Implicit Surface Modeling Suitable for Inside/Outside Tests with Radial Basis Functions

Rongjiang Pan\textsuperscript{1}, Vaclav Skala\textsuperscript{2}

\textsuperscript{1}School of Computer Science and Technology, Shandong University, Jinan, China, panrj@sdu.edu.cn
\textsuperscript{2}Centre of Computer Graphics and Data Visualization, Department of Computer Science and Engineering, University of West Bohemia, Plzen, Czech Republic

Abstract

We describe a method for computing an implicit function that represents a surface by its zero level set, given a set of points scattered over the surface and associated with surface normal vectors. This implicit function is defined as a linear combination of compactly supported radial basis functions. Our method is suitable for testing whether a given point is interior or exterior to the surface, previously only associated with globally supported or globally regularized radial basis functions. We use a two-level interpolation approach. In the coarse scale interpolation, we choose basis function centers by a grid that covers the enlarged bounding box of the given point set and compute their signed distances to the underlying surface using local quadratic approximations of the nearest surface points. Then a fitting to the residual errors on the surface points and additional off-surface points is performed with fine scale basis functions. The final function is the sum of the two intermediate functions and is a good approximation of the signed distance field to the surface in the bounding box. Examples of surface reconstruction and set operations between shapes are provided.

1. Introduction

Reconstructing a surface from sample points is a powerful 3D modeling method in CAD, reverse engineering, cultural heritage protection and other applications. Recent improvements of shape acquisition techniques have made it easier to digitize a world object into large unorganized point sets. Approaches to surface reconstruction can be roughly divided into two classes: Delaunay based methods and implicit surface methods [11].

An implicit surface model is defined as the zero level set $f^{-1}(0)$ of an implicit function $f(x)$. The implicit function $f(x)$ has positive values outside the object and negative values inside the object, which is the convention in this paper. It is more suitable for collision detection and set operations between shapes than other surface representations, as the sign of $f(x)$ indicates whether a point $x$ is interior or exterior to the surface. Among various implicit surface methods, techniques based on Radial Basis Functions (RBF) have proven to be valid and equally expressive to others. For example, it can produce water-tight surfaces and does not impose constraints on the topological complexity of the input data.

The implicit representation with RBFs is a linear combination of basis functions. In order to find the set of weights, a linear system of equations need to be solved given constraints, centers and type of radial basis functions. There are generally three methods known to overcome its computational problem for large point sets. The first approach is FastRBF algorithm based on globally supported basis functions and Fast Multipole Method [1]. However, it is hard to implement. The second way is to divide a large point set into a number of small subdomains and the final function is obtained by blending intermediate implicit functions [6,15]. The third means uses compactly supported RBFs (CSRBF) to make the linear system sparse. The efficient method is easy to implement and has been used in recent publications [7,8]. Nevertheless, it has some problems easily seen in Figure 1. As the function is only defined in a small neighborhood of the sampled surface, CSRBF is sensitive to the quality of input data, and often yields surfaces with some
unwanted artifacts and spurious zero level sets. The sign of $f(x)$ is not consistently negative inside the surface and positive outside the surface. This causes no serious problems in polygonization and rendering of implicit surfaces since we can skip the cubes or tetrahedra far from the sample points and some extraneous zero level sets are also obscured by the surface. However, it is not suitable for collision detection and set operations between shapes.

To keep the advantages of CSRBF and overcome its problems, Ohtake et al. [8] use spatial down sampling to construct a coarse-to-fine hierarchy of point sets and fit hierarchical implicit functions to the residual errors of the previous level with basis functions of diminishing support. Although this approach is insensitive to the density of scattered points and can fill large holes in the surface, the sign of the implicit function is still not fully consistent with the underlining signed distance field to the surface. Using centers of basis functions selected among the vertices of the Voronoi diagram of the input data points, an approximation of the signed distance to the surface defined all around the sampled shape is achieved by Samozino et al [10]. However, it needs a user-defined budget of centers and its efficiency and scalability need to be improved. By regularizing multi-scale compactly supported basis functions, Walder et al. [17] propose a framework that leads to the desirable properties of implicit modeling with CSRBFs. To build the regularisers, a complicated expression has to be evaluated, which covers a whole page in the technical report from the group [18].

The goal of our work is to produce a good approximation of the signed distance to the surface inside the bounding box of the shape as depicted in Figure 1(c), while preserving the simplicity and efficiency of implicit surface interpolation with CSRBFs since these are more suitable for inside and outside tests. We use a two-level fitting approach. In the coarse scale approximation, we choose centers of basis functions by a grid that covers the enlarged bounding box of the given point set and compute their signed distances to the underlying surface using local quadratic approximations of the nearest surface points. Then a fitting to the residual errors on the surface points and additional off-surface points is performed with fine scale basis functions. The final function is the sum of the two intermediate functions and is a good approximation of the signed distance field to the surface inside the bounding box. Moreover, the zero level set $f(x) = 0$ of the implicit function $f(x)$ interpolates the input point set rather than approximation in references [10, 17].

Figure 1: (a) A surface reconstructed from 180 sample points depicted as black dots. (b) A planar slice that cuts the surface – the colors represents the function values and the black curve highlights its zero level. (c) The desired implicit function values of our methods. In (b) and (c), positive values are mapped to red and negative values to blue.

The paper is organized as follows: we explain the details of our algorithm in Section 2. Examples of surface reconstruction and set operations between shapes are given in Section 3. Some future works are discussed in Section 4.

### 2. Algorithm

In the RBF approach, given a set of $N$ 3D points $\mathcal{P} = \{p_1, p_2, \ldots, p_N\}$ and $N$ scalar values $\{f_1, f_2, \ldots, f_N\}$, we interpolate $\mathcal{P}$ by a function in the following form
\[ f(\mathbf{x}) = \sum_{j=1}^{N} w_j \phi_\sigma(\|\mathbf{x} - \mathbf{p}_j\|) + P(\mathbf{x}) \]  

such that  
\[ f(\mathbf{p}_i) = f_i, \quad i = 1, \ldots, N \]  

where \( \phi_\sigma(r) \) is an RBF function,  
\[ P(\mathbf{x}) = c_0 + c_1 x + c_2 y + c_3 z \]  
is a polynomial accounting for the linear and constant portions of \( f \), and \( w_j \) is the weight to be determined.

Let \( \mathbf{p}_i = (p_{i1}, p_{i2}, p_{i3}) \) and \( \phi_\sigma = \phi_\sigma(\|\mathbf{p}_i - \mathbf{p}_j\|) \).

With Equations (1), (2) and side conditions given by Equation (3),
\[
\sum_{i=1}^{N} w_i = 0, \quad \sum_{i=1}^{N} w_i f_i = 0, \\
\sum_{i=1}^{N} w_i p_{i1} = 0, \quad \sum_{i=1}^{N} w_i p_{i2} = 0, \quad \sum_{i=1}^{N} w_i p_{i3} = 0
\]
a linear system is built up as follows:

\[
\begin{bmatrix}
\phi_{o1} & \phi_{o2} & \ldots & \phi_{oN} & 1 & p_{11} & p_{12} & p_{13} & w_1 \\
\phi_{i1} & \phi_{i2} & \ldots & \phi_{iN} & 1 & p_{11} & p_{12} & p_{13} & w_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\phi_{o1} & \phi_{o2} & \ldots & \phi_{oN} & 1 & p_{N1} & p_{N2} & p_{N3} & w_N \\
1 & 1 & \ldots & 1 & 0 & 0 & 0 & 0 & c_0 \\
p_{11} & p_{12} & \ldots & p_{13} & 0 & 0 & 0 & 0 & c_1 \\
p_{N1} & p_{N2} & \ldots & p_{N3} & 0 & 0 & 0 & 0 & c_2 \\
p_{11} & p_{12} & \ldots & p_{13} & 0 & 0 & 0 & 0 & c_k \\
\end{bmatrix}
= 
\begin{bmatrix}
1 \n w_f \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
\]

whose solution gives the unknown weights \( w_j, j = 1, \ldots, N \) and \( c_k = 0, k = 0, 1, \ldots, 3 \).

Although in the following analysis we choose Wendland’s CSRBF \( \phi_\sigma(r) = \phi(r / \sigma) \) with \( C^2 \) continuity,
\[
\phi(r) = \begin{cases} 
(1-r)^4(4r+1), & 0 \leq r \leq 1 \\
r, & r > 1 
\end{cases}
\]

where \( \sigma \) is the support size, or scale of the basis function, our method is general enough to be applicable to any other kinds of CSRBF functions.

Since we use compactly supported radial basis functions, the linear system is symmetric, positive semi-definite and sparse, which can be efficiently stored and solved by a direct LU sparse solver or an iterative conjugate gradients stabilized method [9].

In implicit modeling with radial basis functions, for all the points sampled from a surface, the scalar values are zero, \( f_i = 0 \). To avoid the trivial solution \( w_j = 0, j = 1, \ldots, N \) and \( c_k = 0, k = 0, \ldots, 3 \), surface points \( \mathcal{P} \) are usually equipped with oriented normals \( \{\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_N\} \) and off-surface points are created along the normals.

In the following of this section, we present the main steps of our reconstruction algorithm, namely coarse scale interpolation and fine scale interpolation.

### 2.1. Coarse Scale Interpolation

The axis-aligned bounding box is one of the simplest ways for enclosing a shape. To test whether a point is inside or outside a surface, we first check it against the bounding box of the surface. If it is interior to the surface, we can further decide the sign of the implicit function. Thus, we limit our approximation of the signed distance to the surface into the enlarged axis-aligned bounding box of the sampled points.

The support size \( \sigma \) is estimated as recommended in [8]. We use an octree-based data structure of the input points with each leaf cell containing no more than eight points, and delete the leaf cells containing no points. We set \( \sigma \) equal to three fourth of the average diagonal length of the leaf cells. We enlarge the axis-aligned bounding box of the sample points \((x_{min}, y_{max}) \times (y_{min}, y_{max}) \times (z_{min}, z_{max})\) to \((x_{max} - 2\sigma, x_{max} + 2\sigma) \times (y_{max} - 2\sigma, y_{max} + 2\sigma) \times (z_{min} - 2\sigma, z_{max} + 2\sigma)\).

Then we put a grid of spacing \( 4\sigma \) on the entire region of the enlarged axis-aligned bounding box. For each grid cell that contains sample points, we subdivide it into eight subcells by a fine grid of spacing \( 2\sigma \). If the subcell contains points, we further decompose it into eight smaller subcells of width \( \sigma \). The centers of cells and subcells are selected as the centers \( \{c_1, c_2, \ldots, c_k\} \) of basis functions as illustrated in Figure 2(a). We let the support size \( \sigma_i \) equal to three fourth of the diagonal length of the coarsest cell so that \( \sigma_i = 3\sqrt{3}\sigma \).

![Figure 2](image)
values and the black curve highlights its zero level. Positive values are mapped to red and negative values to blue.

In order to estimate the function value on these centers, for each center \( c_i \) we find the nearest input point \( p_j \in P \). If the distance between \( c_i \) and \( p_j \) is not zero, we fit a local quadratic approximation \( g_j(x) \) of \( P \) at \( p_j \) to its neighborhood \( \{ x \in P : \| x - p_j \| < \sigma \} \) as described in [8].

Let \((u,v,w)\) be the local orthogonal coordinate system with the origin of coordinates at \( p_j \) such that the positive direction of \( w \) coincides with the normal \( n_j \) of \( p_j \). The coefficients of the local fitting function \( g(u,v,w) = w - (Au^2 + 2Buv + Cv^2 + Du + Ev + F) \) are determined by the following minimization procedure

\[
\min \sum_k \phi_k(\|p_k - p_j\|g(u_k,v_k,w_k))^2,
\]

where \((u_k,v_k,w_k)\) are the local coordinates of \( p_k \in P \). The solution of this least-squares problem can be obtained by the singular value decomposition approach [9].

We choose to use Taubin’s first-order approximation \( d_i = \frac{g(u_i,v_i,w_i)}{\|\nabla g(u_i,v_i,w_i)\|} \) of the signed distance from center \( c_i \) to the underlying surface [14], where \((u_i,v_i,w_i)\) are the local coordinates of \( c_i \).

Since \( d_i \) is only meaningful in the nearby of the local quadratic fitting function \( g(u,v,w) \), we use \((c_i - p_j)^Tn_j\) as the signed distance from center \( c_i \) to the underlying surface if \( \|c_i - p_j\| < |d_i| \).

Thus, we get a coarse scale approximation \( f^1 \) of the input points,

\[
f^1(x) = \sum_{j=1}^K w_j \phi_{c_j}(\|x - c_j\|) + P(x), \quad (6)
\]

by solving the linear system

\[
\begin{bmatrix}
\phi_{11} & \phi_{21} & \cdots & \phi_{K1} & 1 & c_1^1 & c_1^2 & c_1^3 & c_1^4 & w_1 & d_1 \\
\phi_{12} & \phi_{22} & \cdots & \phi_{K2} & 1 & c_2^1 & c_2^2 & c_2^3 & c_2^4 & w_2 & d_2 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\phi_{1K} & \phi_{2K} & \cdots & \phi_{KK} & 1 & c_K^1 & c_K^2 & c_K^3 & c_K^4 & w_K & d_K \\
1 & 1 & \cdots & 1 & 0 & 0 & 0 & 0 & c_0 & 0 & 0 \\
c_1^1 & c_1^2 & \cdots & c_1^K & 0 & 0 & 0 & 0 & c_1 & 0 & 0 \\
c_2^1 & c_2^2 & \cdots & c_2^K & 0 & 0 & 0 & 0 & c_2 & 0 & 0 \\
c_3^1 & c_3^2 & \cdots & c_3^K & 0 & 0 & 0 & 0 & c_3 & 0 & 0 \\
c_K^1 & c_K^2 & \cdots & c_K^K & 0 & 0 & 0 & 0 & c_K & 0 & 0
\end{bmatrix}
\begin{bmatrix}
w_1 \\
w_2 \\
\vdots \\
w_K \\
c_0 \\
c_1 \\
c_2 \\
c_3 \\
c_K \\
c_1^1 \\
c_1^2 \\
c_1^3 \\
c_1^4 \\
c_K^1 \\
c_K^2 \\
c_K^3 \\
c_K^4
\end{bmatrix}
= \begin{bmatrix}
d_1 \\
d_2 \\
\vdots \\
d_K \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}, \quad (7)
\]

where \( \phi_{ij} = \phi_{c_j}(\|c_i - c_j\|), \quad i, j = 1, \ldots, K \) and \( \{c_1, c_2, \ldots, c_K\} \) is the set of selected centers.

Figure 2(b) illustrates the approximate signed distance field of input points in Figure 2(a).

2.2. Fine Scale Interpolation

After getting the coarse scale approximation of the input points, we interpolate the points by fine scale radial basis functions with support size \( \sigma \).

If there is no center of the coarse scale approximation in some thin parts of the point set surfaces, the approximate signed distance field is not correct in these regions, as shown in Figure 3 (a) and (b).

Figure 3: (a) Selected centers of the coarse scale approximation are visualized as red spheres. The input sample points depicted as black dots. (b) A planar slice that cuts the coarse scale approximation. Note the distance
function is not correct in the thin parts. (c) A planar slice that cuts the fine scale interpolation with surface points. The extra zero-level set is observed. (d) A planar slice that cuts the fine scale interpolation with surface points and off-surface points. In (b), (c) and (d), the colors represent the function values and the black curve highlights its zero level. Positive values are mapped to red and negative values to blue.

**Observation 1** The thin parts of the model, which cannot be correctly approximated by coarse scale centers, must be included in a subcell of size $\sigma$. Residuals of surface points are usually large in these thin parts than in other parts.

Therefore, we need more constraints in the thin parts. In addition to the surface points constraints, we use additional off-surface points to solve this problem.

First we calculate the residual $r_i = f_i - f^\dagger(p_i)$ at each input point $p_i \in \mathcal{P}$. If the absolute error $|r_i|$ exceeds a threshold $\varepsilon \geq 0$, we add an off-surface points along its normal $n_i$ according to the following bisection algorithm. In our implementation, we set $\varepsilon$ equal to the average absolute residual, $\varepsilon = \frac{1}{N} \sum_{i=1}^{N} |r_i|$.

**Algorithm 1**

**Input:** a surface point $p_i \in \mathcal{P}$ whose absolute error $|r_i|$ exceeds a threshold $\varepsilon \geq 0$.

**Output:** an off-surface point $q$ and its scalar value $r$.

1. $\text{distHigh} \leftarrow 0.5 \times \sigma$
2. $\text{distLow} \leftarrow 0.0$
3. if $(r_i > \varepsilon)$ then
   1. $q \leftarrow p_i - \text{distHigh} \times n_i$
   2. else $q \leftarrow p_i + \text{distHigh} \times n_i$
4. end if
5. if $(p_i \in \mathcal{P}$ is not the nearest point of $q$) then
   1. $\text{distMiddle} \leftarrow (\text{distHigh} + \text{distLow})/2$
   2. if $(r_i > \varepsilon)$ then
      1. $q \leftarrow p_i - \text{distMiddle} \times n_i$
   2. else $q \leftarrow p_i + \text{distMiddle} \times n_i$
   3. end if
5. else if $(p_i \in \mathcal{P}$ is the nearest point of $q$) then
   1. $\text{distLow} \leftarrow \text{distMiddle}$
   2. else
   3. $\text{distHigh} \leftarrow \text{distMiddle}$
   4. end if

After solving the linear system

$$
\begin{bmatrix}
\phi_{11} & \phi_{12} & \cdots & \phi_{1M} & 1 & p_1 & p_1 & p_1 & \cdots & w_1 & w_2 & \cdots \\
\phi_{21} & \phi_{22} & \cdots & \phi_{2M} & 1 & p_2 & p_2 & p_2 & \cdots & w_1 & w_2 & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\phi_{M1} & \phi_{M2} & \cdots & \phi_{MM} & 1 & p_M & p_M & p_M & \cdots & w_1 & w_2 & \cdots \\
\end{bmatrix}
\begin{bmatrix}
\sigma \\
\sigma \\
\vdots \\
\sigma \\
\end{bmatrix}
= \begin{bmatrix}
\sigma \\
\sigma \\
\vdots \\
\sigma \\
\end{bmatrix}
$$

where $\phi_{ij} = \phi_{ij}(|p_i - p_j|)$, $i, j = 1, \cdots, M$ and $\{p_1, p_2, \cdots, p_M\}$ is the set of surface points and additional off-surface points, we get the final scale interpolation $f$ of the input points,

$$f(x) = f^\dagger(x) + \sum_{j=1}^{M} w_j \phi_{\sigma}(\|x - p_j\|) + P(x).$$

The effects of fine scale interpolation with and without additional off-surface points are given in Figure 3 (c) and (d), respectively.

**3. Results**

In our implementation, we used the TAUCS library [16] to solve the linear system. To evaluate the fitting accuracy, we used the Peak Signal to Noise Ratio (PSNR) as suggested in [4]. The PSNR is defined as

$$PSNR[dB] = 20 \log_{10} \frac{\text{peak}}{d},$$

where peak is the diagonal length of the model’s bounding box and $d$ is the average of algebraic sum of the Taubin distances [14],

$$d = \frac{1}{N} \sum_{i=1}^{N} \left[ f(p_i) \right].$$

The details of our reconstruction method for five models obtained from the AIM@SHAPE repository are listed in Table 1 and the reconstructed surfaces are
shown in Figures 4-8. Although more sophisticated methods can be used to extract these surfaces, e.g. [2,3], we use the code published by Lewiner [5] in order to compare the result with others and to be sure that there are no extra zero level-set lying away from the data.

To demonstrate that implicit models reconstructed with our approach are suitable for set operations, we perform the boolean operations union, intersection and difference implemented in Visualization Toolkit (VTK) [12,13]. The results of the set operations between two models are shown in Figure 9.

As demonstrated in Figure 10, our approach also repairs large holes in the input data gracefully.

5. Summary

We have presented a fast and simple approach to reconstruct implicit surface with CSRBFs suitable for inside/outside testing, previously only obtained by globally supported or globally regularized radial basis functions. By combining a coarse scale approximation and a fine scale interpolation, the signed distance field is correctly approximated inside the enlarged bounding box of the model.

Compared with the down sampling multi-scale method [8], our approach builds a more accurate coarse scale approximation based on local quadratic approximations of surface points and eliminates small extra zero level-sets by additional off-surface points. We only have to solve two sparse linear systems and do not require to assemble matrices like $A^T A$ for a large sparse matrix $A$, which is a highly time consuming task as timing results in both [10] and [17].

In our future work, we will investigate how to employ fewer centers in the coarse scale and fine scale interpolation based on the geometric features and local sampling density of the model.

Acknowledgments

The authors thank their colleagues at Shandong University and University of West Bohemia for their critical comments and suggestions. This work has been supported by the international exchange scholarship between China and Czech governments and by the project VIRTUAL 2C06002 Ministry of Education of the Czech Republic.

References


Table 1. Results of our method for five models. Column two is the number of sample points. Column three and four are numbers of centers in the coarse scale approximation and fine scale interpolation, respectively. PSNR is given in the five column. The remaining columns are timing results with a 1.80 GHz Pentium 4 processor. Tcoarse is the time cost by coarse scale interpolation and Tfine is the time cost by fine scale interpolation. Ttotal is the total fitting time.

<table>
<thead>
<tr>
<th>Model</th>
<th>Points</th>
<th>#Centers1</th>
<th>#Centers2</th>
<th>PSNR(dB)</th>
<th>Tcoarse</th>
<th>Tfine</th>
<th>Ttotal</th>
</tr>
</thead>
<tbody>
<tr>
<td>2Torus</td>
<td>4352</td>
<td>884</td>
<td>6231</td>
<td>169.09</td>
<td>0.98</td>
<td>2.58</td>
<td>3.56</td>
</tr>
<tr>
<td>Knot</td>
<td>28659</td>
<td>10640</td>
<td>39802</td>
<td>123.97</td>
<td>11.84</td>
<td>44.13</td>
<td>55.97</td>
</tr>
<tr>
<td>Bunny</td>
<td>34835</td>
<td>14903</td>
<td>48842</td>
<td>189.79</td>
<td>17.86</td>
<td>54.06</td>
<td>71.92</td>
</tr>
<tr>
<td>Hand</td>
<td>36616</td>
<td>11064</td>
<td>46258</td>
<td>188.25</td>
<td>14.54</td>
<td>54.06</td>
<td>68.60</td>
</tr>
<tr>
<td>Armadillo</td>
<td>165954</td>
<td>77595</td>
<td>228626</td>
<td>139.25</td>
<td>110.14</td>
<td>460.33</td>
<td>570.50</td>
</tr>
</tbody>
</table>

Figure 4: Reconstruction of 2Torus. Left: coarse scale approximation; Middle: fine scale interpolation; Right: planar slice that cut the models. The colors represent the function values (red for positive value, blue for negative value) and the black curve highlights its zero level, which has the same meaning in the following figures.

Figure 5: Reconstruction of Knot. Left: coarse scale approximation; Middle: fine scale interpolation; Right: planar slice that cut the models.

Figure 6: Reconstruction of Bunny. Left: coarse scale approximation; Middle: fine scale interpolation; Right: planar slice that cut the models.
Figure 7: Reconstruction of Hand. Left: coarse scale approximation; Middle: fine scale interpolation; Right: planar slice that cut the models.

Figure 8: Reconstruction of Armadillo. Left: coarse scale approximation; Middle: fine scale interpolation; Right: planar slice that cut the models.

Figure 9: (a) Three models used for boolean operations. The three columns in (b) and (c) are the results of union, intersection and difference operations, respectively.