Error-Bound, Comparison and Sub-Sampling for Closed-form HRBF Surface Reconstruction

Shengjun Liu and Charlie C. L. Wang

Abstract—This technical report is a supplementary document for our work [1] for the closed-form formulation of HRBF-based surface reconstruction. Detail formulas can be found in [1]. Here the proof of Lemma for error-bound analysis is provided, and the comparisons with other methods on sets of clear data are given. To support reconstruction on input with highly non-uniformity, a center selection method is also developed.

1 Proof of Lemma

Errors between the quasi-solution $\tilde{\boldsymbol{\lambda}}$ and the exact solution $\boldsymbol{\lambda}$ must be bounded to make the closed-form formulation useful. The following lemmas can be derived.

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

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

when

$$\|\mathbf{D}^{-1}\|_\infty\|\Delta\mathbf{A}\|_\infty < 1. \tag{2}$$

Proof: By $(\mathbf{A}+\eta\mathbf{I})\boldsymbol{\lambda} = \mathbf{y}$, we have

$$(\mathbf{D}+\Delta\mathbf{A})(\tilde{\boldsymbol{\lambda}}+\Delta\boldsymbol{\lambda}) = \mathbf{y}.$$

With the quasi-solution that $\tilde{\boldsymbol{\lambda}} = \mathbf{D}^{-1}\mathbf{y}$, this equation can be converted to

$$\Delta\boldsymbol{\lambda} = \mathbf{D}^{-1}[-(\Delta\mathbf{A})\tilde{\boldsymbol{\lambda}} - (\Delta\mathbf{A})(\Delta\boldsymbol{\lambda})].$$

Then, we apply a maximum norm and get

$$\|\Delta\boldsymbol{\lambda}\|_\infty \leq \|\mathbf{D}^{-1}\|_\infty(\|\Delta\mathbf{A}\|_\infty\|\tilde{\boldsymbol{\lambda}}\|_\infty + \|\Delta\mathbf{A}\|_\infty\|\Delta\boldsymbol{\lambda}\|_\infty),$$

which is also

$$(1-\|\mathbf{D}^{-1}\|_\infty\|\Delta\mathbf{A}\|_\infty)\|\Delta\boldsymbol{\lambda}\|_\infty \leq \|\mathbf{D}^{-1}\|_\infty\|\Delta\mathbf{A}\|_\infty\|\tilde{\boldsymbol{\lambda}}\|_\infty.$$

By the given condition $\|\mathbf{D}^{-1}\|_\infty\|\Delta\mathbf{A}\|_\infty < 1$, we have

$$\|\Delta\lambda\|_\infty \leq \frac{\|\mathbf{D}^{-1}\|_\infty\|\Delta\mathbf{A}\|_\infty}{1-\|\mathbf{D}^{-1}\|_\infty\|\Delta\mathbf{A}\|_\infty}\|\tilde{\boldsymbol{\lambda}}\|_\infty.$$

Combining with $\tilde{\boldsymbol{\lambda}} = \mathbf{D}^{-1}\mathbf{y}$, the lemma has been proved. ■

Assuming there are at most m other centers falling in the support region for each kernel, the error-bound of our quasi-solution can be achieved on the Wendland's CSRBFs.

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

$$1+\eta > m(\frac{5}{4\rho_{\min}} + \frac{35}{\rho_{\min}^2}). \tag{3}$$

Proof: By the definition of the diagonal matrix $\mathbf{D}$ as

$$\begin{aligned}&\mathbf{D} = (\mathbf{D}_{i,j})_{n\times n},\\ &\mathbf{D}_{i,i} = diag(1, \frac{20}{\rho_i^2}, \frac{20}{\rho_i^2}, \frac{20}{\rho_i^2}) + \eta\mathbf{I}_4, \quad \mathbf{D}_{i,j} = 0\ (i\neq j).\end{aligned} \tag{4}$$

we can have

$$\|\mathbf{D}^{-1}\|_\infty = \max_{j=1,\ldots,n}\{\frac{1}{1+\eta}, \frac{\rho_j^2}{20+\eta\rho_j^2}\}.$$

The upper bound of $\|\mathbf{D}^{-1}\mathbf{y}\|_\infty$ can also be obtained from

$$\begin{aligned}\tilde{\boldsymbol{\lambda}} &= \mathbf{D}^{-1}\mathbf{y}\\ &= \{\tfrac{c}{1+\eta}, \tfrac{\rho_1^2\mathbf{n}_1}{20+\eta\rho_1^2}, \cdots, \tfrac{c}{1+\eta}, \tfrac{\rho_n^2\mathbf{n}_n}{20+\eta\rho_n^2}\}.\end{aligned} \tag{5}$$

as

$$\begin{aligned}\|\mathbf{D}^{-1}\mathbf{y}\|_\infty &= \|\tilde{\boldsymbol{\lambda}}\|_\infty\\ &= \max_{j=1,\ldots,n}\{0, \frac{\rho_j^2\mathbf{n}_j^x}{20+\eta\rho_j^2}, \frac{\rho_j^2\mathbf{n}_j^y}{20+\eta\rho_j^2}, \frac{\rho_j^2\mathbf{n}_j^z}{20+\eta\rho_j^2}\}.\end{aligned}$$

Here, superscripts denote the x-, y- and z-components of a vector in $\Re^3$. When $\rho_i > \rho_j > 0$ and $\eta \geq 0$,

$$\frac{\rho_i^2}{20+\eta\rho_i^2} = \frac{1}{20/\rho_i^2+\eta} > \frac{\rho_j^2}{20+\eta\rho_j^2} = \frac{1}{20/\rho_j^2+\eta}.$$

As a result

$$\begin{aligned}\|\mathbf{D}^{-1}\|_\infty &\leq \max(\frac{1}{1+\eta}, \frac{\rho_{\max}^2}{20+\eta\rho_{\max}^2})\\ \|\mathbf{D}^{-1}\mathbf{y}\|_\infty &\leq \frac{\rho_{\max}^2}{20+\eta\rho_{\max}^2}.\end{aligned}$$

When $\rho_j \leq \rho_{\max} < \sqrt{20}$, we can further obtain

$$\|\mathbf{D}^{-1}\|_\infty = \frac{1}{1+\eta}.$$

Now we derive the upper bound of $\|\Delta\mathbf{A}\|_\infty$. From Eq.(4) and

$$\begin{aligned}&\mathbf{A} = (\mathbf{A}_{i,j})_{n\times n},\\ &\mathbf{A}_{i,j} = \begin{pmatrix}\varphi(\mathbf{p}_i-\mathbf{p}_j) & -(\nabla\varphi(\mathbf{p}_i-\mathbf{p}_j))^T\\ \nabla\varphi(\mathbf{p}_i-\mathbf{p}_j) & -\mathbf{H}\varphi(\mathbf{p}_i-\mathbf{p}_j)\end{pmatrix}_{4\times4}.\end{aligned} \tag{6}$$

- *S. Liu is with State Key Laboratory of High Performance Manufacturing Complex and Institute of Engineering Modeling and Scientific Computing, Central South University, Changsha, Hunan 410083, China. E-mail: shjliu.cg@gmail.com*
- *C.C.L. Wang is with Department of Design Engineering, Delft University of Technology, The Netherlands. E-mail: c.c.wang@tudelft.nl*

using $\varphi_{i,j}$ to denote $\varphi_i(\mathbf{p}_j) = \varphi(\mathbf{p}_j - \mathbf{p}_i)$, we can also have

$$\begin{aligned}
\|\Delta\mathbf{A}\|_\infty = \max_{j=1,\ldots,n}\{ & \\
&\sum_i^m(|\varphi_{i,j}| + |\frac{\partial\varphi_{i,j}}{\partial x}| + |\frac{\partial\varphi_{i,j}}{\partial y}| + |\frac{\partial\varphi_{i,j}}{\partial z}|), \\
&\sum_i^m(|\frac{\partial\varphi_{i,j}}{\partial x}| + |\frac{\partial^2\varphi_{i,j}}{\partial x^2}| + |\frac{\partial^2\varphi_{i,j}}{\partial x\partial y}| + |\frac{\partial^2\varphi_{i,j}}{\partial x\partial z}|), \\
&\sum_i^m(|\frac{\partial\varphi_{i,j}}{\partial y}| + |\frac{\partial^2\varphi_{i,j}}{\partial x\partial y}| + |\frac{\partial^2\varphi_{i,j}}{\partial y^2}| + |\frac{\partial^2\varphi_{i,j}}{\partial y\partial z}|), \\
&\sum_i^m(|\frac{\partial\varphi_{i,j}}{\partial z}| + |\frac{\partial^2\varphi_{i,j}}{\partial x\partial z}| + |\frac{\partial^2\varphi_{i,j}}{\partial y\partial z}| + |\frac{\partial^2\varphi_{i,j}}{\partial z^2}|)\}.
\end{aligned}$$

By the derivatives listed in Table 1 and their corresponding upper bounds listed in Table 2, we can have

$$\begin{aligned}
\|\Delta\mathbf{A}\|_\infty &\le \max_j\{m(1+\frac{15}{4\rho_j}), m(\frac{5}{4\rho_j}+\frac{35}{\rho_j^2})\} \\
&\le \max\{m(1+\frac{15}{4\rho_{\min}}), m(\frac{5}{4\rho_{\min}}+\frac{35}{\rho_{\min}^2})\}.
\end{aligned}$$

When $\rho_{\min} \le \rho_{\max} < \sqrt{20}$, it can easily be further simplified to

$$\|\Delta\mathbf{A}\|_\infty \le m(\frac{5}{4\rho_{\min}} + \frac{35}{\rho_{\min}^2}) \equiv \bar{A}.$$

Summarizing all the analysis together, we have

$$\|\Delta\lambda\|_\infty \le \frac{\bar{A}\rho_{\max}^2}{(1+\eta-\bar{A})(20+\eta\rho_{\max}^2)} \tag{7}$$

when $\|\mathbf{D}^{-1}\|_\infty\|\Delta\mathbf{A}\|_\infty \le \bar{A}/(1+\eta) < 1$. To hold this, it should have $1+\eta > \bar{A}$ – that is Eq.(3). The lemma has been proved.

■

2 Comparisons on Clean Data

Here, we test the performance of our approach on sets of clean points, which are uniformly sampled from polygonal meshes. Four models, Ramesses, Raptor, Momento and Neptune, are sampled into sets with $0.58M \sim 4.98M$ points. Our results are compared with three prior methods, including the *Multiple Partition of Unity* (MPU) reconstruction [3], the *Smooth Signed Distance* (SSD) reconstruction [4] and the *Screened-Poisson* reconstruction [5]. Comparisons are shown in Fig.1. We employ the 10-th depth of octree in the SSD and Screen-Poisson methods to generate results in Fig.1. 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 0.001 times of the model's size. Default values are used for other parameters. From observation, it can be found that geometric details on the original mesh can be well preserved by our method while being smoothed out in some prior methods. Publicly available software, Metro tool [2], is employed to compute the average shape approximation error between the reconstructed surface and the original mesh. A bar chat of errors is given in the upper-right of Fig.1. Our method can always generate more accurate results than SSD and MPU. Meanwhile, our results have similar accuracy comparing to Screened-Poisson.

Table 3 gives the computational statistics of tests on these models. We evaluate our methods on a PC with two Intel Core i7-2600K CPUs at 3.4GHz plus 16GB RAM. Due to the close-form formulation, our method does not need any global operation such as solving a large linear system. Therefore, 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

TABLE 1
Derivatives of Wendsland's CSRBF

$\varphi_i(\mathbf{x})$	$(1-\frac{r}{\rho_i})^4(4\frac{r}{\rho_i}+1)$
$\frac{\partial\varphi_i(\mathbf{x})}{\partial x}$	$-\frac{20}{\rho_i^2}(1-\frac{r}{\rho_i})^3(x-x_i)$
$\frac{\partial\varphi_i(\mathbf{x})}{\partial y}$	$-\frac{20}{\rho_i^2}(1-\frac{r}{\rho_i})^3(y-y_i)$
$\frac{\partial\varphi_i(\mathbf{x})}{\partial z}$	$-\frac{20}{\rho_i^2}(1-\frac{r}{\rho_i})^3(z-z_i)$
$\frac{\partial^2\varphi_i(\mathbf{x})}{\partial x^2}$	$-\frac{20}{\rho_i^2}(1-\frac{r}{\rho_i})^3+\frac{60}{\rho_i^3}(1-\frac{r}{\rho_i})^2\frac{(x-x_i)^2}{r}$
$\frac{\partial^2\varphi_i(\mathbf{x})}{\partial y^2}$	$-\frac{20}{\rho_i^2}(1-\frac{r}{\rho_i})^3+\frac{60}{\rho_i^3}(1-\frac{r}{\rho_i})^2\frac{(y-y_i)^2}{r}$
$\frac{\partial^2\varphi_i(\mathbf{x})}{\partial z^2}$	$-\frac{20}{\rho_i^2}(1-\frac{r}{\rho_i})^3+\frac{60}{\rho_i^3}(1-\frac{r}{\rho_i})^2\frac{(z-z_i)^2}{r}$
$\frac{\partial^2\varphi_i(\mathbf{x})}{\partial x\partial y}$	$\frac{60}{\rho_i^3}(1-\frac{r}{\rho_i})^2\frac{(x-x_i)(y-y_i)}{r}$
$\frac{\partial^2\varphi_i(\mathbf{x})}{\partial x\partial z}$	$\frac{60}{\rho_i^3}(1-\frac{r}{\rho_i})^2\frac{(x-x_i)(z-z_i)}{r}$
$\frac{\partial^2\varphi_i(\mathbf{x})}{\partial y\partial z}$	$\frac{60}{\rho_i^3}(1-\frac{r}{\rho_i})^2\frac{(y-y_i)(z-z_i)}{r}$

†Here, $\mathbf{x}=(x,y,z)$, $r=\sqrt{(x-x_i)^2+(y-y_i)^2+(z-z_i)^2}$ and ρ_i is the support size of the radial basis function $\varphi_i(\mathbf{x})$.

TABLE 2
Error Bounds of Derivatives

$\lvert\varphi_i(\mathbf{x})\rvert$	≤ 1
$\lvert\frac{\partial\varphi_i(\mathbf{x})}{\partial x}\rvert$, $\lvert\frac{\partial\varphi_i(\mathbf{x})}{\partial x}\rvert$, $\lvert\frac{\partial\varphi_i(\mathbf{x})}{\partial x}\rvert$	$\le \frac{20}{\rho_i}(1-\frac{r}{\rho_i})^3\frac{r}{\rho_i}$ $\le \frac{5}{4\rho_i}$ (with $r=\frac{\rho_i}{2}$)
$\lvert\frac{\partial^2\varphi_i(\mathbf{x})}{\partial x^2}\rvert$, $\lvert\frac{\partial^2\varphi_i(\mathbf{x})}{\partial y^2}\rvert$, $\lvert\frac{\partial^2\varphi_i(\mathbf{x})}{\partial z^2}\rvert$,	$\le \frac{20}{\rho_i^2}(1-\frac{r}{\rho_i})^2(1+2\frac{r}{\rho_i})$ $\le \frac{20}{\rho_i^2}$ (with $r=0$)
$\lvert\frac{\partial^2\varphi_i(\mathbf{x})}{\partial x\partial y}\rvert$, $\lvert\frac{\partial^2\varphi_i(\mathbf{x})}{\partial x\partial z}\rvert$, $\lvert\frac{\partial^2\varphi_i(\mathbf{x})}{\partial x\partial y}\rvert$,	$\le \frac{60}{\rho_i^2}(1-\frac{r}{\rho_i})^2\frac{r}{\rho_i}$ $\le \frac{15}{2\rho_i^2}$ (with $r=\frac{\rho_i}{2}$)

‡The analysis is based on $|x-x_i| \le r$, $|y-y_i| \le r$ and $|z-z_i| \le r$, and the bound is derived by using the inequality of arithmetic and geometric means.

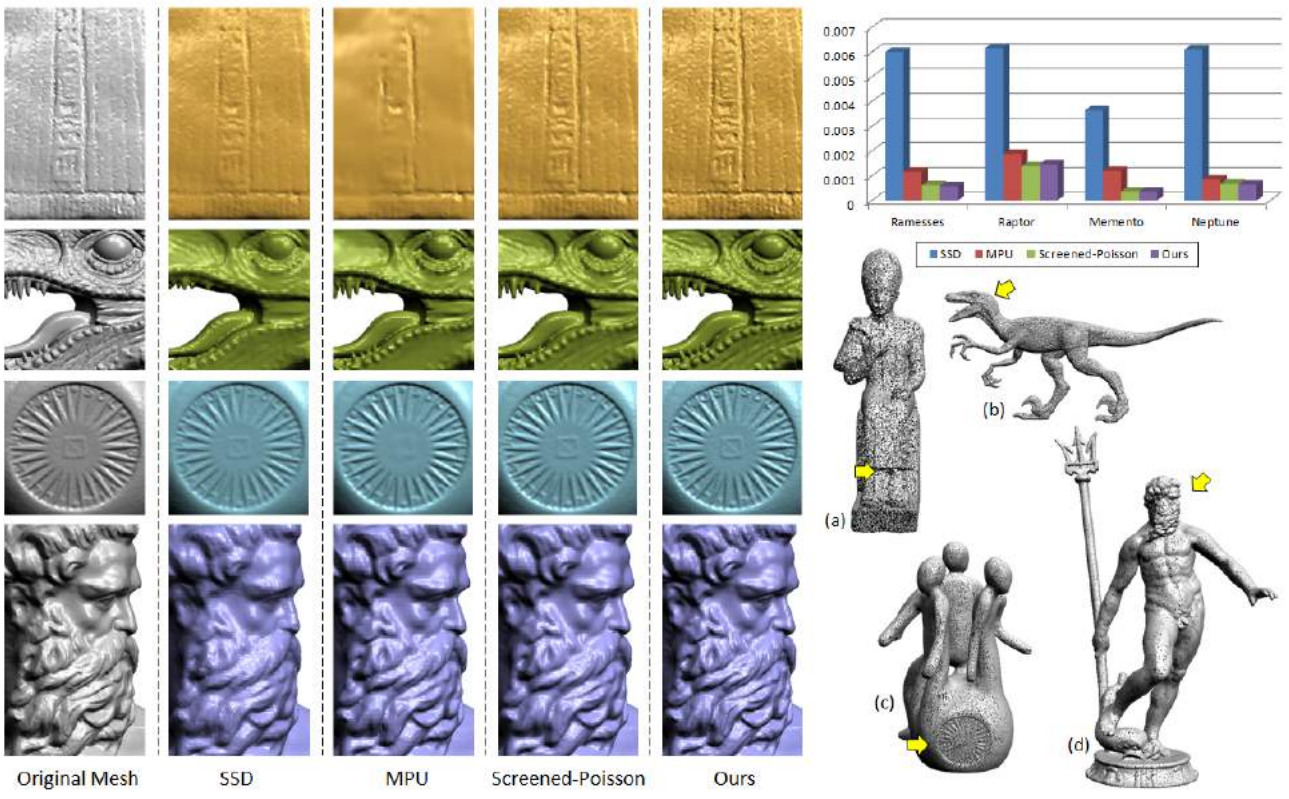

Fig. 1. Experimental tests on clean point cloud that is uniformly sampled from fours mesh models – (a) Ramesses ($0.580M$ pts.), (b) Raptor ($1.00M$ pts.), (c) Momento ($2.52M$ pts.) and (d) Neptune ($4.98M$ pts.). For illustration, only $1/10$ points are displayed for the points of the first three models, and $1/20$ points for the Neptune model. The reconstructions by different methods including SSD, MPU, Screened-Poisson and ours are shown and compared in the left. A bar chat is also given to report the average shape approximation errors on different reconstructions by the Metro tool [2]. We use $s\bar{d} = 0.003$ as the support size and $w = 0.001$ for the grid width of polygonization in all examples here. To conduct a fair comparison, similar number of triangles are generated through the polygonization for different approaches.

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 $O(n^{1.5})$. This leads to a significantly slower performance on models with large number of points (see Table 3). 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 ours. Moreover, our method generates results with smaller shape approximation error than MPU (see Fig.1). In summary, our method is the fastest method and can generate similar results as the best of other three in terms of quality.

It is also interesting to study the error between $\tilde{\boldsymbol{\lambda}}$ and $\boldsymbol{\lambda}$. We measure $\|\tilde{\boldsymbol{\lambda}} - \boldsymbol{\lambda}\|_\infty$ in examples shown above and the results are listed in Table 4. The numerical solver for computing the exact solution runs out of memory on the two examples – Momento and Neptune in Fig.1. Thus, the errors cannot be evaluated. The statistics on Ramesses and Raptor show very small difference between $\tilde{\boldsymbol{\lambda}}$ and $\boldsymbol{\lambda}$.

TABLE 3
Runtime performance of different reconstruction approaches on clean point sets†

Model	Pts.	Time in Seconds*			
		SSD	MPU	Poisson	Ours
Ramesses	0.58M	14,314	61.2	40.8	8.3
Raptor	1.00M	1,799	47.2	31.6	6.8
Memento	2.52M	24,195	138.8	92.6	20.4
Neptune	4.98M	6,772	139.4	114.0	18.9

*Note that, the time reported here includes both the surface reconstruction and the mesh extraction.
†To have a fair comparison, similar number of triangles are generated for different approaches.

TABLE 4
Error Statistics of Quasi-Solution

Model	η	$\|\boldsymbol{\lambda}\|_\infty$	$\|\tilde{\boldsymbol{\lambda}}\|_\infty$	$\|\tilde{\boldsymbol{\lambda}} - \boldsymbol{\lambda}\|_\infty$
Ramesses	$4.57E+5$	$2.15E-6$	$2.16E-6$	$9.52E-8$
Raptor	$1.67E+6$	$6.06E-7$	$5.98E-7$	$1.98E-8$

3 CENTER SELECTION

This is an optional step to be applied when high non-uniformity is observed on the input points. For such a point set, the direct reconstruction by using all points as centers of CSRBFs results in a reconstruction with holes in the sparse regions (see Fig.2(b) for an example). Here, we adaptively select samples from $\mathcal{P}$ and $\mathcal{N}$ to form a subset $\mathcal{C}$. The Hermite points in $\mathcal{C}$ will be used as centers of HRBF in the above method to obtain a better surface reconstruction. Each center, $\mathbf{c}_k$, is also associated with a radius, r_k, which therefore forms a local spherical cover of the given points.

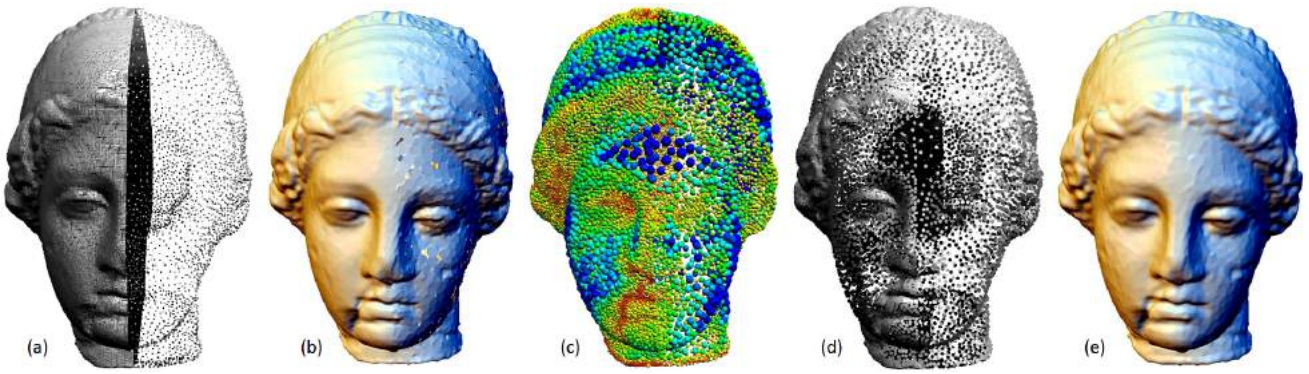

Fig. 2. Adaptive HRBF implicits are generated by our method with the help of center selection: (a) the input set with $100,371$ points in high non-uniformity, (b) the reconstruction using all points as centers of HRBF implicits will lead to holes in the sparse regions, (c) the spherical cover – the spheres 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.

Definition 2 The *degree of coverage* (DoC) at a point $\mathbf{x} \in \Re^3$ is defined as a function

$$g(\mathbf{x}) = \sum_{k=1}^{l} \phi_{r_k}(\|\mathbf{x} - \mathbf{c}_k\|) \tag{8}$$

according to a set of down-sampled Hermite points, $\mathcal{C} = \{\mathbf{c}_1, \mathbf{n}_1, \cdots, \mathbf{c}_l, \mathbf{n}_l\}$.

We wish to generate a minimal spherical cover by controlling DoC in an iterative procedure.

The basic idea of center selection is to form spherical covers by letting DoC at every point in $\mathcal{P}$ not less than a criterion $g_{\min}$ (i.e., $\forall \mathbf{p}_j \in \mathcal{P},\ g(\mathbf{p}_j) \geq g_{\min}$). To this end, the following steps are iteratively run until the criterion is satisfied at all points.

1) In the initial step, $\mathcal{C} = \emptyset$ and $g_j = 0$ is assigned to all points $\mathbf{p}_j \in \mathcal{P}$.
2) Randomly selecting ϖ points with their DoCs less than $g_{\min}$. Among these ϖ points, the point with the smallest $g(\cdot)$ is chosen as a center $\mathbf{c}_k$ to add into $\mathcal{C}$ together with its normal vector.
3) The radius r_k of sphere centered at $\mathbf{c}_k$ is then determined by a quadric-error function

$$q(\mathbf{c}_k, r_k) = \frac{\sum_j \delta_j \phi_{r_k}(\|\mathbf{p}_j - \mathbf{c}_k\|)(\mathbf{n}_j \cdot (\mathbf{c}_k - \mathbf{p}_j))^2}{\sum_j \delta_j \phi_{r_k}(\|\mathbf{p}_j - \mathbf{c}_k\|)},$$

which evaluates how curved the surface inside the sphere is – the shape is represented by sample points in $\mathcal{P}$. In other words, for a highly curved region, a sphere with smaller r_k should be used to reduce the error. Here, δ_j is the average of the squared distances between a point $\mathbf{p}_j$ and its 15 nearest neighboring points. The value of δ_j indicates a weight of point density. The bi-sectional search is taken to obtain a maximal r_k that still satisfies

$$q(\mathbf{c}_k, r_k) \leq q_{err}\bar{L}$$

with $\bar{L}$ being the diagonal length of the input points' bounding box.
4) Updating DoC at all points $\mathbf{p}_j$ within the range $\|\mathbf{p}_j - \mathbf{c}_k\| < r_k$ while $g(\mathbf{p}_j) < g_{\min}$. DoC of $\mathbf{c}_k$ is assigned as $g_{\min}$ to avoid being selected as candidates of centers once again.
5) Go back to step 2) until DoC at all points are not less than $g_{\min}$.

Note that, this iterative procedure is a variant of our prior work in [6] with certain modification to fit the formulation of CSRBF. The efficiency of computation has also been improved. An example result of our minimal spherical covering is shown in Fig.2(c), where the selected centers of RBFs are displayed as spheres. Colors are used to represent the sizes of spheres with red for the smallest and blue for the biggest ones. Samples adaptive to the geometric details have been illustrated as Fig.2(d). In all examples of this paper, $g_{\min} = 1.5$, $q_{err} = 5 \times 10^{-4}$ and $\varpi = 15$ work well.

With the centers selected above, a reduced implicit function can be obtained by

$$\tilde{f}(\mathbf{x}) = -\sum_{j=1}^{n} \langle \frac{\rho_j^2}{20 + \eta\rho_j^2}\mathbf{n}_j, \nabla\varphi(\mathbf{x} - \mathbf{p}_j)\rangle. \tag{9}$$

but with fewer centers of CSRBFs. After applying the center selection step, the density of centers at each region becomes compatible to its neighboring regions – i.e., no sharp change. As a result, a better reconstruction can be obtained (see the result shown in Fig.2(e)).

REFERENCES

[1] S. Liu, C. C. L. Wang, G. Brunnett, and J. Wang, "A closed-form formulation of HRBF-based surface reconstruction by approximate solution," *Computer-Aided Design (Special issue of SPM 2016)*.
[2] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: measuring error on simplified surfaces," *Computer Graphics Forum*, vol. 17, no. 2, pp. 167–174, 1998.
[3] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H. Seidel, "Multi-level partition of unity implicits," *ACM Transaction on Graphics*, vol. 22, no. 3, pp. 463–470, 2003.
[4] F. Calakli and G. Taubin, "SSD: Smooth signed distance surface reconstruction," *Computer Graphics Forum*, vol. 30, no. 7, pp. 493–501, 2011.
[5] M. Kazhdan and H. Hoppe, "Screened poisson surface reconstruction," *ACM Transactions on Graphics*, 2013.
[6] S. J. Liu and C. C. L. Wang, "Orienting unorganized points for surface reconstruction," *Computers & Graphics*, vol. 34, no. 3, pp. 209–218, 2010.