A Multi-scale Approach to 3D Scattered Data Interpolation with Compactly Supported Basis Functions

Yutaka Ohtake Alexander Belyaev* Hans-Peter Seidel

Computer Graphics Group, Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany
E-mails: {ohtake,belyaev,hpseidel}@mpi-sb.mpg.de

1 Introduction

Since the pioneering works of Ricci [32] and Blinn [4] geometric modeling with implicit surfaces remains to be an active research area [7]. Recent developments in this field include level set methods [34], variational implicit surfaces [33, 35, 36], and adaptively sampled distance fields [17]. Novel trends in implicit surface modeling are closely related to interpolating and approximating point set surfaces using level set methods [40], via Radial Basis Functions (RBFs) [9, 12, 11, 26], and by Moving Least Squares (MLS) [2, 15, 30], see also references therein. As demonstrated in [9, 10], implicit surfaces are especially useful for repairing incomplete data since no topological constraints are required.

Interpolation and approximation of scattered data with RBFs has a variational nature [18] which supplies a user with a rich palette of types of radial basis functions. The basic question is whether to choose local or global RBFs.

Fitting scattered data by local, compactly supported, RBFs leads to a simpler and faster computation procedure, while a practical usage of global RBFs is based on sophisticated mathematical techniques such as the fast multipole...
method employed in [9]. On the other hand, global RBFs are extremely useful in repairing incomplete data [9] while approaches based on compactly supported RBFs are sensitive to the density of interpolated/approximated scattered data and, therefore, a careful selection of RBF influence domains controlled by certain parameters is required.

A promising way to combine advantages provided by locally and globally supported basis functions consists of using locally supported basis functions in a hierarchical fashion. To the best of our knowledge, a multi-scale approach to fitting range data with bump-like basis functions was first used in [27]. At present hierarchical methods for scattered data fitting quickly gain popularity in computational mathematics and computer graphics research societies. For example, recently an RBF-based multilevel approach to scattered height data interpolation was employed in [27] (see also [1, 20] for very recent developments in this area), hierarchical Gaussians were used in [24] for reconstruction and modification of motion and image data. In [23] it was demonstrated that, chosen an appropriate carrier implicit surface, scattered data fitting with locally supported RBFs can be done very fast. Thus a hierarchical approach with locally supported basis functions where data reconstructed at coarser levels serve as carriers for finer levels may substantially accelerate scattered data fitting.

The approach developed in this paper is an attempt to integrate the best aspects of 3D scattered data fitting with locally and globally supported basis functions. We use compactly supported functions to interpolate a given 3D point set surface in a hierarchical way. Employing locally supported functions leads to an efficient computational procedure, while a coarse-to-fine hierarchy makes our method insensitive to the density of scattered data and allows us to restore large parts of missed data. We propose to use a new type of compactly supported basis functions: quadrics multiplied by compactly supported radial weights, where the quadric coefficients are determined via local weighted least squares fitting and via a global interpolation procedure. Given a point cloud distributed along a surface, we first use spatial down sampling to construct a coarse-to-fine hierarchy of point sets. Then we interpolate the sets starting from the coarsest level. We interpolate a point set of the hierarchy by an offset of the interpolating function computed at the previous level. Numerical experiments suggest that our method is essentially faster than the state-of-art scattered data approximation with globally supported RBFs [9]. In addition, our approach is much simpler to implement than that developed in [9].

Fig. 2 demonstrates a reconstruction of an incomplete data by our multi-scale scattered data interpolation procedure. We smoothed slightly the angel mesh data from Caltech 3D Gallery [8] and then removed all connectivity information.

The rest of the paper is organized as follows. In Section 2 we explain our scattered data interpolation procedure at a single level. In Section 3 we present a multi-level interpolation scheme. We demonstrate and discuss advantages and limitations of our approach in Section 4 and conclude in Section 5.

2 Single-level Interpolation

In this section we demonstrate how our scattered data interpolation procedure works at a single level.

Let us consider a set of \( N \) points \( P = \{ p_i \} \) scattered along a surface. We assume that the points are equipped with inner unit normals \( n_i \) defining an orientation. The normals are usually computed during the shape acquisition stage from range images. They can also be estimated directly from point set data [19]. We want to generate a 3D...
scalar field \( f(x) \) such that its zero level-set \( f = 0 \) interpolates \( \mathcal{P} \). Implicit surface \( f(x) = 0 \) separates the space into two parts: \( f(x) > 0 \) and \( f(x) < 0 \). Let us assume that the orientation normals are pointing into the part of space where \( f(x) > 0 \). Thus \( f(x) \) has negative values outside the surface and positive values inside the surface.

We interpolate \( \mathcal{P} \) by “function-valued” RBFs

\[
f(x) = \sum_{p_i \in \mathcal{P}} \psi_i(x) = \sum_{p_i \in \mathcal{P}} [g_i(x) + \lambda_i] \psi(x - p_i),
\]

where \( \lambda_i = \lambda \) is Wendland’s compactly supported RBF [38], \( \sigma \) is its support size, and \( g_i(x) \) and \( \lambda_i \) are unknown functions and coefficients to be determined. An appropriate value of \( \lambda \) is estimated from the density of \( \mathcal{P} \). The functions \( g_i(x) \) and coefficients \( \lambda_i \) are chosen via the following two-step procedure.

1. At each point \( p_i \), we define a function \( g_i(x) \) such that its zero level-set \( g_i(x) = 0 \) approximates the shape of \( \mathcal{P} \) in a small vicinity of \( p_i \).

2. We determine the coefficients \( \lambda_i \) from the interpolation

\[
f(p_j) = 0 = \sum_{p_i \in \mathcal{P}} [g_i(p_j) + \lambda_i] \psi(x - p_i).
\]

Notice that (2) can be rewritten as

\[
\sum_{p_i \in \mathcal{P}} \lambda_i \Phi_{ij} = - \sum_{p_i \in \mathcal{P}} g_i(p_j) \Phi_{ij}, \quad \Phi_{ij} = \psi(x - p_i)
\]

and therefore (2) leads to a sparse system of linear equations with respect to \( \lambda_i \). Since Wendland’s compactly supported RBFs are strictly positive definite [38], the \( N \times N \) interpolation matrix \( \Phi = \{ \Phi_{ij} \} \) is positive definite if \( \mathcal{P} \) consists of pairwise distinct points.

For each point \( p_i \in \mathcal{P} \) we determine a local orthogonal coordinate system \( (u, v, w) \) with the origin of coordinates at \( p_i \) such that the plane \( (u, v) \) is orthogonal to \( n_i \) and the positive direction of \( w \) coincides with the direction of \( n_i \). We approximate \( \mathcal{P} \) in a vicinity of \( p_i \) by a quadric

\[
w = h(u, v) \equiv Au^2 + 2Buv + Cv^2,
\]

where the coefficients \( A, B, \) and \( C \) are determined via the following least-squares minimization

\[
\sum_{(u_j, v_j, w_j) = p_j \in \mathcal{P}} \phi(x - p_i) \right( w - h(u_j, v_j) \right)^2 \rightarrow \min.
\]

Now we set

\[
g_i(x) = w - h(u, v).
\]

Thus the zero level-set of \( g_i(x) \) coincides with the graph of \( w = h(u, v) \).

A geometric idea behind our interpolating procedure is illustrated in Fig. 3. Fig. 4 shows the graph of the 2D version of a basic function

\[
\psi_i(x) = [g_i(x) + \lambda_i] \psi(x - p_i),
\]

a summand in (1).}

![Fig. 3. Geometric idea behind our approach for scattered point data interpolation at a single level.](image)

![Fig. 4. Graph of 2D version of basic function \( \psi_i(x) \) used in (1). Zero level \( \psi_i(x) = 0 \) (parabola) is drawn by bold line.](image)

Parameter \( \sigma \), the support size of \( \phi(x) \), is estimated from the density of \( \mathcal{P} \). We start an octree-based subdivision of a bounding box of \( \mathcal{P} \) and stop the subdivision if each leaf cell contains no more than 8 points of \( \mathcal{P} \). Then we compute the average diagonal of the leaf cells. Finally we set \( \sigma \) equal to three fourth of that average diagonal.

To solve the linear system corresponding to (2) we use the preconditioned biconjugate gradient method [31] with the initial guess \( \lambda_i = 0 \). The size of the linear system is \( N \times N \), where \( N = |\mathcal{P}| \) is the number of interpolating points. Note that methods developed in [33, 35, 26, 9, 36] require also interior/exterior constraints which together with interpolation conditions \( f(p_i) = 0 \) lead to a bigger system of linear equations.

Basis functions (4) used in (1) are similar to the surflets introduced by Perlin [13].
Fig. 5. The Stanford bunny and Igea model reconstructed from scattered point data as polygonized zero level-sets of (1). Fitting time is 6 seconds for the Stanford bunny (35K points) and 47 seconds for the Igea model (134K points). After rescaling both the models in order to fit them into a unit cube we use $\sigma = 0.02$ and $\sigma = 0.0125$ for the Stanford bunny and Igea model, respectively.

As demonstrated in Fig. 5, the above interpolation procedure is quite fast. However using compactly supported basis functions implies several essential limitations.

- It has no ability of repairing incomplete data, in particular interpolating irregularly sampled data (see Fig. 6) and filling holes (see the right image of of Fig. 7). Enlarging the support size parameter $\sigma$ in order to fix these drawbacks slows down the reconstruction process essentially.

- The interpolating implicit surface has a narrow band support (the left image of Fig. 7). It requires, for example, the polygonization grid to be smaller than the width of the support band.

Fig. 6. Left: Irregularly sampling points on the belly part of the Stanford Buddha. Right: Mesh generated from the zero level-set of the single-level compactly supported implicit function.

Implementing a multi-scale interpolation procedure described in the next section eliminates these problems.

Fig. 7. Left: The bottom part of the Stanford bunny shown the left image of Fig. 5. The holes are not filled. Right: the white region indicates the support of the implicit function used to reconstruct the right image of Fig. 5.

3 Multi-level Interpolation

To overcome problems mentioned at the end of the previous section we build a multi-scale hierarchy of point sets $\{\mathcal{P}^1, \mathcal{P}^2, \ldots, \mathcal{P}^M = \mathcal{P}\}$ and interpolate a point set $\mathcal{P}^{m+1}$ of the hierarchy by offsetting the interpolation function used in the previous level to interpolate $\mathcal{P}^m$. Fig. 8 demonstrates the main steps of our multi-level interpolation approach.

3.1 Construction of Point Set Hierarchy

To construct the multi-scale hierarchy of point sets $\{\mathcal{P}^1, \mathcal{P}^2, \ldots, \mathcal{P}^M = \mathcal{P}\}$ we first fit $\mathcal{P}$ into a parallelepiped and then subdivide it and its parts recursively into eight equal octants. Point set $\mathcal{P}$ is clustered with respect to the cells of the built octree-based subdivision of the parallelepiped. For each cell we consider the points of $\mathcal{P}$ contained in the cell and compute their centroid. A unit normal assigned to the centroid is obtained by averaging the normals assigned to the points of $\mathcal{P}$ inside the cell and normalizing the result. Set $\mathcal{P}^1$ corresponds to the subdivision of the bounding parallelepiped into eight equal octants.

3.2 Multi-level Interpolation via Offsetting

After constructing hierarchy $\{\mathcal{P}^1, \mathcal{P}^2, \ldots, \mathcal{P}^M = \mathcal{P}\}$, our multi-level interpolation procedure proceeds in the coarse-to-fine way. First we define a base function

$$f^0(x) = -1$$

and then recursively define the set of interpolating functions

$$f^k(x) = f^{k-1}(x) + o^k(x) \quad (k = 1, 2, \ldots, M),$$

where $f^k(x) = 0$ interpolates $\mathcal{P}^k$. An offsetting function

$$o^k(x) = \sum_{\mathcal{P}^k \subset \mathcal{P}} \left[ g^k(x) + \lambda^k \right] \phi_{\sigma^k}(\|x - \mathcal{P}^k\|).$$
Fig. 8. Multi-scale interpolation overview. We use a monk model (60K points) obtained as a laser scanner data. Top row: multi-scale hierarchy of points where the radii of the spheres at each level $k$ are proportional to $\sigma_k$, the support size of RBFs used for the interpolation (the actual sizes the spheres are five times larger than that used for visualization). Middle row: interpolating implicit surfaces polygonized at each level of the hierarchy. Bottom row: cross-sections of the interpolating; the bold black lines correspond to the zero level sets of the functions.

has the form used in the previous section for the single-level interpolation. In particular, local approximations $g^k_i(x)$ are determined similar to (3) via least square fitting applied to $\mathcal{P}^k$. The shifting coefficients $\lambda^k_i$ are found by solving the following system of linear equations

$$ f^{k-1}(p^k_i) + \sigma^k(p^k_i) = 0. \tag{5} $$

Similar to the single-level interpolation case we use the preconditioned biconjugate gradient method [31] to determine $\lambda^k_i$.

The support size $\sigma^k$ is defined by

$$ \sigma^{k+1} = \frac{\sigma^k}{2}, \quad \sigma_1 = cL, $$

where $L$ is the length of a diagonal of the bounding paral-
lelogram and the parameter $c$ is chosen such that an octant of the bounding box is always covered by a ball of radius $\sigma_1$ centered somewhere in the octant. In practice we use $c = 0.75$.

Finally, the number of subdivision levels $M$ is determined by $\sigma_1$ and $\sigma_0$ where $\sigma_0$ is the support size for the single-level interpolation. According to our experience, $M = \lceil -\log_2 (\sigma_0/(2\sigma_1)) \rceil$ produces good results.

4 Results and Discussion

The developed method can be applied to point set surfaces consisting of several hundreds thousand points on standard PCs. All examples presented in this paper were computed on a Pentium 4M 1.6 GHz PC.

Fig. 9 presents results of interpolating and approximating a large and complex point set surface, the Stanford Buddha model. The left image shows the result of the complete interpolation of the original Stanford Buddha data while the right image demonstrates an incomplete fitting procedure: only $(M-1)$ levels of the multi-scale hierarchy of point sets were used. No visual difference between the left and right images is observed.

Fig. 10 shows a result of interpolation of irregularly sampled points. First the right part of the original Igea mesh was 90\% decimated, then all connectivity information of the model was removed. Notice that the sharp drop of the sampling density produce no visual artifacts in the implicit surface reconstructed by our method from the cloud of points.

**Comparison with FastRBF** In Fig. 11 we compare our approach with the Fast RBF method [9]. For the comparison we use a free version of FastRBF toolbox, release 1.2, available from [14]. We use the toolbox with the -direct-accuracy=0.25 options which mean that a RBF center reduction procedure is not applied and approximation accuracy is equal to 0.25. According to Fig. 11, there is no visual difference in the Hand model reconstruction by the Fast RBF method [9] and our multi-scale fitting technique. However our approach works approximately four times faster then the Fast RBF method. FastRBF works approximately two times faster with the center reduction procedure than without it for the Hand model. Our method is still faster, moreover it can be accelerated if a similar center reduction procedure is implemented. Unfortunately we were unable to test FastRBF on more complex models because the free version of FastRBF has limited capabilities.

**Visualization** To visualize implicit surfaces we polygonize them. The models displayed in Fig. 1, Fig. 8 and Fig. 9 were polygonized by Bloomenthal’s method [5, 6]. Other models considered in the paper were polygonized by the

![Fig. 9. Polygonized implicit surfaces interpolating (left) and approximating (right) the Stanford Buddha point data (544K points). Nine levels of the point set hierarchy were generated. The left model was generated by multi-level fitting using all nine levels: computational time is 19.1 min., maximal RAM used is 332 Mb, the model is represented by the sum of 901K basis functions. The right model was generated using eight levels only: computational time is 7.5 min., maximal RAM used is 198 Mb, the model is represented by the sum of 362K basis functions.](image)

![Fig. 10. Interpolation of irregularly sampled data (73K points, 38 sec.).](image)
Fig. 11. To compare our method with FastRBF [9] we use the Hand model (13,348 points). Left: the surface is approximating by FastRBF (computational time is 30 sec., 26,696 RBFs are used to represent the model). Right: the surface is interpolated by our method (computational time is 7 sec., the model is represented by the sum of 18,647 basis functions).

dual-contouring method [21]. Of course, other polygonization methods such as the Marching Cubes [25] and extended Marching Cubes [22] can be used. As a postprocessing step we can also employ a method proposed in [28] in order to improve the mesh quality.

Extra zero-sets All the models considered in this paper except one displayed in the right image of Fig. 12 were interpolated and polygonized by inspecting their bounding boxes and no extra zero level sets were observed. However, if an interpolating point set surface has thin parts, extra zero level-sets may be generated near the surface, as seen in the right image of Fig. 12. These extra zero level-sets will not be polygonized if a polygonization procedure starts from seed points chosen on the interpolated point set surface, see [5] where an appropriate implicit surface polygonization procedure was developed. However extra zero level-sets may be harmful for the boolean operations with implicits. One possible way to solve this problem of extra zero level-sets is to use a more sophisticated downsampling procedure respecting topology and/or complex geometry of the interpolating point set surface. Some hints to solve this problem can be also found in [30].

Robustness with respect to quality of normals. The normals of a point set surface are either computed during a shape acquisition process or estimated directly from the points. The normals are more prone to noise than points themselves. According to our experience, our multiscale approach is quite robust with respect to quality of normals. In particular it is implied by a smoothing effect of the downsampling procedure we use.

If the normal associated with point $p_i$ is zero (sometimes it happens due to errors during the shape acquisition process) we can not decide the local shape orientation at $p_i$. A common way to handle such case within the standard RBF approach [35, 9, 26, 36] consists of not using normal constraints at $p_i$. Similarly we set $g_i(x) = 0$ if the normal at $p_i$ is zero.

Fig. 13 demonstrates robustness of our method with respect to the quality of normals.

Approximation vs. Interpolation If scattered data is noisy, approximation procedure is preferable over interpolation. This is why in we interpolate a smoothed version of the original angel data, see Fig. 2. Boundaries of range data are usually more corrupted by noise than inner parts. Thus it is reasonable to introduce a fidelity measure and use an approximation procedure which takes that measure into account.

5 Conclusion

We have presented a multi-scale approach to interpolating point set surfaces by implicit surfaces. Our method generate implicit solids that can be further used for morphing, surface carving and other implicit surface processing operations, as seen in Fig. 15. Our contribution is twofold. Besides a new scheme to build a hierarchy of 3D scattered datasets we have introduced a new type of compactly sup-

Fig. 12. Interpolation of point set surfaces representing complex topological objects. Left: no extra zero level-sets are generated. Right: extra zero level-sets are generated.
ported basis functions. The interpolation procedure developed in the paper demonstrates a good performance while working with irregularly sampled and/or incomplete data. Using compactly supported basis functions makes our approach faster than those based on globally supported basis functions.

In the future, we hope to improve our approach in order to handle large point set surfaces (several millions of points). We hope that our method can be easily adopted to scattered data approximation which is important for processing noisy scattered data [12, 9]. Scattered data approximation will also allow us to use fewer points [39, 9]. We are planning to combine our method with feature extraction procedures [3, 37, 29] in order to adapt it for processing very incomplete data, see Fig. 14 for our first steps in this direction. Reconstruction of scattered data with sharp features [11] also intrigues us in this area.

Fig. 15. CSG operations with implicit solids reconstructed using our approach. Left: torus is subtracted from bunny. Right: blended dragons.

Acknowledgments

We are indebted to Vladimir Savchenko who has read an early version of this paper and given us several valuable comments and to Alexander Pasko for a fruitful discussion. We are grateful to anonymous reviewers for their useful comments and suggestions. The angel model is from Caltech 3D Gallery [8]. The bunny, Buddha, and dragon models are courtesy of Stanford 3D scanning repository. The Igea and rabbit models are due to Cyberware. We are grateful to FarField Technology Ltd [14] for its free version of the FastRBF toolkit and the Hand model.

References


