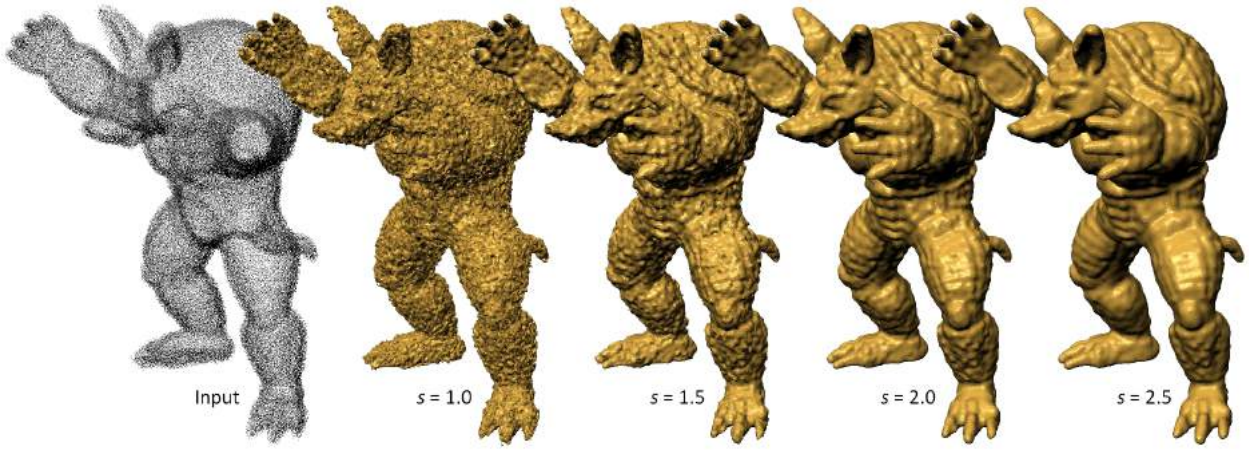

Figure 5: Results with different levels of denoising can be obtained by using different amplifiers on a noisy point set: (from left to right) input noisy points and our reconstruction results. The value of η suggested by Eq.(15), $\eta = 2416.24$, is used here.

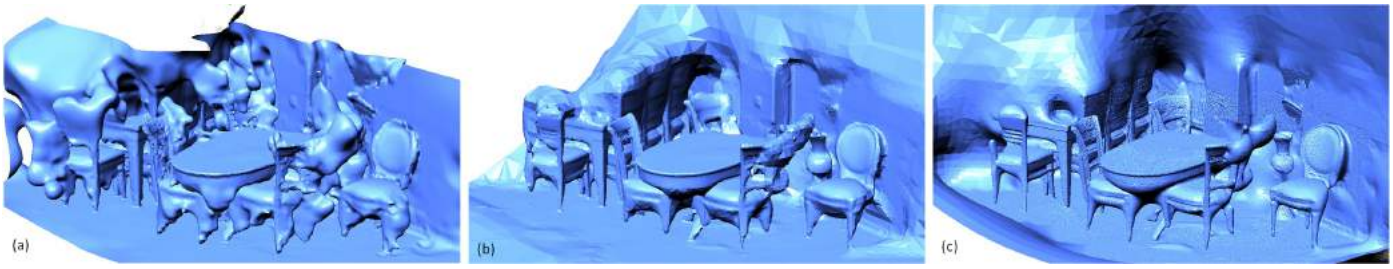

Figure 6: Prior methods generate unsatisfactory results on the incomplete set of indoor scan: (a) MPU [21], (b) SSD [19] and (c) Screened-Poisson [18].

Cubes (MC) [40] and *Dual-Contouring* (DC) [38]. DC algorithm in general produces fewer polygons than MC while generating surfaces with similar quality. We develop a variant of DC algorithm to extract zero isosurfaces from the regions spanned by the RBFs in Eq.(10).

The reconstruction with compactly supported implicit functions leads to open mesh surfaces and leaves holes in regions not covered by the supports of RBFs, where $\tilde{f}(\mathbf{x})$ returns an undefined value. This is very useful for reconstructing the scenes that have not been completely captured (e.g., the scene shown in Fig.1). Moreover, as $\tilde{f}(\mathbf{x})$ is defined in a closed-form, its evaluation (therefore also the isosurface extraction) is highly scalable and can be efficiently performed by local operations running in parallel. Another potential benefit of our approach is that the surface can be reconstructed in an out-of-core manner, which is hard for a hierarchical structure. Similarly, progressive reconstruction can be developed by only updating a local region when new points are added as centers of CSRBFs.

5. Results and Discussion

A surface reconstruction algorithm based on the closed-form formulation of HRBF implicits has been implemented with Microsoft Visual C++ and OpenGL. We evaluate our methods on a PC with two Intel Core i7-2600K CPUs at 3.4GHz plus 16GB RAM. Our approach has been applied to various data with up to 14M points, where a surface with 723k triangles can be reconstructed in 26.5 seconds. Comparisons with other approaches are given below to verify the quality and the speed of our method. In our tests, all models have been scaled into a bounding-box of $[-1, 1]^3 \in \mathfrak{R}^3$.

5.1. Raw data

In practice, data obtained from an industrial acquisition process is usually a large point set with noise. The data-sets are also incomplete in most cases (e.g., the indoor scan shown in Fig.1 and the models shown in Fig.7). The results of reconstruction are unsatisfactory when using the state-of-the-art surface reconstruction techniques developed for clean sets of complete data points, including the *Multiple Partition of Unity* (MPU) reconstruction [21], the *Smooth Signed Distance* (SSD) reconstruction [19] and the *Screened-Poisson* reconstruction [18]. See Fig.6 for an example. Large errors are generated in regions of missing data. Similar errors can be found when applying these methods for the data sets tested in Fig.7. In these tests, we employ a depth 10 octree in the SSD and Screen-Poisson methods to generate results. For MPU and our method, we adjust the resolutions of polygonization methods to extract meshes with similar numbers of triangles as SSD and Screened-Poisson. The parameter Max_Error of MPU is set as 0.01 times of the model's size. Default values are used for other parameters.

The recently developed FSSR method [5] targets on fast surface reconstruction from large real data sets with missing regions. We compare our method with FSSR on two sets of real scanned data in Fig.7. Fuhrmann and Goesele [5] assumed the

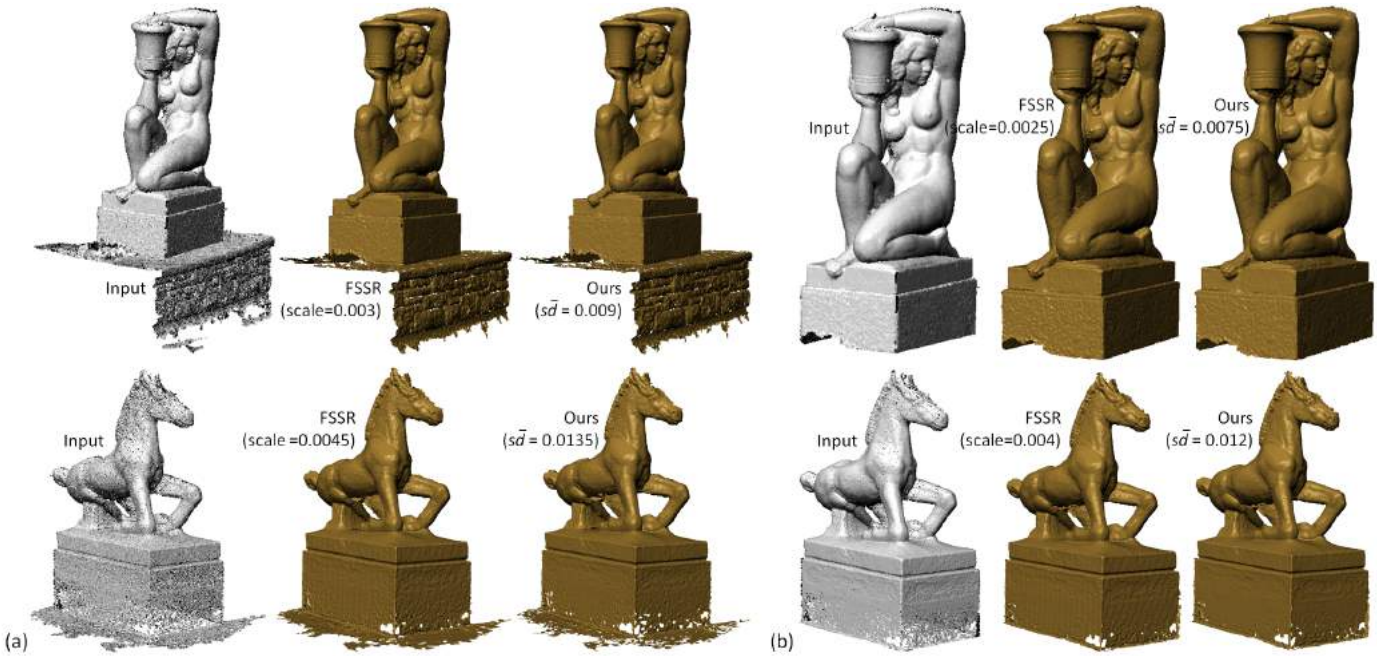

Figure 7: Examples of surface reconstruction on incomplete set of points: (a) reconstruction from raw data and (b) reconstruction from data sets processed by the consolidation method [41]. Our results are comparable with that obtained by FSSR but our method is about 30× faster.

scale of an input point set is known, which however is not the case here. In order to make an appropriate comparison, we use $1/3$ of the average support size determined by our method as the scale used in FSSR. This is consistent with the formulation presented in [5], where the support size is set as three times of the input scale. The actual values of scale and $s\bar{d}$ are also given in Fig.7. It can be found that similar number of triangles are generated in FSSR and our method by setting the value of scale in this way.

Table 2 gives the statistics of tests on these models. Due to the closed-form formulation, our method does not need any global operation such as solving a large linear system. Its computational time is only spent on constructing an octree to computing the support size and the step of function value evaluation in iso-surface extraction. Both SSD and Screened-Poisson need to solve linear systems globally. In Poisson reconstruction, the multi-grid solver performs a constant number of conjugate-gradient iterations at each level, which gives linear complexity w.r.t to the number of nodes in the octree. The SSD reconstruction uses conjugate-gradients to determine all the coefficients simultaneously, which has a complexity of $O(n^{1.5})$. In MPU reconstruction, only local fitting is taken at leaf-nodes of an octree. These surfaces are blended together to form the resultant surface, which is fast but still slower than our method. On the other aspect, as both FSSR and our method do not need any global operations during the computation of surface reconstruction, it can easily be parallelized on the PC with multi-cores – OpenMP is used in our implementation. We test both approaches on a PC with 8-cores. As shown in the computational statistics in Table 2, the program can be effectively speed up on 8-cores. The multi-core version of FSSR is provided by the authors on their homepage. When generating meshes with similar number of triangles, our method is up to 30× faster than FSSR. In summary, our method is the fastest method and can generate similar results as the other methods.

We also study the effectiveness of our approach on the benchmark of FSSR with shape density variation caused by superposing point sets obtained from multiple scans. As shown in Fig.8, the input set from four synthetic scans is downloaded from the homepage of FSSR's authors. We reconstruct mesh surfaces with different resolutions on this data set to verify the performance of our method. By default, an octree with 7 levels is employed in FSSR to generate the resultant mesh surface. It has been changed to 9 levels in our tests to generate more triangles on the resultant surface. When producing meshes with similar number of triangles, our reconstruction is similar to FSSR.

5.2. Verification of numerical and geometric errors

Error-bound of the quasi-solution $\tilde{\boldsymbol{\lambda}}$ with reference to the exact solution $\boldsymbol{\lambda}$ has been derived in Section 3.2. It is also interesting to study the error between $\tilde{\boldsymbol{\lambda}}$ and $\boldsymbol{\lambda}$ in experimental tests. We measure $\|\tilde{\boldsymbol{\lambda}} - \boldsymbol{\lambda}\|_\infty$ in examples shown above and the results are listed in Table 3. The geometric error between the surfaces obtained from exact and quasi-solutions is also measured by the Hausdorff distance w.r.t. the diagonal length of a model's bounding box (see the last column of Table 3). From the statistics, it is easy to find that our quasi-solution provides results having small geometric difference to the exact solution.

The quality of scalar field in terms of interpolation can be controlled by the error-bound in our reconstruction. As normal approximation is an important characteristic of HRBF interpolant, we study the differences between the input normals and the gradients of a reconstructed field, $\nabla\tilde{f}(\mathbf{x})$. As shown in Fig.9, a subset of data points are sampled from the point cloud. Both the input normals (in red) and $\nabla\tilde{f}(\mathbf{x})$ (in blue) are displayed to illustrate the differences. It can be observed that the gradients

Table 2: Computational performance of different reconstruction approaches†

Model	Pnt. #	MPU Trgl. #	MPU Time (Sec.)	SSD Trgl. #	SSD Time (Sec.)	SPR Trgl. #	SPR Time in Sec. 1-cor.	SPR Time in Sec. 8-cor.	FSSR Trgl. #	FSSR Time in Sec. 1-cor.	FSSR Time in Sec. 8-cor.	Our Method Trgl. #	Our Method Time in Sec. 1-cor.*	Our Method Time in Sec. 8-cor.
Indoor	922.0k	313k	8.4	315k	19.1	319k	19.0	13.3	319k	470.6	98.4	313k	17.1 (16.0)	5.5
Aquar.	253.9k	368k	3.6	366k	12.3	369k	13.3	6.9	350k	407.3	89.6	375k	8.0 (7.8)	2.7
Horse	239.8k	242k	2.6	248k	6.0	249k	10.1	5.3	241k	262.3	56.2	245k	5.4 (5.2)	1.8

*Note that, the time reported here includes both the surface reconstruction and the mesh extraction, and the time shown in the bracket is the time for the isosurface extraction (i.e., field value evaluation and mesh generation). '1-cor.' and '8-cor.' stand for the number of cores.
†To have a fair comparison, similar number of triangles are generated for different approaches.

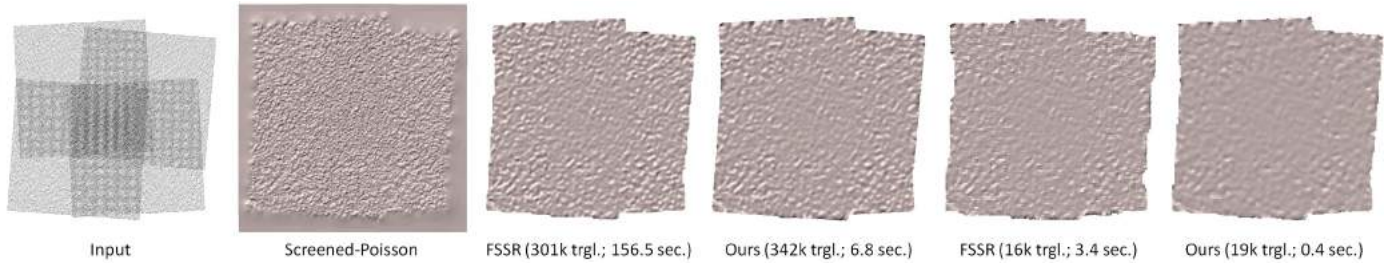

Figure 8: When processing an input with significant density variation – e.g., from four synthetic scans (most-left), FSSR (with scale=0.0105) and ours (with $s = 1.0$) can avoid generating unwanted artifacts caused by high frequency noise. The computation time for generating meshes in different resolutions has been reported, where both FSSR and ours are tested on a CPU with eight-cores. An octree with 7 levels (default value) is employed in FSSR to generate the result with 16k triangles, and it is set to have 9 levels to produce a result with 301k triangles.

Table 3: Error Statistics of Quasi-Solution

Model*	$\eta/10^3$	$\|\lambda\|_\infty$	$\|\tilde{\lambda} - \lambda\|_\infty$	Err.
Armadillo	2.42	4.58×10^{-4}	1.74×10^{-4}	0.22%
Bimda	1.05	6.87×10^{-4}	3.29×10^{-4}	0.14%
Aquarius	7.64	1.46×10^{-4}	7.26×10^{-5}	0.53%
Horse	5.39	2.04×10^{-4}	1.21×10^{-4}	0.39%

*We do not include the example of indoor scan in this statistic as the exact reconstruction runs out of memory when solving the huge system of linear equations with $4 \times 922k$ unknown variables.

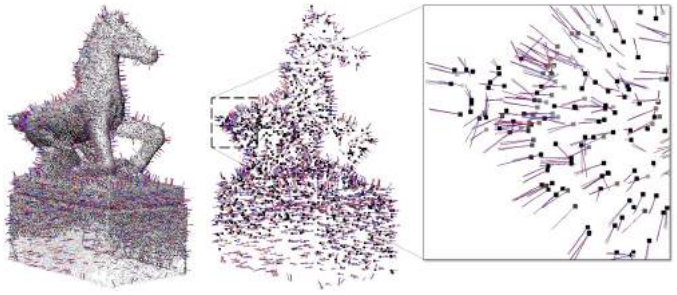

Figure 9: An example for displaying the deviation of gradients (in blue) on the reconstructed field w.r.t. input normals (in red). To have a better visualization, the normal and gradient vectors are only displayed on 1/10 of the original points.

of field are quite consistent with the input normals. Quantitatively, the average/maximal angular errors between $\nabla\tilde{f}(\mathbf{p}_i)$ and $\mathbf{n}_i$, $\theta(\nabla\tilde{f}(\mathbf{p}_i), \mathbf{n}_i)$, are measured and listed in Table 4. The errors of position interpolation are also evaluated by the distances between input points and the reconstructed mesh surfaces – denoted by $d(\mathbf{p}_i, S)$. From the statistics on angular errors, we can observe relative large variation on raw data with noise, but a smaller error (both Max. and Avg.) on a clean data set – Bimda. This is because that the reconstructed surface must compromise to the imperfect positions/orientations given on noise.

5.3. Noisy data

In the following tests, we verify the robustness of our approach on input with noise at different levels. For a given point set $\mathcal{P}$ with normal vectors $\mathcal{N}$, if the diagonal length of its bounding box is d, a new point set with $\delta\%$ Gaussian noise is obtained as follows:

- Select $\delta\%$ points from $\mathcal{P}$ to add into a sub-set $\mathcal{G}$;
- Randomly generate a set of scales with Gaussian distribution, $\mathcal{D}_G = \{d_i\}$ $(d_i \in [0, \delta d/1000])$;
- Impose the noise onto the points in $\mathcal{G}$ by $\mathbf{p}'_j = \mathbf{p}_j + d_j\mathbf{n}_j$ for all $\mathbf{p}_j \in \mathcal{G}$.

Normal vectors on a set of noisy points are re-generated by the orientation-aware *Principal Component Analysis* (PCA) [7] with p-nearest neighbors – here $p = 6$ is used in all tests.

We reconstruct mesh surfaces from a filigree model with 30% and 60% Gaussian noise by different methods, including Screen-Poisson, FSSR and ours (see Fig.10). The noise-free set of points are sampled from a mesh model so that we can compare the results of reconstruction with the original mesh to evaluate the shape approximate errors generated by different methods. Screen-Poisson reconstruction does not perform well on noisy models. As shown in Fig.10, models with incorrect

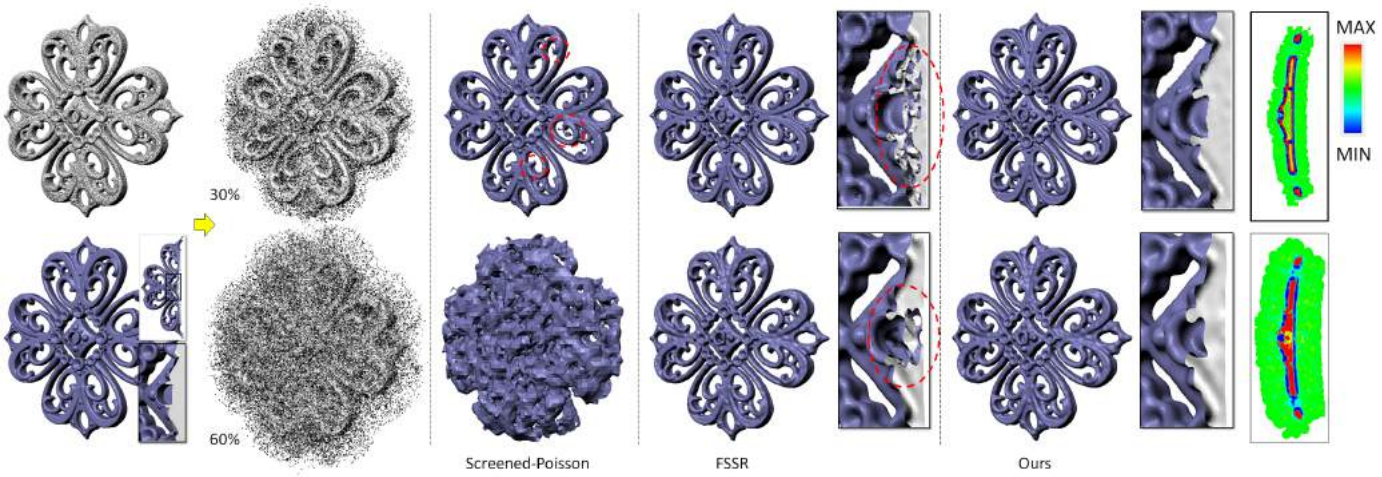

Figure 10: Examples of reconstruction from sets (having 250k points) with different level of Gaussian noise. Topological errors can be found on the results of Screen-Poisson and FSSR (see the regions circled by dashed lines in red). The cross-sectional view of function values in our reconstruction has also been given in the right, where the regions in white color have undefined function values.

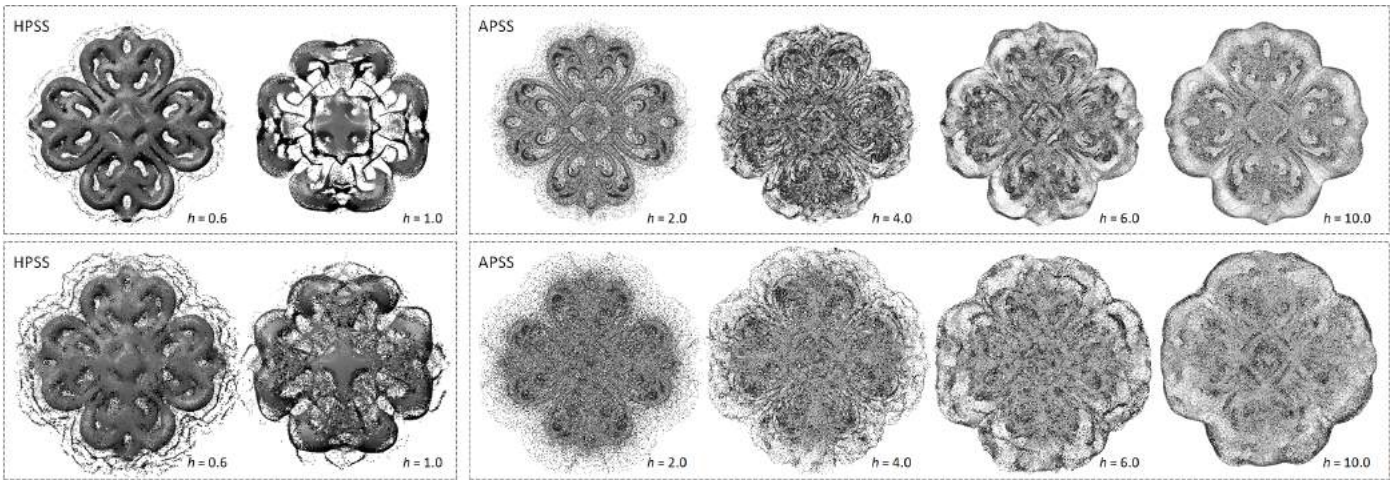

Figure 11: Results of MLS-based reconstruction from highly noisy points shown in Fig.10: (top row) the results from an input with 30% Gaussian noise and (bottom row) the results on points with 60% Gaussian noise. Different values are tried for the support scaling factor, h, on both HPSS [34] and APSS [30].

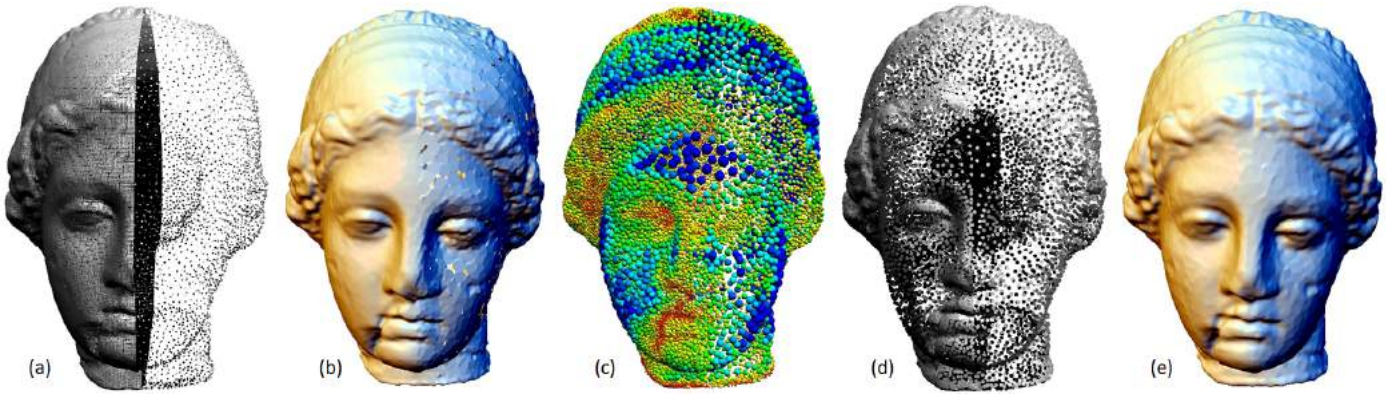

Figure 12: Adaptive HRBF implicits are generated by our method with the help of center selection [39]: (a) the input set with 100, 371 points in high non-uniformity, (b) the reconstruction using all points as centers of HRBF implicits will lead to holes in the sparse regions, (c) the spherical cover – the spheres are displayed in radii as 1/4 of the real ones, (d) the selected 13, 446 centers of RBFs, and (e) reconstruction from the selected centers – no hole will be generated as the densities of centers in the left and the right are similar to each other.

Table 4: Statistics of Angular and Position Errors[†]

Model	$\lvert\theta(\nabla\tilde{f}(\mathbf{p}_i), \mathbf{n}_i)\rvert$		$\lvert d(\mathbf{p}_i, S)\rvert$	
	Max.	Avg.	Max.	Avg.
Indoor	179.56°	5.34°	0.016	9.6×10^{-4}
Armadillo[†]	176.60°	15.66°	0.023	7.5×10^{-3}
Bimda	33.69°	1.53°	0.0041	2.1×10^{-4}
Aquarius	136.57°	9.87°	0.013	3.8×10^{-4}
Horse	167.12°	3.64°	0.014	5.3×10^{-4}

[†]Large average errors are found on the noisy Armadillo in Fig.5.
[*]All models have been scaled into a bounding box of $[-1, 1]^3 \in \Re^3$.

Table 5: Error Statistics of Reconstruction on Noisy Input

FSSR on the Filigree Model (Fig.10)					
Noisy	**Forward Dist.**		**Backward Dist.**		
Level	**Max.**	**Ave.**	**Max.**	**Ave.**	**Scale**
10%	.00360	.000295	.0175	.00315	.00375
30%	.00800	.00600	**.0325**	.00140	.00670
60%	.0115	.00145	**.0300**	.00170	.0112
Our Method on the Filigree Model (Fig.10)					
Noisy	**Forward Dist.**		**Backward Dist.**		
Level	**Max.**	**Ave.**	**Max.**	**Ave.**	s
10%	.00405	.000255	.0115	.000275	1.9
30%	.00700	.000800	.00700	.000800	2.7
60%	.0135	.00210	.0150	.00220	3.5

[*]The errors are measured by the Metro tool [37].

topology are generated. FSSR and ours can still reconstruct acceptable models even after embedding 60% Gaussian noise. We then compare these two methods in terms of shape approximation error by the Metro tool [37] (see Table 5). For the measurements based on forward distances from ground-truth to the reconstruction, FSSR has slightly smaller errors. In the errors based on backward distances (i.e., from reconstruction to ground-truth), our method outperforms FSSR. This is because FSSR generates some interior isolated regions (i.e., topological errors) but our method does not – see the zoom-view in Fig.10. Moreover, our method is 17.5× and 36.4× faster than FSSR on the 30% and 60% noisy models respectively.

Similar to *Moving Least-Squares* (MLS) based methods for point-sampled geometry, the implicit function for surface reconstruction is constructed in our approach by weighted sum of data points. However, different from the MLS methods, enforcement on the gradients of RBF functions strengthens the robustness of our method. When applying the normal interpolated – HPSS [34] or approximated – APSS [30] methods to the highly noisy input, results with poor quality are observed (see Fig.11). The results of HPSS are based on our implementation, and we use the plug-in of MeshLab to generate the results of APSS. Different scaling factors have already been tested to achieve the best results.

5.4. Limitation

The limitations of our approach are mainly caused by the nature of locally compact support of our method. Therefore, we share the following common limitations as the FSSR method.

Figure 13: Reconstructions from an input with noises and outliers: (left) input point cloud, (middle) the reconstructed surface before post-processing and (right) the resultant surface after mesh cleaning.

- Near the boundary of regions with function-value defined, some small fragments isolated from the main reconstruction could be formed by the numerical oscillation. Such isolated fragments must be removed by a post-processing step taken on the mesh surface after polygonization (see Fig.13 for an example).

- Although reconstruction with high quality can be found at the example shown in Fig.7, misaligned multiple scan could lead to multi-layers of points, which produce multiple surface layers at those 'overlapped' regions.

- Lastly, holes are easily formed in the sparse region when the input point set has high non-uniformity.

To overcome the last limitation, an optional step can be applied to select a subset of input points to serve as the centers of kernels to obtain a better surface reconstruction (see the illustration shown in Fig.12). A method akin to the spherical covering method proposed in prior work [42] is used here, and more details can be found in [39]. An alternative is the adaptive center selection method employed in [43]. Besides, as stated in Section 4.1 that the support size selection algorithm cannot be theoretically proved to converge, it could (although rarely happens) result in a reconstruction without error-bound.

6. Conclusion

In this paper, we present a novel surface reconstruction method based on computing an approximate solution of HRBF-based implicit surface fitting. The approximate solution is formulated as a weighted sum of compactly supported basis functions centered at input data points equipped with normal vectors. We provide a closed-form solution without any global operation. As a result, the implicit function for surface reconstruction can be evaluated efficiently and robustly. Experimental results have shown the performance of our approach by comparing to the state-of-the-arts.

No global operation needs to be applied during the surface reconstruction of our approach. As a result, it is easy to extend our implementation to run in the out-of-core manner or on a distributed PC-cluster. We would like to further investigate the strength of our method in this aspect in our future work, which can make it possible to realize the on-site reconstruction of large-scale 3D models (e.g., outdoor scenes like city scale). Many robotics and virtual reality applications could benefit from this work.

Acknowledgment

We would acknowledge the CGG Laboratory of EPFL and the authors of MPU, SSD, SPR, and FSSR for sharing the data of point clouds and the source code. This research was partially funded by NSFC (No. 61572527), the Program for New Century Excellent Talents in University (No. NCET-13-0590), Hunan Province Science and Technology Project (No. 2014FJ2008), and the Open Research Fund of Key Laboratory of High Performance Complex Manufacturing (No. Kfkt2015-05) at Central South University.

References

[1] J. Carr, R. Beatson, J. Cherrie, T. Mitchell, W. Fright, B. McCallum, T. Evans, Reconstruction and representation of 3d objects with radial basis function, in: Proceedings of ACM SIGGRAPH 2001, 2001, pp. 67–76.

[2] G. Turk, J. F. O'Brien, Modeling with implicit surfaces that interpolate, ACM Transactions on Graphics 21 (4) (2002) 855–873.

[3] H. Wendland, Scattered data approximation, Cambridge University Press, Cambridge, 2005.

[4] I. Macedo, J. P. Gois, L. Velho, Hermite radial basis functions implicits, Computer Graphics Forum 30 (1) (2011) 27–42.

[5] S. Fuhrmann, M. Goesele, Floating scale surface reconstruction, ACM Trans. Graph. 33 (4) (2014) 46:1–46:11.

[6] K. Xu, H. Huang, Y. Shi, H. Li, P. Long, J. Caichen, W. Sun, B. Chen, Autoscanning for coupled scene reconstruction and proactive object analysis, ACM Transactions on Graphics 34 (6).

[7] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle, Surface reconstruction from unorganized points, in: Proceedings of ACM SIGGRAPH 1992, 1992, pp. 71–78.

[8] B. Morse, T. Yoo, P. Rheingans, D. Chen, K. Subramanian, Interpolating implicit surfaces from scattered surfaces data using compactly supported radial basis functions, in: Shape Modeling International 2001, 2001, pp. 89–98.

[9] Y. Ohtake, A. Belyaev, H. Seidel, 3D scattered data interpolation and approximation with multilevel compactly supported RBFs, Graphical Models 67 (2005) 150–165.

[10] R. Pan, X. Meng, T. Whangbo, Hermite variational implicit surface reconstruction, Science in China Series F: Information Sciences 52 (2) (2009) 308–315.

[11] M. Samozino, M. Alexa, P. Alliez, M. Yvinec, Reconstruction with voronoi centered radial basis functions, in: Proceedings of the fourth Eurographics symposium on Geometry processing, 2006, pp. 51–60.

[12] J. Sussmuth, Q. Meyer, G. Greiner, Surface reconstruction based on hierarchical floating radial basis functions, Computer Graphics Forum 29 (6) (2010) 1854–1864.

[13] Y. Ohtake, A. Belyaev, H. Seidel, Sparse surface reconstruction with adaptive partition of unity and radial basis functions, Graphical Models 68 (1) (2006) 15–24.

[14] I. Tobor, P. Reuter, C. Schlick, Reconstructing multi-scale variational partition of unity implicit surfaces with attributes, Graphical Models 68 (1) (2006) 25–41.

[15] C. Walder, B. Schölkopf, O. Chapelle, Implicit surface modeling with a globally regularised basis of compact support, Computer Graphics Forum 25 (3) (2006) 635–644.

[16] P. Alliez, D. Cohen-Steiner, Y. Tong, M. Desbrun, Voronoi-based variational reconstruction of unoriented point sets, in: Proceedings of Symposium on Geometry Processing, 2007, pp. 39–48.

[17] M. Kazhdan, M. Bolitho, H. Hoppe, Poisson surface reconstruction, in: Proceedings of Symposium on Geometry Processing, 2006, pp. 61–70.

[18] M. Kazhdan, H. Hoppe, Screened poisson surface reconstruction, ACM Transactions on Graphics.

[19] F. Calakli, G. Taubin, SSD: Smooth signed distance surface reconstruction, Computer Graphics Forum 30 (7) (2011) 493–501.

[20] J. P. Gois, V. Polizelli-Junior, T. Etiene, E. Tejada, A. Castelo, L. G. Nonato, T. Ertl, Twofold adaptive partition of unity implicits, The Visual Computer 24 (12) (2008) 1013–1023.

[21] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, H. Seidel, Multi-level partition of unity implicits, ACM Transaction on Graphics 22 (3) (2003) 463–470.

[22] M. Berger, J. A. Levine, L. G. Nonato, G. Taubin, C. T. Silva, A benchmark for surface reconstruction, ACM Trans. Graph. 32 (2) (2013) 20:1–20:17.

[23] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, J. A. Levine, A. Sharf, C. T. Silva, State of the Art in Surface Reconstruction from Point Clouds, in: Eurographics 2014 - State of the Art Reports, 2014.

[24] D. Shepard, A two-dimensional interpolation function for irregularly-spaced data, in: Proceedings of the 1968 ACM National Conference, 1968, pp. 517–524.

[25] M. Buhmann, On quasi-interpolation with radial basis functions, Journal of Approximation Theory 72 (1) (1993) 103–130.

[26] Z. Wu, Z. Xiong, Multivariate quasi-interpolation in $l^p(r^d)$ with radial basis functions for scattered data, International Journal of Computer Mathematics 87 (3) (2010) 583–590.

[27] X. Han, M. Hou, Quasi-interpolation for data fitting by the radial basis functions, in: GMP'08: Proceedings of Geometric Modeling and Processing, 2008, pp. 541–547.

[28] S. Liu, C. Wang, Quasi-interpolation for surface reconstruction from scattered data with radial basis functions, Computer Aided Geometric Design 29 (7) (2012) 435–447.

[29] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, C. Silva, Point set surfaces, in: Proceedings of the conference on Visualization'01, 2001, pp. 21–28.

[30] G. Guennebaud, M. Gross, Algebraic point set surface, in: Proceedings of ACM Siggraph 2007, 2007, pp. 23.1–23.9.

[31] N. Amenta, Y. Kil, Defining point-set surfaces, ACM Transactions on Graphics 23 (3) (2004) 264–270.

[32] C. Shen, J. F. O'Brien, J. R. Shewchuk, Interpolating and approximating implicit surfaces from polygon soup, ACM Trans. Graph. 23 (3) (2004) 896–904.

[33] C. Oztireli, G. Guennebaud, M. Gross, Feature preserving point set surfaces based on non-linear kernel regression, Computer Graphics Forum 28 (2) (2009) 493–501.

[34] M. Alexa, A. Adamson, Interpolatory point set surfaces-convexity and hermite data, ACM Transactions on Graphics 28 (2) (2009) No.20.

[35] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial basis functions of minimal degree, Advances in Computational Mathematics 4 (1995) 389–396.

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

[37] P. Cignoni, C. Rocchini, R. Scopigno, Metro: measuring error on simplified surfaces, Computer Graphics Forum 17 (2) (1998) 167–174.

[38] T. Ju, F. Losasso, S. Schaefer, J. Warren, Dual contouring of hermite data, in: Proceedings of ACM SIGGRAPH 2002, 2002, pp. 339–346.

[39] S. Liu, C. C. L. Wang, Error-bound, comparison and sub-sampling for closed-form HRBF surface reconstruction, Technical Report (2015).

[40] W. E. Lorensen, H. E. Cline, Marching cubes: A high resolution 3d surface construction algorithm, Computer Graphics 21 (4) (1987) 163–169.

[41] J. Wang, K. Xu, L. Liu, J. Cao, S. Liu, Z. Yu, X. D. Gu, Consolidation of low-quality point clouds from outdoor scenes, Comput. Graph. Forum 32 (5) (2013) 207–216.

[42] S. J. Liu, C. C. L. Wang, Orienting unorganized points for surface reconstruction, Computers & Graphics 34 (3) (2010) 209–218.

[43] H. C. Batagelo, J. a. P. Gois, Least-squares hermite radial basis functions implicits with adaptive sampling, in: Proceedings of Graphics Interface 2013, 2013, pp. 109–116.