3D Reconstruction Methods Based on Radial Basis Functions for Laser Scanned Data Point Sets

Xiaojun Wu[1], Michael Yu Wang[2] and Qi Xia[2]

[1]Harbin Institute of Technology Shenzhen Graduate School, wuxj@hitsz.edu.cn
[2]The Chinese University of Hong Kong, yuwang, qxia@acae.cuhk.edu.hk

ABSTRACT

Two new approaches of 3D implicit surfaces reconstruction with radial basis functions (RBFs) are proposed in this paper. With the first method, a point set is organized by a balanced binary tree. The cells are controlled to be mildly overlapped and to contain adequate number of points for efficiency and stability. In each subdomain, only one off-surface point in the quasi-normal direction which is estimated by an eigen analysis is used in RBF interpolation. Another method is least square RBFs. This method can overcome the problem of numerical ill-conditioning and over-fitting of traditional RBF reconstruction and it offers a methodology for reconstruction with less centers. These method are versatile and with topological flexibility and numerical efficiency.

Keywords: 3D reconstruction; Radial Basis Functions; Partition of Unity; Least Square Method

1. INTRODUCTION

The problem of surface reconstruction from scattered point datasets has been studied extensively in computer graphics and engineering since the pioneer work of Boissonnat [1] and Hoppe [2]. In particular, the use of a range scanner or laser scanner produces large amount of unorganized point sets in industry, entertainment and archeology, etc. It is desirable to quickly and robustly reconstruct a continuous surface with attributes from these point datasets. Generally, the methods of surface reconstruction fall into two categories [3]. One is Delaunay-based methods [4-6] and another is volumetric and implicit based methods [7-11]. Delaunay triangulation is usually utilized in this kind of method to find the possible neighbors for each point in all directions from all samples. Curst and Cocone algorithm are two of the most known methods [4], [6], [5]. Implicit surface modeling is popular because it can describe complex shapes and its capabilities for surface and volume modeling. Complex editing operations are easily to be performed on such models. Among them, level set methods [12], moving least square methods [13], variational implicit surfaces [14] and adaptively sampled distance field [15] are recent development in this field. Radial basis functions (RBFs) attract more attentions recently in data interpolation in multi-dimensions [14]. It is identified as one of the most accurate and stable methods to solve scattered data interpolation problems. Using this technique, an implicit surface is constructed by calculating the weights of a set of radial basis functions such that they interpolate the given data points. The surface is represented as the zero level set of this implicit function.

The problem of 3D reconstruction from scattered points based on radial basis functions involves two challenges. One is the approach of fast fitting of RBFs for the large scale datasets, and another is the method of fast evaluation. Ohtake and Morse's compactly supported RBFs [9], [16], Carr's greedy algorithm [7], Beatson's GMRES iteration method and Domain Decomposition Method (DDM) [17] and Ohtake and Tobor's Partition of Unity [9], [18] are methods of fast fitting to solve the coefficients of RBFs. Fast multipole method (FMM) is a efficient algorithm to evaluate FRBs with large number of centers [19]. But the far field expansion in the method has to be done separately for every radial basis function and its implementation is intricate and complicated. In RBFs interpolation, if the sampled surface points are used directly, it will lead to trivial solutions to the RBF linear system. In practice, some interior or exterior constraints along the normal direction, called off-surface points, are required to avoid trivial solution. This is a common practice, but it doubles or triples the number of interpolation centers. Furthermore, the RBFs are of a global support and the resulting coefficient matrix is dense. Therefore, it is difficult to use this technique to reconstruct implicit surfaces from large number of point sets consisting of more than several thousands of points. Although compactly supported RBFs can offer a way to deal with large scale point sets since the involved RBF coefficient matrix becomes sparse [9], [20], [16]. Unfortunately, the radius of support has to be chosen globally, which means the approach is not robust and stable against highly non-uniformly distributed point sets where the density of the samples may vary significantly.

In this paper we describe two contributions to the problems of surface reconstruction from large scale unorganized point sets. Firstly, we take a binary tree to subdivide the global domain into overlapping local subdomains. To reconstruct the local surfaces for each subdomain, we only add a single point as the off-surface point to the input points. We demonstrate that, when the off-surface point is chosen properly, the technique is not only efficient but also robust, which has a higher level of scalability. Secondly, we adapt the methodology of least square (LSQ) optimization to surface reconstruction and deduce the corresponding formulations. We show that with this LSQ RBF scheme we can use less centers in reconstruction, moreover, it can avoid the numerical ill-conditioning and "over-fitting" problems of conventional RBF interpolation scheme.

Paper is organized as follows. The related works are summarized in section 2, and in section 3 some relevant theoretical backgrounds are described. The detailed reconstruction approach and some reconstructed examples are presented in section 4. In part 5, formulations of LSQ RBF are developed and some examples are given. The part 6 is conclusion section.

2. RELATED WORKS

The method of RBFs reconstruction based on Partition of Unity (POU) from scattered point datasets is proposed in the [8] and [18]. In [8], a point set is adaptively organized by an octree structure according to the complexity of the local shape. Quadric functions are used to approximate the local shape in each octant cell and the partition of unity functions are utilized to blend the local shapes to obtain the global surface. However, it is not practical to interpolate the noisy data. Tobor utilizes a binary tree to divide the global domain into overlapping subdomains, and the partition of unity functions are employed to blend the local solutions [18]. However, to avoid the trivial solution of RBF linear system, they add a off-surface point to each of the surface points. Contribution of our new scheme is to reduce the number of off-surface points greatly.

3. THEORETICAL BACKGROUNDS

3.1 RBF Formulations

The problem of scattered data interpolation can be stated as: given a set of fixed points $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N \in R^n$ sampled on a surface S in R^3 and a set of function values $f_1, f_2, \cdots, f_N \in R$, find an interpolant $\phi : R^3 \to R$ such that

$$\phi(\boldsymbol{x}_i) = f_i, \quad i = 1,2,\cdots,N. \tag{1}$$

It is proved that the smoothest interpolation function has the simple expression

$$\phi(\boldsymbol{x}) = \sum_{j=1}^{N} \alpha_j g_j \left(\left\|\boldsymbol{x} - \boldsymbol{x}_j\right\|\right) + p(\boldsymbol{x}) \tag{2}$$

where $p(\boldsymbol{x})$ is a polynomial, α_j are coefficients corresponding to each basis and $\left\| \cdot \right\|$ is the Euclidean norm on R^3. The basis function $g(r), r = \left\|\boldsymbol{x} - \boldsymbol{x}_j\right\|$, is a real valued function on $[0, \infty)$, and it is usually unbounded and has global support. The polynomial $p(\boldsymbol{x})$ is appended for achieving polynomial precision. For example, if $p(\boldsymbol{x})$ is a linear polynomial, the coefficients α_j must satisfy the following constraints:

$$\sum_{j=0}^{N} \alpha_j = 0 \text{ and } \sum_{j=1}^{N} \alpha_j x_j = \sum_{j=1}^{N} \alpha_j y_j = \sum_{j=1}^{N} \alpha_j z_j = 0 \tag{3}$$

The RBFs interpolation problem can be expressed as $f(\boldsymbol{x}_i) = \sum_{j=1}^{N} \alpha_i g(\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|) + p(\boldsymbol{x}_i)$ (4)

Solving the linear system (4), we can get the weight coefficients and the polynomial coefficients for the interpolation function $\phi(\boldsymbol{x})$.

3.2 RBF POU Interpolations

It is unpractical to solve the linear system (4) directly for large scale input point sets due to the complexity of $O(N^3)$ surface fitting and $O(N)$ evaluation. A natural solution is the so-called divide-and-conquers strategy which is the central idea of Partition of Unity (POU). The concept of POU is rooted in applied mathematics [22]. The main idea of the partition of unity method is to divide the global domain into smaller overlapping subdomains where the problem can be solved locally on a small scale. The local solutions are thereafter combined together by using blending functions. The smoothness of the global solution in the overlap regions can be guaranteed by the blending functions. The POU method for RBFs surface reconstruction has been applied by Tobor et al. [18]. For completeness, the method is described here briefly.

The global domain Ω is first divided into M overlapping subdomians $\{\Omega_i\}_{i=1}^{M}$ with $\Omega \subseteq \bigcup_i \Omega_i$. For a partition of unity on the set of subdomains $\{\Omega_i\}_{i=1}^{M}$, we then need to define a collection of non-negative blending functions $\{w_i\}_{i=1}^{M}$ with limited support and with $\sum w_i = 1$ in the entire domain Ω. For each subdomain Ω_i, the points within the subdomain are used to compute a local reconstruction function ϕ_i that interpolates these data points. The global reconstruction function Φ is then defined as a combination of the local functions:

$$\Phi = \sum_{i=1}^{M} \phi(\boldsymbol{x}) w_i(\boldsymbol{x}) \tag{5}$$

The blending functions are obtained from a set of smooth functions W_i by a normalization procedure

$$w_i(\boldsymbol{x}) = \frac{W_i(\boldsymbol{x})}{\sum_j W_j(\boldsymbol{x})} \tag{6}$$

where the condition $\sum w_i = 1$ is satisfied. The weighting functions W_i must be continuous at the boundary of the subdomains Ω_i. It can be defined as the composition of a distance function $D_i : R^n \rightarrow [0,1]$ and a decay function $V : [0,1] \rightarrow [0,1]$, i.e., $W_i(\boldsymbol{x}) = V \circ D_i(\boldsymbol{x})$ [18]. The distance function has to satisfy $D_i(\boldsymbol{x}) = 1$ at the boundary of Ω_i. For a 3D axis-aligned box defined from two opposite corners S and T, the distance function D_i is chosen as

$$D_i(\boldsymbol{x}) = 1 - \prod_{r \in (x,y,z)} \frac{4(\boldsymbol{x}_r - S_r)(T_r - \boldsymbol{x}_r)}{(T_r - S_r)^2} \tag{7}$$

The main condition in choosing the decay function V is the continuity between the local solutions ϕ_i in the global reconstruction function Φ. The following functions are suggested to meet various continuity conditions [18].

$$C^0 : V^0(d) = 1 - d; C^1;\ V^1(d) = 2d^3 - 3d^2 + 1; C^2;\ V^2(d) = -6d^5 + 15d^4 - 10d^3 + 1 \tag{8}$$

As for the local error estimates for radial basis function interpolation was studied by Wendland and Wu [21,22].

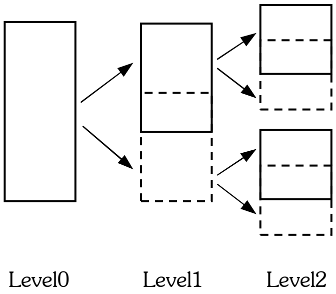


Fig. 1. Process of a binary tree set up.

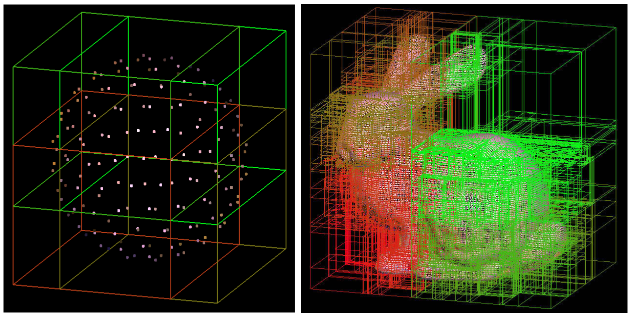


Fig. 2. Examples of binary tree decomposition.

4. RECONSTRUCTION OF RBF IMPLICIT SURFACES

4.1 Space Subdivision with Binary Tree

In order to answer the range query problem efficiently, a data structure has to be set up to divide the scattered point sets into local problems. The range query problem can be state as given a set of points $X \subseteq \mathbb{R}^d$ and a range $R \subseteq \mathbb{R}^d$

and the task is to indicate all points $x_j \in X$ with $x_j \in \mathbb{R}^d$ [21]. Actually, there are several data structures can cope with this problem. Here, we also use a binary tree structure to organize the point sets as in [18]. In our algorithm, a quick sorting scheme is utilized to set up the data structure efficiently. There are four parameters needed to set up a binary tree. They are *Tleafnode* that controls the bound of points in a leaf node, *Tmaxnum* that decides the maximal number of points in each subdomain, *Tminnum* that determines the minimal number of points in each subdomain, and *Tq* that is the overlap quotient to control the number of points in the overlap area. Figure 1 shows a binary tree setup, while some examples of the binary tree decomposition of the real data set are illustrated in Figure 2. The root of binary tree is the bounding box of the whole point dataset such that an arbitrary point in the global domain could find a containing leaf node.

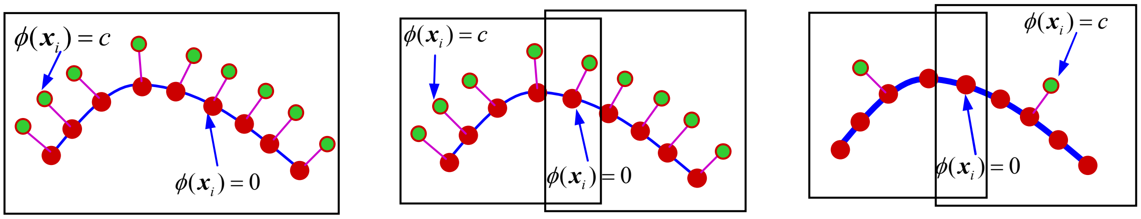
4.2 Generation of Off-surface Points

In the implicit representation of the surfaces constructed with radial basis functions from Eqn. (2), the surface samples satisfies

$$\phi(\boldsymbol{x}_i) = 0, \quad i = 1,2,\cdots,N \tag{9}$$

Therefore, the system Eqn. (4) becomes trivial. The problem can be overcome by introducing additional constraints with artificially generated so called off-surface points with non-zero values $\phi(\boldsymbol{x}_i) = c \neq 0$. A common practice, as suggested in [10], is to introduce an off-surface point for each data point, usually along the normal of the surface. This technique is shown in Figure 3 for the traditional reconstruction scheme (left) and the scheme with the partition of unity (center).

A problem of the conventional technique is that the off-surface points substantially increase the number of data points for interpolation. The total variables would double or triple the amount of sampled data points, which will deteriorate the computation for a large scale problem. In general, it is not necessary to use such a large number of off-surface points. Theoretically, a single off-surface point might be sufficient to avoid the trivial solution of linear system (4). Therefore, we propose a scheme utilizing a single off-surface point for the local reconstruction of the RBF interpolation in each subdomain. Our scheme is illustrated in Figure 3 (right).


$\phi(\boldsymbol{x}_i) = c$
$\phi(\boldsymbol{x}_i) = 0$
$\phi(\boldsymbol{x}_i) = c$
$\phi(\boldsymbol{x}_i) = 0$
$\phi(\boldsymbol{x}_i) = c$
$\phi(\boldsymbol{x}_i) = 0$

Traditional RBF scheme

Traditional POU-RBF scheme

Our POU-RBF scheme

Fig. 3. Comparison of three RBF schemes.

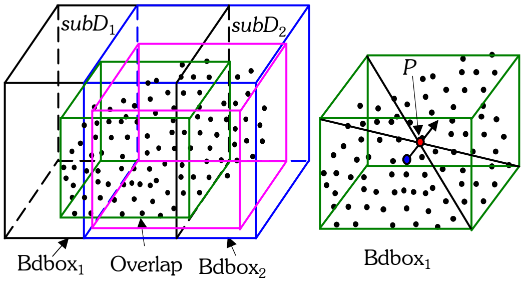


Fig. 4. Generation of the one off-surface point.

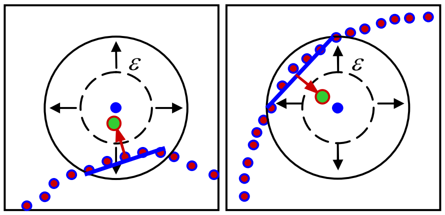


Fig 5. Enlarging the spherical domain for total least square fitting to estimate normal. Left: the center of ball locates outside of the model surface. Right: the ball center is outside.

Theoretically speaking, the off-surface point can be anywhere in the subdomain. But if the off-surface point is located on the boundary of the subdomain, the reconstruction system (4) may become unstable. In our algorithm, an efficient scheme is used to generate the off-surface point in each subdomain. As shown in Figure 4, Bdbox_1 and Bdbox_2 are the bounding boxes of two neighboring subdomains, *subD*$_1$ and *subD*$_2$, respectively. Taking Bdbox$_1$ as an example, we firstly find the center *P* of the bounding box which may lie inside or outside of the object. Then, a point nearest to the point *P* is found. This data point is then offset along its normal with a small distance to yield an off-surface point. This off-surface point is then used for the corresponding subdomain to reconstruct the local RBF function ϕ_i using Eqn. (2).

If the surface samples are not equipped with normal, this normal has to be estimated to generate the off-surface point. Here, we use the covariance analysis for the estimation. In this estimation, a best fitting plane is computed to the subset, and the problem reduces to an eigen analysis. The scheme is illustrated in the figure 5.

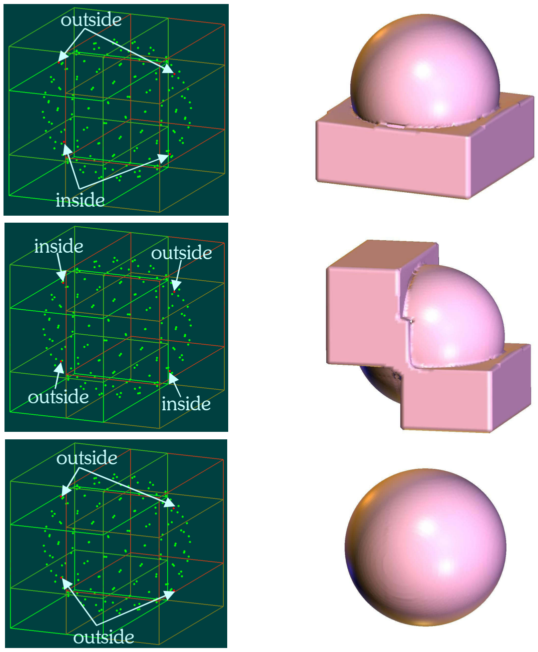


Fig. 6. The influence of the off-surface point position.

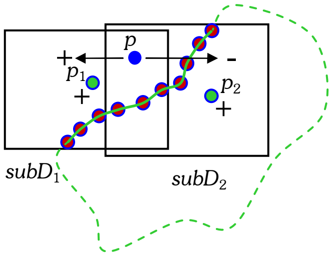


Fig. 7. A point is evaluated in different subdomain with different sign.

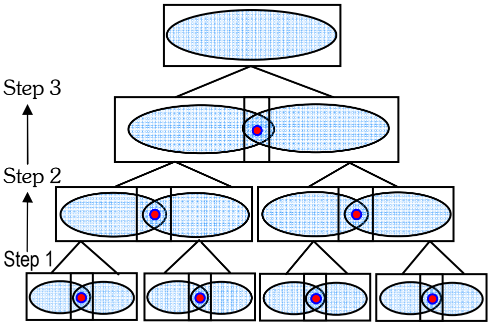


Fig. 8. The process of coefficients consistency.

4.3 Making Consistency of the Coefficients

However, it is difficult to distinguish the inside and outside of the off-surface point generated from the above described method. If these points are used directly to reconstruct surface, some unexpected results can be caused, as illustrated in figure 6. To overcome this drawback, we propose a method to guarantee the inside/outside consistency by using the hierarchy of binary tree in a bottom to up way.

To reveal this procedure, we assume $\phi(p_1)$ and $\phi(p_2)$ are positive but they lie on different side of the zero level-set, shown in figure 7. From p_1 and $x_i \in subD_1$, we get coefficients $\alpha_{d1,i}$, and from p_2 and $x_j \in subD_2$, we get $\alpha_{d2,j}$. Let $p \in subD_1$ and $p \in subD_2$, so $\phi(p) > 0$ in $subD_1$, but $\phi(p) < 0$ in $subD_2$. From Eqn. (2), we can see if the off-surface point is put in the wrong side of zero level set, it just affects the sign of coefficients not the magnitude. So if we fix the sign of coefficients in $subD_1$ and flip the coefficients in $subD_2$, that is $\alpha_{\ d2,j} = -\alpha_{d2,j}$, we can get the correct coefficients orientation which not being influenced by the direction of the estimated normal.

In our reconstruction scheme, the process of coefficients consistency adaptation starts from the leaf node of the binary tree. Since our binary tree is perfect full, each pairs of leaf node has an overlap area. So we take the center of the overlap as a test point to evaluate and check the product of two function values. If the sign of product is negative, it shows that one of the set of coefficients should be flipped. After this step, the coefficients in each pair of leaf nodes with a same parent are consistent, but the coefficients in leaf node which have different parents may be inconsistent. Then we go to step 2. The process is the same as the step 1. Finally, it goes to the root of the binary tree; the flipping process can be completed.

4.4 Experimental Results

In this section we present some results of our reconstruction method. In our implementation, we solve the local linear system (3) with both singular value decomposition (SVD) and LU factorization in each subdomain. The LU is little bit faster than SVD, but it is less stable. We also employ the marching cubes algorithm for surface polygonization. The hardware configuration is Intel Pentium 1.5 GHz with 512 MB RAM and WinXP running on notebook. In Figure 9 some visual examples of the reconstruction of our proposed method are shown. The quality of the reconstructions is highly satisfactory.

Model	*Num. Points*	*Ttree* (s)	*Trec* (s)	*Tpoly* (s)	*Ttotal* (s)
Capsule	338	0	0.88	2.03	2.91
Knot I	1,440	0.02	2.39	1.73	4.14
Venus I	11,121	0.24	19.25	1.64	21.13
Knot II	28,659	0.44	58.13	1.47	60.04

Tab. 1. Computational cost reconstruction with our scheme.

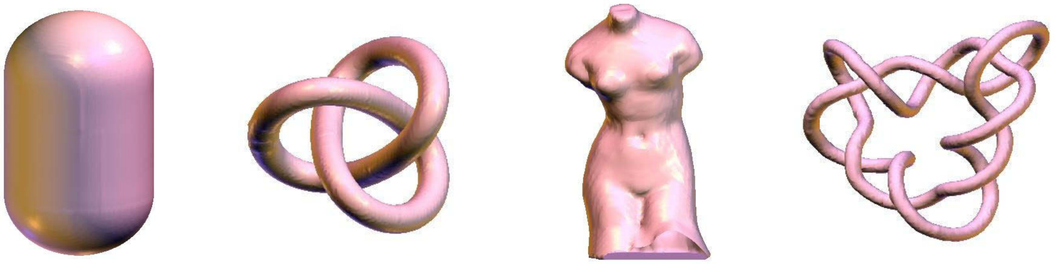


Fig. 9. Reconstructed results with one off-surface point POURBF.

In Table 1 we list the quantitative results of the reconstruction of four different data sets with different point density, where Ttree stands for the time cost of binary tree set up, *Trec* is the reconstruction time of the local RBF system, *Tpoly* means the polygonization time of RBF implicit surface, and *Ttotal* is total time cost of reconstruction. The parameters of binary tree are *Tleafnode,* 100*; Tmaxnum,* 180*; Tminnum*, 40 and *Tq* 0.03 respectively. As reported in [18], the POU-RBF technique exhibits a linear complexity in the reconstruction time with respect to the number of data points. It is clearly shown that for the total reconstruction time, the use of a single off-surface point reduces the computational effort substantially.

5. LEAST SQUARE RBF BASED SURFACE RECONSTRUCTION

5.1 LSQ RBF

Though POU offers an efficient reconstruction scheme, some other significant problems still remain unsolved. Since the collocation method uses the whole data sets both as data and centers, numerical ill-conditioning often occurs due to small distance between some centers, especially for the very dense datasets, which will cause linear dependency of coefficient matrix. Another issue of POU scheme is the so-called "over-fitting" problem as the interplant is too flexible such that it not only fits the surface but also fits the noise which was introduced into the data during sampling. Fortunately, theories of RBF networks lend powerful tools to solve the above mentioned problems.

We rewrite the RBF formulation into

$$\phi(x)=\sum_{j=1}^{M}\alpha_j g(x-x_j)+\alpha_0=\boldsymbol{g}^T\boldsymbol{\alpha} \tag{10}$$

where $\boldsymbol{g}=\left[\boldsymbol{g}_1,\boldsymbol{g}_2,\cdots,\boldsymbol{g}_M,1\right]^T$, $\boldsymbol{\alpha}=\left[\alpha_1,\alpha_2,\cdots,\alpha_M,\alpha_0\right]^T$, M is the number of points used in reconstruction (M is less than the total number of given point sets $\boldsymbol{N}$), and *T* is the transpose operator. Function values on the data points $\boldsymbol{h}$ and $\hat{\boldsymbol{h}}$ indicating the function value with noisy data. Let $\boldsymbol{G}=\left[\boldsymbol{g}_1^T,\boldsymbol{g}_2^T,\cdots,\boldsymbol{g}_N^T\right]_{N\times M}^T$, we have

$$\phi(x)=\boldsymbol{G\alpha} \tag{11}$$

The RBF interpolant interpolates the sample points with function values $\boldsymbol{h}$, so the interpolation equations can be derived from the following optimization problem

$$J=\frac{1}{2}(\phi-\hat{\boldsymbol{h}})^T(\phi-\hat{\boldsymbol{h}})=\frac{1}{2}(G\boldsymbol{\alpha}-\hat{\boldsymbol{h}})^T(G\boldsymbol{\alpha}-\hat{\boldsymbol{h}})\to\min \quad (12)$$

To deal with the noisy data, a "cost function" is augmented the Eqn. (19) to penalize the large weights.

$$J=\frac{1}{2}\left[(G\boldsymbol{\alpha}-\hat{\boldsymbol{h}})^T(G\boldsymbol{\alpha}-\hat{\boldsymbol{h}})+\boldsymbol{\lambda}\boldsymbol{\alpha}^2\right]\to\min \quad (13)$$

Eqn. (13) is differentiated to get the solution

$$\frac{\partial J}{\partial\alpha}=G^T(G\boldsymbol{\alpha}-\hat{\boldsymbol{h}})+\boldsymbol{\lambda}\boldsymbol{\alpha} \quad (14)$$

Let Eqn. (14) equals to zero and gets

$$\left(G^TG+\Lambda\right)\boldsymbol{\alpha}=G^T\hat{\boldsymbol{h}} \quad (15)$$

Then $\boldsymbol{\alpha}$ can be obtained as

$$\boldsymbol{\alpha}=\left(G^TG+\Lambda\right)^{-1}G^T\hat{\boldsymbol{h}} \quad (16)$$

In ordinary least square where there is no weight penalty, the solution becomes

$$\boldsymbol{\alpha}=\left(G^TG\right)^{-1}G^T\hat{\boldsymbol{h}} \quad (17)$$

The dimension in the above formulations can be refined as $G:N\times M$, $\boldsymbol{h}$ and $\hat{\boldsymbol{h}}:N\times 1$, $\boldsymbol{\lambda}:1\times M$, $\boldsymbol{\alpha}:M\times 1$ and $\Lambda:M\times M$ a identity matrix respectively, where the low degree of polynomial is not considered. It can be appended in accordance with specific basis functions. When coefficients $\boldsymbol{\alpha}$ are solved, the implicit surface can be reconstructed with fewer centers, M , than the total of samples $\boldsymbol{N}$.

Model	*Num. Points*	*Ttree* (s)	*Trec* (s)	*Tpoly* (s)	*Ttotal* (s)
Fandisk	6,745	0.12	114.76	2.65	117.53
Knot II	28,659	0.48	324.05	3.28	327.81
Bunny	34,834	0.61	525.82	6.44	532.87
Venus II	72,545	1.22	1,086.45	10.45	1098.12

Tab. 2. Quantitative results of LSQ RBF.

5.2 Some Examples

As to verify our LSQ RBF scheme, we test lots of scattered point datasets. Tab.2 illustrates the computational costs of four of them. From the table, we can see that the total time cost is more than the scheme proposed in section 4 as the conventional off-surface-point method is utilized to avoid the trivial solution of RBF. Figure 10 shows examples in 2D case, figure (a) shows the original curve and samples, say $N=138$. (b), (c) and (d) are the reconstructed curves, with the numbers of knots decreasing gradually; say $M=68,46,26$ respectively, the maximal errors increase correspondingly. We found that using forty to fifty percent of the original samples can get satisfactory results. Figure 11 shows the 3D cases, with knot points decreasing; the reconstructions do not exhibit great visual difference. More complex examples are shown in figure 12. These examples show highly satisfactory quality.

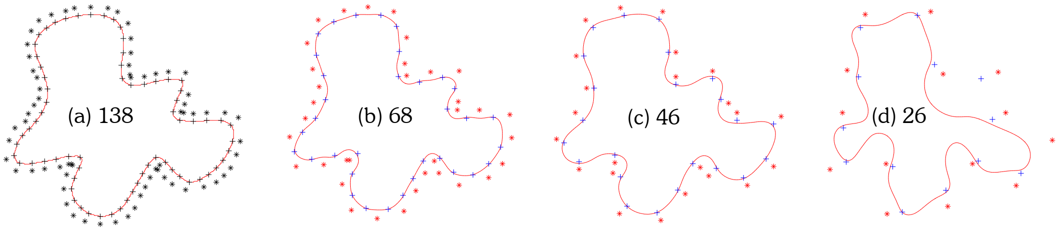


Fig. 10. LSQ RBF reconstruction in 2D case.

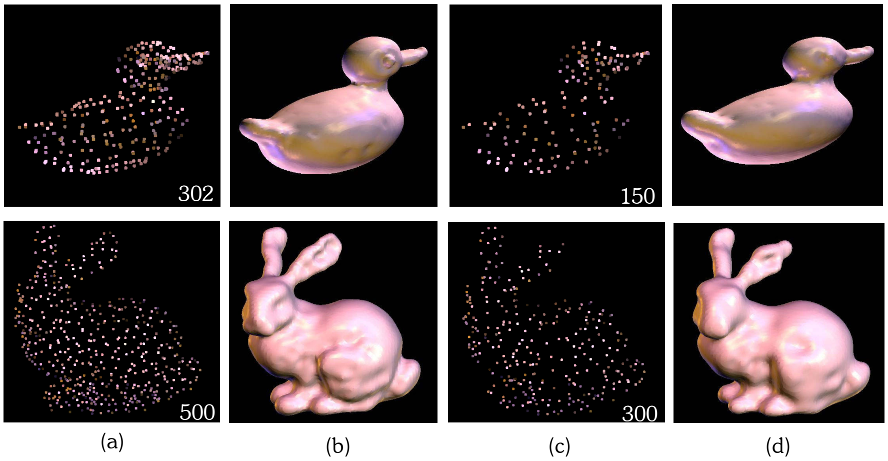


Fig. 11. LSQ RBF reconstructions with the number of knots decreasing.

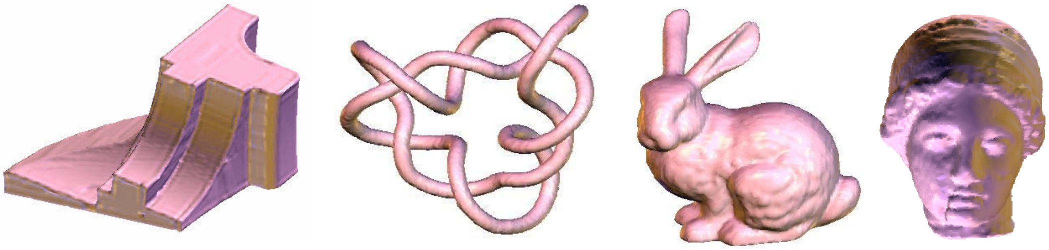


Fig. 12. Examples of LSQ RBF reconstruction in 3D cases.

6. CONCLUSIONS

Reconstruction algorithm with RBF is positive definite, most stable and accurate. It can offer powerful tools in reverse engineering, medical imagery, archeology or other fields. We contribute two approaches to the problem of surface reconstruction in this paper. The first method is based on the implicit representation with radial basis functions and the partition of unity. We propose to estimate a normal and generate a single off-surface point for each subdomain of the partition of unity, in contrast to the conventional RBF reconstruction methods, where a full set of off-surface points are used. Accordingly, a RBF coefficients consistency scheme is proposed. It is shown that this approach reduces the local reconstruction time substantially. We also developed another reconstruction method based on least square optimization, call LSQ RBF. Although this algorithm is time consuming, it can overcome the problem of numerical ill-conditioning and over-fitting of traditional RBF reconstruction. Moreover, fifty to sixty percent samples can be gotten rid of and it can still get quantitatively satisfactory results. With the methods proposed in this paper, it will be more efficient to reconstruct very large scale point datasets and carry out geometry processing, such as hole filling, mesh repairing, model bending, etc.

7. REFERENCES

[1] Boissonna, J.-D., Geometric structures for three dimensional shape representation. *ACM Transactions on Graphics*, Vol.3, No. 1, 1984, pp 266-286.

[2] Hoppe, H, DeRose, T., Duchamp, T., McDonald, J. and Stuetzle, W., Surface reconstruction from unorganized points. *In Proceedings of ACM SIGGRAPH 1992*, pp 71-78.

[3] Oliver Schall and Marie Samozino, Surface from scattered points: a brief survey of recent developments, 1st *International Workshop on Semantic Virtual Environments*, Villars sur Ollon, Switzerland, 2005, pp 138-147.

[4] Amenta, N., Bern, M. and Kamvysselis, M., A new Voronoi-based surface reconstruction algorithm. *In Proceedings of ACM SIGGRAPH 1998*, pp 415-421.

[5] Dey, T. K. and Goswami, S., Tight cocone: A watertight surface reconstructor. In *Proc. 8th ACM Sympos. Solid Modeling Applications*, 2003, pp 127-134.

[6] Amenta, N., Choi, S. and Kolluri, R., The power crust. In *Proceedings of 6th ACM Symposium on Solid Modeling*, 2001, pp 249-260.

[7] Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C. and Evans, T. R., Reconstruction and representation of 3D objects with radial basis functions. *In Proceedings of ACM SIGGRAPH 2001*, 2001, pp 67-76.

[8] Ohtake, Y., Belyaev, A., Alexa, M., Turk, G. and Seidel, H. P., Multi-level partition of unity implicits. *ACM Transactions on Graphics*, *Proceedings of SIGGRAPH 2003*. Vol. 22, No. 3, pp463-470.

[9] Ohtake, Y., Belyaev, A. G. and Seidel, H. P., A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions. *In Shape Modeling International 2003*, 2003, pp 153-161.

[10] Turk, G., Dinh, H. Q., O'Brien, J. and Yngve, G., Implicit surfaces that interpolate. *In Shape Modeling International 2001*, Genova, Italy, 2001, pp 62-71.

[11] Tobor, P., Reuter, and Schilck, C., Efficient reconstruction of large scattered geometric datasets using the partition of unity and radial basis functions. *Journal of WSCG 2004*, Vol. 12, 2004, pp 467-474.

[12] Zhao, H., Osher, S. and Fedkiw, R., Fast surface reconstruction using the Level Set method, *1st IEEE Workshop on Variational and Level Set Methods*, in conjunction with the 8th International Conference on Computer Vision (ICCV), Vancouver, Canada, 2001, pp 194-202.

[13] Klein, J. and Zachmann,G., Point cloud surfaces using geometric proximity graphs, *Computers & Graphics*, Vol. 28, No. 6, 2004, pp 839-850.

[14] Turk, G. and O'Brien, J., Variational implicit surfaces. *Technical Report GIT-GVU-99-15*, Georgia Institute of Technology, 1998.

[15] Frisken, S. F. and Perry, R. N., A. Rockwood, and T.R. Jones. Adaptively sampled distance fields: A general representation of shape for computer graphics. *In Proceedings of ACM SIGGRAPH2000*, 2000, pp 249-254.

[16] Morse, B., Yoo, T.S., Rheingans, P. et al., Interpolating implicit surfaces from scattered surfaces data using compactly supported radial basis functions. *In Proceedings of Shape Modeling International (SMI2001)*, 2001, pp 89-98.

[17] Beatson, R. K., Light, W. A. and Billings, S., Fast solution of the radial basis function interpolation equations: domain decomposition methods. *SIAM J. Sci. Comput*. Vol. 22, No. 5, 2000, pp 1717–1740.

[18] Tobor, I., Reuter, P. and Schlick, C., Multiresolution reconstruction of implicit surfaces with attributes from large unorganized point sets. *In Proceedings of Shape Modeling International (SMI 2004)*, 2004, pp 19-30.

[19] Beatson, B. and Newsam, G., Fast evaluation of radial basis functions. *Computational Mathematics and Applications*, Vol. 24, No. 12, 1992, pp 7-20.

[20] Ohtake, Y., Belyaev, A. and Seidel, H., 3D Scattered Data Approximation with Adaptive Compactly Supported Radial Basis Functions. *In Shape Modeling and Applications (SMI 2004)*, 2004, pp 31-39.

[21] Wendland, H., Fast evaluation of radial basis functions: methods based on partition of unity. In C.K. Chi, L.L. Schumaker, and J. Stöckler, editors, *Approximation Theory X: Abstract and Classical analysis*, Vanderbilt University Press, Nashville, 2002, pp 473-483.

[22] Wu, J. and Kobbelt, L. P., A stream algorithm for the decimation of massive meshes. *In Graphics Interface 2003 Proceedings*, Halifax, Canada, 2003, pp 185-192.