Algorithm of Scattered Data Reduction for Surface Reconstruction Using Radial Basis Function

Xinping Ji¹, Xiaojun Wu¹ and Michael Yu Wang²
¹ HIT Shenzhen Graduate School, Shen zhen, university town, 518055, China
{jixinping, wuxj} @hitsz.edu.cn
² The Chinese University of Hong Kong, N.T. Shatin, Hong Kong, China
yuwang@caec.cuhk.edu.hk

Abstract

This paper concerns a method to reduce centers progressively for radial basis function surface reconstruction from scattered point set to decrease the computational complexity, which belongs to the category of data filtering. In the procedure of the algorithm, we use kd tree to reduce the computational complexity and make it practical for large number of points, then use the radial basis function to approximate, the good performance of the proposed filtering scheme is finally shown by some experimental examples.

1. Introduction

The surface reconstruction from scattered points is one of the important problem and hot research area in industry, motion picture, virtual reality etc. In surface reconstruction, two methods are often used, one is using explicit surface or parametric method and another is using implicit surface [1]. For the virtue of implicit surface, it has attracted the attention of many researchers. However, the computation of implicit surfaces has often been hampered by the constraints of available processing power and the limited complexity of the models that can be created. Much effort is contributed on solving this problem, such as Shepard’s method [2], Thin plate spline method, B-spline method and so on. To create interpolating implicit surface is the task of scattered data interpolation, moreover the desired solution is a function that subject to the interpolation constrains. There are several numerical methods that can be used to solve this type of problem, among various techniques, the radial basis function [3, 4] is one of the most popular method for the good interpolation property [5], which can prove that RBF is one of the most accurate and robust interpolation techniques. The radial basis function approach is especially well suited for surface reconstruction, but it has much limitations with the increasing of the point sets. Here we introduce another efficient technique for dealing with scattered data in three dimensions, which will be helpful for surface reconstruction with radial basis function (RBF). In addition, a large variety of applications of radial basis function prove that it has ability to produce high quality shapes, process irregularly sampled and noisy data, repair incomplete data, handle shapes of arbitrary topological complexity, RBF is one of the most accurate and robust interpolation techniques. The radial basis function approach is especially well suited for surface reconstruction, but it has much limitations with the increasing of the point sets. Here we introduce another efficient technique for dealing with scattered data in three dimensions, which will be helpful for surface reconstruction with radial basis function (RBF).
and define the interpolating implicit surface in a small vicinity of the interpolated points [8].

Recently, there are another discussion [9] shows that with relative small number of centers, we can also get good performance of approximation. So we think that a center reduction scheme combined with RBFs methods may also achieve high quality approximation in this paper, we describe an approach of scattered data reduction for RBF implicit surface reconstruction. The RBF interpolation methods briefly discussed in section 2, and algorithm details are provided in section 3. Section 4 offers some experimental results and analysis. The final section is conclusion.

2. RBF interpolation

In modern approximation theory, the radial basis function method is one of the most often used approach [10, 11] when the task is to approximate scattered data in multiple dimension. Radial basis function is radially symmetric at a point, different radial function can be chosen to solve different equation. Radial basis function interpolation is to define the interpolation function as a linear combination of radial symmetric basis functions, each center at a data point. Given a set of piecewise point sets \( z_1, \ldots, z_N \in \mathbb{R}^n \), and a set of function value \( \{h_1, \ldots, h_N\} \), we want to find a function \( f : \mathbb{R}^d \rightarrow \mathbb{R} \) with
\[
\forall i \quad f(z_i) = h_i \tag{1}
\]
Using radial basis function to reconstruct the point set \( Z \), the function \( f \) the equation
\[
f(z) = \sum_{i=1}^{N} \alpha_i \phi(z - z_i) + p(z) \tag{2}
\]
\( \|z - z_i\| \) is the Euclidean distance, \( \alpha_i \) is the weights, \( \phi : \mathbb{R} \rightarrow \mathbb{R} \) is basis function, and \( p \) is polynomial. If \( p \) is linear polynomial, the coefficient \( \alpha_i \) must satisfy the following constrain,
\[
\sum_{i=1}^{N} \alpha_i = 0 \quad \text{and} \quad \sum_{j=1}^{N} \alpha x_j = \sum_{j=1}^{N} \alpha y_j = \sum_{j=1}^{N} \alpha z_j = 0 \tag{3}
\]
The RBF interpolation problem [12] can be expressed as
\[
f(z) = \sum_{i=1}^{N} \alpha_i \phi(z - z_i) + p(z) \tag{4}
\]
the equation above (1), (2), and (3) determine the linear system
\[
Ax = b \tag{5}
\]
Where \( A = \begin{bmatrix} \Phi & Z^T \\ Z & 0 \end{bmatrix} \) and the vector \( x \) is composed of the weights \( \alpha_i \) and the polynomial coefficients \( p_i \) for equation. The solution of this system (4) can be achieved by solving above equation.

3. The Description of Algorithm

In most applications, the approximation method is often utilized instead of interpolation because it is not necessary to use all the points as the RBF centers. Moreover, the above description also shows that when the points are extremely large, to get the solution will be difficult and complicated. So it is practical to reduce the centers for efficient RBF reconstruction when the precision is not strictly.

Given a finite scattered point sets \( Z = \{z_1, \ldots, z_N\} \subset \mathbb{R}^d \) \( d \geq 1 \), and assume \( M = \|Z\| \) be the size of the set \( Z \), and let \( X \) denote its nonempty subsets, i.e. \( X \subset Z, X \neq \emptyset \). Let \( \|x - z\| \) be any normal on \( \mathbb{R}^d \), and for any point \( z \in Z \) let \( d_z(z) = \min \|x - z\| \) denotes the distance between \( z \) and the subset \( X \). Similar to [13], this section also concerns the construction of a sequence \( \{X_i\}_{i=1}^{n-M} \subset Z \) of subsets, with decreasing sizes \( |X_i| = M - i \), such that for each \( x \in X_i \subset Z \), its covering radius \( r_{X_i} = \max d_z(z) \) on \( Z \) is small. Through this iterative procedure, we can get the significant centers of RBF.

The construction of such a sequence is accomplished by using the operators called filter, one at a time, the action returns a locally optimal subset \( X_i \subset Z \). This scheme is composed of greedy thinning, a recursive point removal strategy, and exchange, a postprocessing local optimization procedure. In our algorithm, greedy thinning and exchange are also two main ingredients for scattered data reducing. In section 3.1, a standard for controlling any optimal current subset is provided.

3.1 Greedy Thinning and Exchange.

Greedy thinning is known as an efficient and effective method of dynamic programming for solving optimization problems. In our particular situation, a greedy thinning algorithm is one procedure that at each step one point is removed, such that the resulting covering radius is minimal among all other possible point removals. Below is the definition for a removable point [14].

**Definition1**: for any \( X \in Z \) with \( |X| \geq 2 \), a point \( x' \in X \) is said to be removal from \( X \), iff \( x' \) minimizes the covering radius \( r_{X \setminus \{x\}} \) among all points in \( X \), i.e. \( r_{X \setminus \{x\}, Z} = \min_{x \in X} r_{X \setminus \{x\}, Z} \).
This definition for a removal point is different from those used in [13,15,16], where a removal point is a point that minimizes the distance to its nearest neighbour in the current subset $X$. While, the removal criterion of definition1 here depends also on the point set $Y = Z \setminus X$, which has already been removed in previous steps. To facilitate the computation, we also use the following characterization for removable points [14], which works with Voronoi diagrams introduced above. To this end, recall that for any finite point set $X$ and $x \in X$ the convex polyhedron $V_x(x) = \{ y \in R^d : d_x(y) = \| y - x \| \}$ denotes the Voronoi region of $x$ w.r.t. $X$, comprising all points in space whose nearest neighbor in $X$ is $x$.

**Theorem:** Let $X \subset Z$ with $|X| \geq 2$. Every point $x \in X$, which minimizes the local covering radius

$$r(x) = r_{x,x(Z\setminus V_x(x))}$$

among all points in $X$ is removable from $X$ [14].

The relation of the covering radius and the local covering radius is shown in [14]. We can use the local covering radius to take the place the covering radius, in another words, the point that minimizes the local covering radius is also a removable point.

For the construction of suitable filters, we offer one additional useful ingredient, i.e. exchange, and the idea of exchange is to reduce the covering radius by additional useful ingredient, i.e. exchange, and the idea of exchangeable point pairs between $X$ and $Y = Z \setminus X$. The following section provides useful criteria for an efficient localization of exchangeable point pairs.

Let $X \subset Z$ denotes a fixed subset of $Z$, let $Y = Z \setminus X$. Moreover, $Y' = \{ y \in Y : d_x(y) = r_{x,Y} \}$ stands for the set of all points in $Y$ where the maximum $r_{x,Y}$ is attained. The following corollary [14] provides sufficient conditions, which are helpful for the purpose of quickly locating exchangeable points.

**Corollary:** Let $y' \in Y$ satisfy $d_x(y') \geq d_x(y)$ for all $y \in Y \setminus \{ y' \}$. Then, for $x \in X$, the pair $(x,y') \in X \times Y$ is exchangeable, if they satisfy $d_x(y') \geq r(x)$ (7).

For the points set $Z = \{ z_1, \ldots, z_n \}$, the significances of $Z$ is $\sigma(z) = d_{z,x}(z)$, then the local optimal subset satisfies the condition: suppose $x \in Z$ is an optimal subset of size $|X| = M - n$, then $\sigma(y) \leq r^n_x$, for all $y \in Y = Z \setminus X$. In the above formula, $r^n_x$ is cover radius of optimal subset [14]. Then according to the corollary and the condition of optimal subset, in our algorithm, after satisfying the precondition in corollary, we use the following formula

$$d_x(y') \geq \sigma(x)$$

(8) to be the condition of exchange. Then after finite many steps, we will get the optimal subset.

### 3.2 The Procedures of the algorithm

As we have known, greedy thinning is used to solve the optimal problem, so there is a standard for the subset, which returns from the action of greedy thinning [14]. However, in our situation, we cannot consider the condition, and just find the removal point for the current set $X$ and $Y$, and the set that returns from greedy thinning is not always the optimal set. Then we implement the action of exchange, according to the condition (8), the point pairs are located and exchanged. The action terminates when there are no points satisfying the condition (8). At this time, the set is the local optimal. Therefore, we must complete the whole procedure to get the local optimal set. For greedy thinning, recall the definition1 for a removable point, and the characterization in the theorem. For the efficient implementation of the exchange algorithm, we work with condition (8) for locating point pairs. During the performance of the greedy thinning and exchange, firstly we randomly separate the points of current set into two sets, In Fig.1, these two subsets are shown as part1 and part2, and these two sets are stored in two lists. When we remove one point, we should accomplish following steps.

**Step 1.** Locating the removal point $x'$ from $X$ list, which satisfies the theorem of removal point.

**Step 2.** Let $X \setminus x'$ and so $Y = Y \cup x'$.

**Step 3.** Locating $y'$ from the $Y$ -list and $x$ from $X$ list which satisfies the corollary of the exchange.

**Step 4.** Let $X = (X \setminus x') \cup y'$ and so $Y = (Y \cup x') \cup y' \cup x$.

The exchange process terminates after finitely many steps, and we get the local optimal subset. Then Update the voronoi diagram in order to obtain the voronoi diagram of the point set $X = (X \setminus x') \cup y'$, update the local covering radius of the point in $X$, and update the significance $d_x(y')$ of those points in $Y = (Y \cup x') \setminus y'$, which is prepared for the next point.

The whole procedure can be described as following Fig.1. Then the final results we want are achieved by recursively applying the above steps. For the small number points, this algorithm achieves the good quality and costs very short time. However, when the number of points becomes very larger, to deal with the points in a domain simply using this method will take a lot of time, so we employ a kd tree for range query. Using k-d tree, we can convert the globe domain into some local domain. Then we can deal with the points in local domain with the above procedure, one after another.
local domain. Moreover, when we use the k-d tree, we
should control the number point of overlap, and the
number of point to remove in every local domain.
Finally, we will process the whole points of the domain.
The following figures in section 4 also show that using
k-d tree, we can not only deal with a large point using
short time, but we also get the similar quality as in
global domain, which will benefit surface
reconstruction. We can see the good performance in the
next section.

Figure 1. The procedure of algorithm

4. Experimental Results and Analysis

Applying the algorithm described above to the point
sets, we can get the RBF centers we desired. Fig.2
demonstrates the experimental result. The left point
set is original containing 10,111 points. Then the
number of points reduced gradually. The reduction
time corresponding to last three subsets in Fig.2 is
1.688s, 3.438s and 5.890s. For the sculpture data set,
we got three point subsets, and the center reduction
time is 14.484s, 38.703s and 92.906s, respectively.
Because the distances are computed in whole kd-tree
cell, so for the large number of points set, such as the
skeleton hand set, the computation time becomes
longer. Therefore, to improve the efficiency of our
scheme will be the future work. The particular
thinning method prefers to remove data points in flat
region of the surface, while keeping data points in
high curvature areas.

Then we reconstruct the surface from the point sets
that have been progressed by our method with RBF
implicit, shown in Fig.3, Fig.4 Fig.5, and Fig.6
represent the figures of man-head, sculpture, house and
Venus-half separately. In every figure, it shows four
images that are reconstructed from four different point
sets. The first is reconstructed from original points, and
then we cannot find any visual difference among these
four images, and the same conclusion can get from the
quantitative error analysis, which is shown in the Fig.7.

Figure 2. This is man-head model, the first is the original
point set contains 10,111 points, the second 6,777 points, the
third contains 5,105 points, the final contains 3,371 points.

Figure 3. The model of man-head, the first is reconstructed
from 10,111 points, the second is reconstructed from 6,777
points, the third is reconstructed from 5,105 points, the final
is reconstructed from 3,371 points.

Figure 4. The model of sculpture, the first is reconstructed
from 25,382 points, the second is reconstructed from 17,053
points, the third is reconstructed from 12,848 points, the final
is reconstructed from 8,461 points.

Figure 5. The model of house, the first is reconstructed
from 27,478 points, the second is reconstructed from 17,831
points, the third is reconstructed from 12,564 points, the final
is reconstructed from 8,292 points.

Figure 6. The model of Venus-half, the first is reconstructed
from 24,000 points, the second is reconstructed from 15,892
points, the third is reconstructed from 11,458 points, the final
is reconstructed from 7,422 points.
In the Table 1, we compare results of computational time of surface reconstruction (Re-time) and the iso-surface reconstruction (Iso-time) of several models. The computation time is different for these models, and depends on the number of points, while the computation time decreases evidently as the number of points is reduced in each case. For example, the original points of the house model is 48,485, and the time of surface reconstruction and iso-surface reconstruction is 48.485s and 18.77s, when the number of points decreases to about 50%, the time drops to 24.349s and 5.329s, when the number of points decreases to about 33%, the time drops to 16.169s and 3.673s, which demonstrates the points reduction is very helpful to save the computation time. Moreover, Table 1 also quantifies the distance errors of different surfaces, which is derived by comparing the difference between a couple of models. Here, we analyze the maximal distance errors (Max-error) and the mean distance errors (Mean-distance) which are measured by Metro tool, a publicly available package [17]. At the same time, we also show the Hausdorff distance (Haus-distance). For example, using the surface of Fig.6, by comparing the middle right with the left, the maximal distance error and the mean distance error are 0.1074 and 0.0048. By comparing the right with the left, the errors are 0.1782 and 0.0085.

### Table 1. Comparison of computation time, the Hausdorff distance and the error of max_distance and the mean_distance

<table>
<thead>
<tr>
<th>Model</th>
<th>Number of surface points</th>
<th>Re-time (s)</th>
<th>Iso-time (s)</th>
<th>Max-error</th>
<th>Mean-error</th>
<th>Haus-distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Man-head</td>
<td>Original 10,111</td>
<td>3.718</td>
<td>31.078</td>
<td>0.0656</td>
<td>0.0038</td>
<td></td>
</tr>
<tr>
<td></td>
<td>66% 6,777</td>
<td>2.188</td>
<td>21.453</td>
<td>0.0674</td>
<td>0.0046</td>
<td>0.0674</td>
</tr>
<tr>
<td></td>
<td>50% 5,105</td>
<td>1.562</td>
<td>15.704</td>
<td>0.0674</td>
<td>0.0046</td>
<td></td>
</tr>
<tr>
<td></td>
<td>33% 3,371</td>
<td>0.985</td>
<td>11.375</td>
<td>0.3593</td>
<td>0.0083</td>
<td>0.3593</td>
</tr>
<tr>
<td>Sculpture</td>
<td>Original 25,382</td>
<td>9.625</td>
<td>238.11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>66% 17,053</td>
<td>5.719</td>
<td>52.219</td>
<td>0.0471</td>
<td>0.0020</td>
<td>0.0471</td>
</tr>
<tr>
<td></td>
<td>50% 12,848</td>
<td>4.048</td>
<td>28.813</td>
<td>0.0467</td>
<td>0.0023</td>
<td>0.0467</td>
</tr>
<tr>
<td></td>
<td>33% 8,461</td>
<td>2.873</td>
<td>25.516</td>
<td>0.1348</td>
<td>0.0037</td>
<td>0.1348</td>
</tr>
<tr>
<td>House</td>
<td>Original 48,485</td>
<td>18.77</td>
<td>163.359</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>66% 33,371</td>
<td>9.097</td>
<td>101.485</td>
<td>0.084</td>
<td>0.0014</td>
<td>0.1084</td>
</tr>
<tr>
<td></td>
<td>50% 24,349</td>
<td>5.329</td>
<td>43.141</td>
<td>0.0911</td>
<td>0.0014</td>
<td>0.0911</td>
</tr>
<tr>
<td></td>
<td>33% 16,169</td>
<td>3.673</td>
<td>14.074</td>
<td>0.1133</td>
<td>0.0016</td>
<td>0.1133</td>
</tr>
<tr>
<td>Venushalf</td>
<td>Original 72,545</td>
<td>21.167</td>
<td>291.719</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>66% 49,925</td>
<td>12.64</td>
<td>336.797</td>
<td>0.1210</td>
<td>0.0058</td>
<td>0.1210</td>
</tr>
<tr>
<td></td>
<td>50% 36,287</td>
<td>10.839</td>
<td>297.546</td>
<td>0.1074</td>
<td>0.0048</td>
<td>0.1103</td>
</tr>
<tr>
<td></td>
<td>33% 23,339</td>
<td>6.156</td>
<td>98</td>
<td>0.1782</td>
<td>0.0085</td>
<td>0.1782</td>
</tr>
<tr>
<td>Skeleton</td>
<td>Original 32,732</td>
<td>365.64</td>
<td>1261.039</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>50% 167,741</td>
<td>87.338</td>
<td>707.156</td>
<td>0.0626</td>
<td>0.0012</td>
<td>0.0625</td>
</tr>
<tr>
<td></td>
<td>33% 122,554</td>
<td>57.34</td>
<td>683.359</td>
<td>0.0490</td>
<td>0.0012</td>
<td>0.0490</td>
</tr>
<tr>
<td></td>
<td>25% 82,656</td>
<td>21.31</td>
<td>385.094</td>
<td>0.0703</td>
<td>0.0014</td>
<td>0.0703</td>
</tr>
</tbody>
</table>

From the figures shown in table1, the error is very small, and the Fig.7 also demonstrates the same results. In Fig.7 there are two models, and we all compare the first image with the final image, and using different...
color to represent the errors. From these images, no evident distance error is taken on. Therefore, to reconstruct surface using comparable small number of centers also can get better performance comparing to the original. It also indicates from these experiments that the more the number of points is, the better results our algorithm can offer as revealed in the skeleton hand.

5. Conclusion

We have reviewed the surface reconstruction with radial basis function, and enhance it by a method of center reduction, also represent the whole procedure of the method. Our method bases on the viewpoint of [14], while we use another exchange condition (8) to perform the procedures and extended it to three dimensions. During the center reduction, we can not only reduce point uniformly, but also can get keep feature points. In addition, using k-d tree makes our approach adapt to process a large number of points efficiently. These all benefits for surface reconstruction by using the radial basis function, and the above stated experimental results also demonstrate the good performance while working with fewer data. Our method can be efficiently used in areas of virtual reality, game developing, film visual effects and archeology, etc. In the future, we can improve the performance of the center reduction algorithm presented, though it has achieved better reduction in the number of points, the reduction time also depends on the number of points. Moreover, the computation time of the surface reconstruction time is still not much better, so we hope that we can combine the algorithm with a better interpolation method using radial basis function, and get the fast and better quality of method of surface reconstruction method.

6. Reference