Duplex Fitting of Zero-level and Offset Surfaces

Shengjun Liu[1,2] Charlie C.L. Wang[1*]

[1]Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong

[2]School of Mathematical Science and Computing Technology, Central South University

Abstract

Offset surfaces play an important role in various CAD/CAM applications. Given a set of oriented points, we propose a hierarchical method in this paper to fit both the zero-level surface and its offset surface with a single implicit function. The implicit function is formed by compactly supported radial basis functions (CSRBFs). Different from other existing methods in literature, our approach reconstructs an implicit function which interpolates or approximates both the zero-level surface and the offset surface of a given point set simultaneously. Employing locally supported functions leads to an efficient computational procedure, while the coarse-to-fine hierarchy makes our approach insensitive to the density of scattered data and allows us to reconstruct large parts of missing data. The performance of our method is demonstrated by a number of examples and the application of adaptive slicing hollowed models in rapid prototyping.

Keywords: surface reconstruction; implicit surface; offset surface; hollowed model; CAD/CAM

1. Introduction

Offset surfaces play a very important role in geometry processing and especially in various CAD/CAM applications. They can be used for rapid prototyping and tolerance analysis in machine processing. In the context of tool path generation for numerically controlled (NC) milling machines, offset surfaces are used to define the location surfaces of cutters where the machine tool's positions are constrained to lie. These location surfaces of cutters provide the input for collision-free path planning. Furthermore, offset surfaces are also used in generating hollowed or shelled versions of geometric solids, filleting and rounding of 3D models.

Rapid advancement of 3D capture technology encourages the growing need for geometric processing algorithms of point clouds, which can convert 3D cloudy points (captured by 3D data acquisition sensors) into suitable geometric models that are used in downstream applications of product design, analysis and manufacturing. Most existing approaches in literature devote to interpolating or approximating the incomplete point clouds using parametric surfaces, mesh surfaces or implicit surfaces. Few of them addressed the problem of fitting both the surface sampled by input cloudy points and its offset surface simultaneously.

Offset surfaces are generally created from surface models. Given a point cloud, we can first reconstruct the surface from it and then calculate the offset surface by the reconstructed surface. However, this two-step strategy suffers from all the problems in existing offset surface generation methods [3-8] and wastes a lot of computing time. In this paper, we present a novel method which reconstructs both the zero-level surface (i.e., the surface approximating or interpolating input clouds) and its offset surface using a single implicit function. The implicit representation in terms of Radial Basis Functions (RBF) is adopted here, which simplifies the reconstruction problems and offers an elegant computational framework. An RBF implicit surface offers a compact functional

*Corresponding author (Charlie C. L. Wang); e-mail: cwang@mae.cuhk.edu.hk; fax: (852) 2603 6002

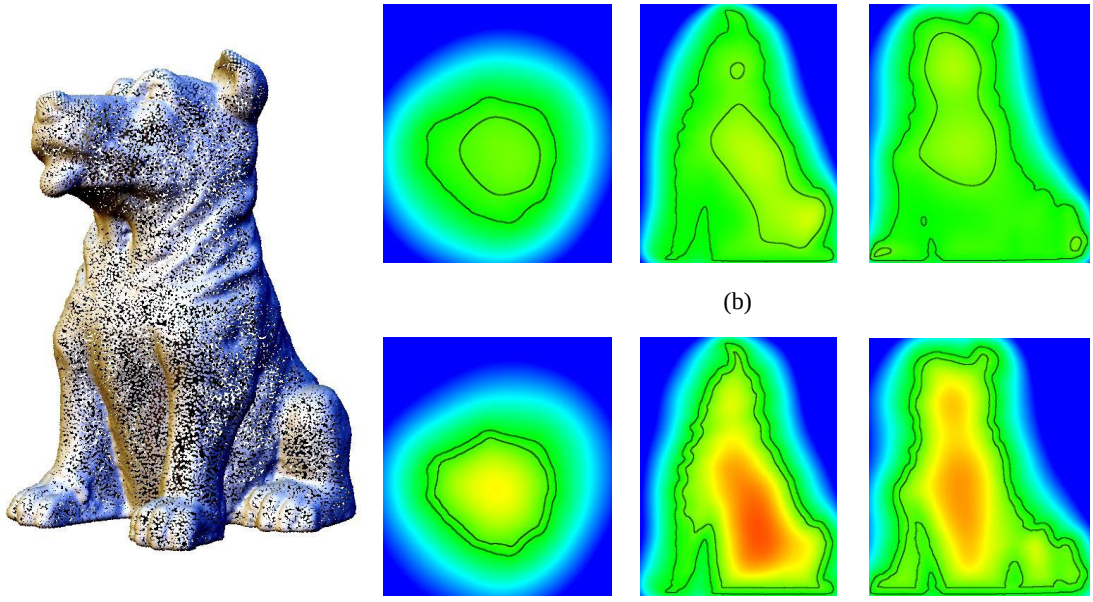
description for a set of surface data, where interpolation and extrapolation are inherent in such a functional representation. Gradients and higher derivatives can be determined analytically, and depending on the choice of basis function, they can be continuous and smooth. Moreover, surface normals can therefore be calculated reliably, and the isosurfaces extracted from the implicit RBF model are two-manifold.

Problem Definition Given a point set $\widetilde{P} = \{p_1, p_2, ..., p_n\}$ coupled with normal vectors $N = \{n_1, n_2, ..., n_n\}$ that is sampled from a surface P, letting P_δ denote the offset surface to P with distance δ, an implicit function $f(p)$ needs to be reconstructed that

$$\begin{cases} f(p_i) = 0, & p_i \in \widetilde{P} \\ f(p_j) = \delta, & p_j \in \widetilde{P}_\delta \end{cases},$$

where $\widetilde{P}_\delta$ approximates a set of points sampled from P_δ.

Different from other implicit surface reconstruction methods (e.g. [1, 2, 16-33]), which generate offset surface with $f(p) = \delta$ based on algebraic distance (actually hard to use in practice), the isosurface based on Euclidean distances is required in many applications. Figure 1 shows an example of the isosurface generated based on algebraic distance versus Euclidean distance. Colors are employed to present the function values where blue denotes negative maximum and red represents positive maximum.

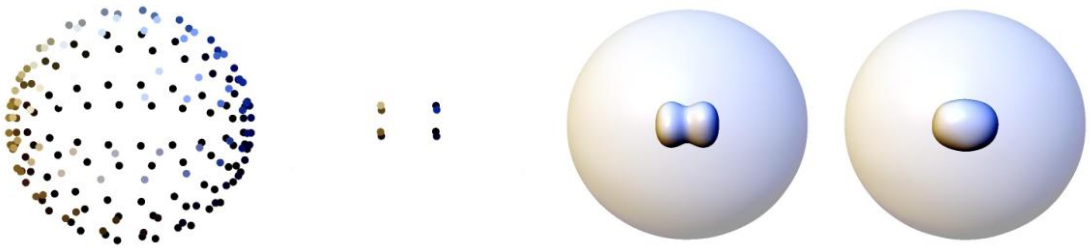


(a) (b) (c)

Figure 1. An example of implicit function generated from a dog model with 50k points: (a) the dog model which is scaled into a unit bounding box, (b) the isosurfaces with f(x,y,z)=0 and f(x,y,z)=0.025 generated by [2], and (c) the isosurfaces with f(x,y,z)=0 and f(x,y,z)=0.025 generated by our method. The cross sections are shown in the top view (the first column), the front view (the second column), and the right view (the third column).

One straightforward thinking is that: why don't we simply reconstruct an implicit surface by the points in $\widetilde{P}_\delta$ for the isosurface $f(p) = \delta$? This will be explained in more detail later (in section 4). As the offset points in $\widetilde{P}_\delta$ are generated from $\widetilde{P}$, when the value of offset is large, many of the points will be invalid and removed from $\widetilde{P}_\delta$. Then, there may be too few points left to describe the offset surface correctly if we reconstruct the offset surface directly from these points. Figure 2 shows a simple example for this, where the transparent surface is the zero-level surface of a given point set. The distances between zero level-set and corresponding offset surface are measured using [42]. For the reconstructed result in (c), the maximal distance from zero level-set to

offset surface is 0.404 and the mean distance is 0.371, while from offset surface to zero level-set, the maximal distance is 0.418 and the mean distance is 0.377. For reconstructing the result in (d), the corresponding distances are 0.374, 0.365, 0.374, 0.3655 respectively.



(a) (b) (c) (d)

Figure 2. Original surface and offset surface reconstruction: (a) The sphere model with 180 points, (b) The twelve offset points with δ = 0.365, (c) Surfaces reconstructed using [2] two times from the original points in (a) and the offset points in (b), and (d) The surfaces generated by our method with one single function.

The rest of this paper is structured as follows: In section 2 we review previous works in offset surface generation and implicit surface reconstruction. After that, we introduce RBFs and the multi-level CSRBFs interpolation method briefly in section 3. The details of duplex fitting of zero-level and offset surfaces are presented in section 4, where the compactly supported RBF (CSRBF) is employed to represent the zero-level and offset surfaces implicitly. Then, the experimental results and statistics are presented in section 5 followed by the application of adaptive slicing. After that, we conclude our paper in section 6.

2. Related Works

Our algorithm consists of two steps. First, we generate a set of offset points from the input point cloud. After that, we interpolate (or approximate) the original points and the offset points with a single implicit function. The related existing work in literature about the offset surface generation and the implicit surface based surface reconstruction will be reviewed below.

2.1. Offset surfaces

The offset surface of a solid H is a set of points having the same distance δ (i.e., offset distance) from the boundary surface of H. The mathematical basis for offsetting solids is described in the early work of Rossignac et al. in [3]. A number of methods for computing offset surfaces have been developed since then.

The offsetting operation can be considered as a special case of the Minkowski sum. Varadhan et al. in [4] have proposed a method to approximate the Minkowski sum of polyhedral models, which also covers the computation of offset surfaces as a special case. Some surface-based approaches generate offset surface by offsetting vertices into spheres, edges into cylinders, and faces into parallel faces, and combining those by trimming to get the final offset surface (ref. [3, 5]). This is a very computationally expensive process and the trimming at tangential intersections is numerically unstable. Some surface-based approaches for generating offset surfaces simply shift the original vertices in offset direction (ref. [6]). This is problematic as self-intersections can occur either locally in areas of high curvature or globally when different parts of the input surface meet during the vertex shifting. If the input mesh is convex or can be decomposed into convex pieces, the computation then becomes simple and effective (ref. [4]). However, such a decomposition is not easy for complex objects (ref. [7]). In [8], simplification envelopes were introduced for global error-control in mesh simplification. Their offset

surface generation method requires manifold meshes which do not contain any degenerated configurations, so it is not general.

There are many other methods for offset surface generation with the help of other representations (i.e., not B-rep). Volumetric methods were presented based on distance volumes and the fast marching method in [9, 10]. These methods work on regular voxel grids. Recently, a point-based offsetting approach in [11, 12] was introduced, in which point samples are first generated on the input surface and are then moved in normal direction. After that, the Minkowski sum volume is rasterized on voxel grids in order to remove self-intersections. In [13, 14], another offsetting approach was presented which aims mainly at visualizing the offset via surface splats. In their cases the classification whether a cell intersects the offset surface is based on conservative estimates for both the minimum and maximum distances. While this is sufficient for visualization purposes, it would be non-trivial to extract a proper manifold offset surface from it. A hybrid method combining surface and volume was proposed in [15]. The algorithm is also based on computing the union of a set of primitives (spheres, cylinders and planes). Its computations are stable since the method works on a volumetric representation and hence self-intersections can be easily removed by computing the min/max operations applied to distance functions. This method can process all kinds of mesh inconsistencies because it treats every triangle independently. Nevertheless, none of the above methods can be directly applied to generate offset surfaces P_δ for the input point cloud $\tilde{P} = \{\mathrm{p}_1, \mathrm{p}_2, \ldots, \mathrm{p}_n\}$ that gives the discrete surface representation of surface P.

Offsetting is a very important operation in many CAD/CAM applications. In this paper, we generate the offset points from a set of oriented points $\tilde{P}$ by the normal shifting and the removal of invalid offset points. There may be holes in the set of offset points although the original points $\tilde{P}$ are good for representing the surface P. Therefore, in order to construct a complete offset surface from the generated offset points, we adopt an implicit surface reconstruction method to fit both the point set $\tilde{P}$ and the offset points generated from it.

2.2. Implicit surface reconstruction

Implicit surface is an important tool for surface reconstruction from point cloud. The existing approaches in literature usually define an implicit function in 3D to interpolate (or approximate) the input point samples and then extract the reconstructed surface as an isosurface of the function (e.g., [1, 2, 16-22]). The advantage of these approaches is two-fold: First, the extracted surface is guaranteed to be watertight, resulting in a model with well-defined interior and exterior, and second, the use of an implicit function does not place any restrictions on the topological complexity of extracted isosurfaces, providing a reconstruction algorithm that can be applied to a variety of 3D models.

Recent developments in geometric modeling with implicit surfaces include level set methods in [23], variational implicit surfaces in [22, 24], adaptively sampled distance fields in [25], and multi-level partition of unity implicit functions in [1]. New trends in implicit surface modeling are closely related to interpolating and approximating point set surfaces using level set methods (ref. [26]), via radial basis functions (RBFs) in [2,21,27-29], or by moving least squares (MLS) (ref. [30-32]). As demonstrated in [19, 20], implicit surfaces are especially useful for repairing incomplete data since no topological constraints are required. This is a reason why we adopt implicit method to reconstruct surface in our paper.

In general, modeling a surface implicitly with a function $f(x, y, z)$ can be stated as the following. If a surface P consists of all the points (x, y, z) that satisfy the equation

$$f(x, y, z) = 0,$$

we say that f implicitly defines P. Describing surfaces implicitly with various functions is a well-known

technique (ref. [33]). The distinction between our approach and these well-known techniques is that we wish to model the entire object with a single function which is continuous, differentiable and can satisfy both the above equation at zero-level and the equation $f(x, y, z) = \delta$ where the points (x, y, z) are on the offset surface with distance δ to P. Based on the best of our knowledge, there is no similar work in literature found.

3. Radial Basis Functions for Formulating Implicit Surfaces

Scattered data interpolation can be computed using radial basis functions centered at the constraints. A radial basis function (RBF) is a real-valued function whose value depends only on the distance from some center points c as $\phi(\mathrm{x},\mathrm{c}) = \phi(||\,\mathrm{x}\text{-}\mathrm{c}\,||)$. Euclidean distance is usually employed for the norm. Radial basis functions are typically adopted to form the interpolation function as

$$f(\mathrm{x}) = \sum_{i=1}^{N} w_i \phi(||\,\mathrm{x} - \mathrm{c}_i\,||), \tag{1}$$

where the interpolation function $f(\mathrm{x})$ is represented as a sum of N radial basis functions, each associated with a different center c_i, and weighted by a coefficient w_i. A first-degree polynomial can also be added for the linear and constant terms of f.

According to the support size, RBFs can be classified into two categories (ref. [19]): global and compactly supported RBFs. The traditional RBFs fitting method uses global RBF which is a real valued function on $[0,\infty)$, such as the thin-plate spline $\phi(r) = r^2 \log(r)$, the Gaussian $\phi(r) = \exp(-cr^2)$, the multiquadric $\phi(r) = \sqrt{r^2 + c^2}$, the biharmonic spline $\phi(r) = r$ and the triharmonic spline $\phi(r) = r^3$. However, when using this sort of global RBFs, the linear equation system for determining coefficients w_i in Eq.(1) is dense. Although the fast multipole method employed in [19] can speed up the computation, one has to face the fact that large data sets interpolation using these classical functions requires an unacceptable computation time. Therefore, fitting scattered data with CSRBFs leads to a simpler and faster computation. Smooth CSRBFs have been constructed in [35-37]. Wendland in [37] constructed CSRBFs for particular space dimensions, which possess the lowest possible degree among all piecewise polynomial compactly supported radial functions which are positive definite on R^d and of a given order of smoothness. CSRBFs in [37] have then become a popular tool for surface reconstruction from scattered data (e.g., [2, 29, 38]).

Our method is stimulated by Ohtake's method in [2], where his multi-level reconstruction is insensitive to the density of scattered data, can restore large parts of missing data, and leads to a fast and memory efficient fitting procedure. The method is briefed below.

3.1 Single-level CSRBF interpolation

Considering a set of n points $\tilde{P} = \{\mathrm{p}_1, \mathrm{p}_2, ..., \mathrm{p}_n\}$ with unit normals $\mathrm{N} = \{\mathrm{n}_1, \mathrm{n}_2, ..., \mathrm{n}_n\}$ which scattered along a surface, an implicit surface $f(\mathrm{x}) = 0$ is reconstructed such that it interpolates $\tilde{P}$ and separates the entire space into two parts: $f(\mathrm{x}) > 0$ and $f(\mathrm{x}) < 0$. Ohtake's method interpolates the set of n scattered points $\tilde{P}$ by the following equation

$$f(\mathrm{x}) = \sum_{\mathrm{p}_i \in \tilde{P}} [g_i(\mathrm{x}) + w_i] \phi_\sigma(||\,\mathrm{x} - \mathrm{p}_i\,||) = 0 \tag{2}$$

where $\phi_\sigma(r) = \phi(r/\sigma)$ with

$$\phi(r) = (1-r)_+^4 (4r+1) = \begin{cases} (1-r)^4(4r+1) & (r<1) \\ 0 & (r \geq 1) \end{cases}, \quad r = ||\,\mathrm{x} - \mathrm{p}_i\,||$$

being Wendaland's CSRBF in [37], σ is its support size, w_i s are unknown coefficients to be determined, and $g_i(\mathrm{x})$ are unknown functions which define a local quadratic approximation of $\widetilde{P}$ in a small neighborhood of $\mathrm{p}_i \in \widetilde{P}$. Here, Eq.(2) can be subdivided into two terms: partition of unity (PU) $\sum g_i(\mathrm{x})\phi_\sigma(\|\mathrm{x}-\mathrm{p}_i\|)$ and RBF approximations $\sum w_i\phi_\sigma(\|\mathrm{x}-\mathrm{p}_i\|)$. According to the definition of function $g_i(\mathrm{x})$, for each point $\mathrm{p}_i \in \widetilde{P}$, $g_i(\mathrm{x})$ can be obtained by approximating $\widetilde{P}$ in a vicinity of p_i with a quadric function. Thus, from Eq.(2), the implicit surface can be determined by solving a sparse linear equation system with respect to w_i as

$$\sum_{\mathrm{p}_i \in \widetilde{P}} w_i\phi_\sigma(\|\mathrm{x}-\mathrm{p}_i\|) = -\sum_{\mathrm{p}_i \in \widetilde{P}} g_i(\mathrm{x})\phi_\sigma(\|\mathrm{x}-\mathrm{p}_i\|). \tag{3}$$

3.2 Multi-level CSRBF interpolation

Although single-level CSRBF interpolation procedure is quite fast, there are several limitations. Firstly, it is unable to repair incomplete data and fill holes, and the reconstruction will be slowed down if we enlarge the support size. Secondly, a correct surface extraction requires the grid of polygonization algorithms (e.g., [43, 44]) to be smaller than the width of the support band generated by CSRBFs. To overcome these problems, Ohtake et al. proposed a multilevel interpolation procedure in [2]. Its main idea is to build a coarse-to-fine hierarchy of point sets $\{P^1, P^2, \ldots, P^M = \widetilde{P}\}$ and interpolate a point set P^k in the hierarchy by shifting the interpolation function used in the previous level that interpolates P^{k-1}.

The hierarchy of point sets $\{P^1, P^2, \ldots, P^M = \widetilde{P}\}$ can be constructed by subdividing the points of $\widetilde{P}$ into eight equal octants recursively. For each cell, the centroid of all points in this cell is computed and assigned with the unit average normal determined from these points. The centroids equipped with unit normal vectors of cells in level *k* are then considered as point samples in P^k. After that, Ohtake et al. interpolate (or approximate) a point set of the hierarchy P^k as an offsetting of the interpolating function computed at the previous level P^{k-1}. Therefore, the interpolating function for P^k is defined as

$$f^k(\mathrm{x}) = f^{k-1}(\mathrm{x}) + h^k(\mathrm{x}) = 0 \qquad (k = 1,2,\ldots,M)$$

with a base function $f^0(\mathrm{x}) = -1$. The offsetting function $h^k(\mathrm{x})$ has the same form as Eq.(2) used in the single-level interpolation. The coefficients w_i^k are evaluated by solving a linear system

$$\sum_{\mathrm{p}_i^k \in P^k} w_i^k\phi_\sigma(\|\mathrm{x}-\mathrm{p}_i^k\|) = -f^{k-1}(\mathrm{p}_i^k) - \sum_{\mathrm{p}_i^k \in P^k} g_i^k(\mathrm{x})\phi_\sigma(\|\mathrm{x}-\mathrm{p}_i^k\|), \tag{4}$$

where local approximations $g_i^k(\mathrm{x})$ are determined by least square fitting to P^k.

The approaches based on CSRBFs are sensitive to the density of interpolated/approximated scattered data. In order to adapt the interpolation to scattered data of different densities, it is clearly necessary to be able to scale the support size, σ, of each basis function ϕ. If σ is too small, the approximation will be poor; the surface may exhibit small peaks and flat spots, holes and gaps. On the other hand, if too large values are chosen for σ, the linear equation system for computing w_is is no longer sparse. Therefore, a careful selection of the support size for CSRBFs is necessary. Ohtake et al. presented a method to determine the support size σ^k and the number of subdivision levels M in [2]. The support size σ^k is defined by $\sigma^{k+1} = \sigma^k/2$, $\sigma^1 = \alpha L$, where L is the length of a diagonal of the bounding box of $\widetilde{P}$. The parameter α is chosen such that an octant of the bounding box is always covered by a ball of radius σ^1 centered somewhere in the octant. The number of subdivision levels M is determined by σ^1 and σ^0, where σ^0 is the support size for the single-level interpolation and equals to three fourth of the average diagonal of the leaf cells which contain no more than eight points of $\widetilde{P}$. The equation $M = \lceil -\log_2(\sigma^0/(2\sigma^1)) \rceil$ provided in [2] is used to get the number of levels. In our implementation later, we will also determine the support size and the number of levels using this method in [2].

Although our duplex surface interpolation method uses the multi-level interpolation idea in [2], there are two essential differences: the first is that the fitting function $f(\mathrm{x})$ is formulated by standard RBFs instead of the sum of PU and RBF terms; the second is that our method can fit the zero-level and offset surface at the same time with a single function instead of just for zero-level points. In next section, we will detail our surface reconstruction method and the reason for selecting different techniques.

4. Duplex Surface Interpolation and Approximation

Offsets of solids, which are defined as grown or shrunken versions of the solid, have been defined precisely using the point sets theory in [3]. In [11, 12, 15], offset results are generated individually for surfaces, edges and vertices of a solid model and then combined by a Boolean union. Differently, the solid is represented by points in this paper, so we use an implicit function to fit the zero-level and the offset surface simultaneously after generating offset points.

4.1. Overview

In our fitting algorithm, we assume the given data is a set of oriented points. According to the density of the point set, the support size and the number of levels are determined. In each level, we firstly generate the point set, its offset points and the off-surface points from the original point set $\tilde{P}$, and then construct the fitting function. All steps of our method are illustrated in pseudo-code below.

Algorithm 1. Duplex multi-level fitting	
Input:	An oriented point set P
Output:	A single function f consisted of a set of RBFs
(1)	Determine the support size σ^1 and the number of subdivision levels M according to the given point set P
(2)	For each level k
(3)	Generate the point set P^k of level k
(4)	Generate offset points P_δ^k and off-surface points P_d^k
(5)	Obtain the RBF centers using center reduction (optional)
(6)	Get the offsetting function o^k by solving the system (5), and then get the function f^k
(7)	Set $f = f^k$ and return.

As the reconstruction involving offset points at the non-zero-level set, the single-level method can hardly provide a correct reconstruction. Details will be explained in Section 4.4. If the point set is with noises in normal and position, we will apply a pre-processing for denoising the points. This will be discussed in Section 4.5.

Our method in Algorithm 1 has the following advantages:

1) For a given set of oriented points $\tilde{P}$ and the offset points $\tilde{P}_\delta$ generated from $\tilde{P}$ with an offset distance δ, a single function formed with RBFs is used to describe a zero-level set and its offset surface simultaneously, satisfying $f(\mathrm{x}) = 0\ (\mathrm{x} \in \tilde{P})$ and $f(\mathrm{x}) = \delta\ (\mathrm{x} \in \tilde{P}_\delta)$. The reconstructed implicit function actually gives an elegant mathematical model for the hollowed model of a given object sampled into $\tilde{P}$.

2) A hierarchical computation method has been developed in this paper which makes the algorithm insensitive to the density of scattered data and allows us to reconstruct large parts of missed data.

4.2. Offset points generation

Consider a set of points $\tilde{P} = \{\mathrm{p}_1, \mathrm{p}_2, \ldots, \mathrm{p}_n\}$ with unit normal vectors $\mathrm{N} = \{\mathrm{n}_1, \mathrm{n}_2, \ldots, \mathrm{n}_n\}$ that indicates the

oriented boundary surface of input solid. The offset points can be represented in $\tilde{P}_\delta = \{p_j \mid \min(dis(p_j, p_i)) = \delta, p_i \in \tilde{P}\}$. In practice, the points in $\tilde{P}$ can be generated from range scanners and the normals in N are usually estimated from range data during shape acquisition phase by local least-square fitting.

When generating offset points from a point cloud, we consider all points as face points and produce corresponding offset points in two steps:

- Generating the candidate offset points for all points in $\tilde{P}$ by offsetting a distance δ along the points' normal directions;
- Removing the invalid offset points that locate in the sphere centered at other surface points with radius δ.

After these two steps, the remaining candidate offset points form the offset point set $\tilde{P}_\delta$.

As illustrated in Figure 3, it is not difficult to find that there are holes near corners in the offset point set. This is one of the major reasons why we select the implicit surface to reconstruct the offset surface as it can fill the gaps or holes automatically. In the following two subsections, we will detail the technology of constructing the zero-level surface and the offset surface simultaneously using CSRBFs.

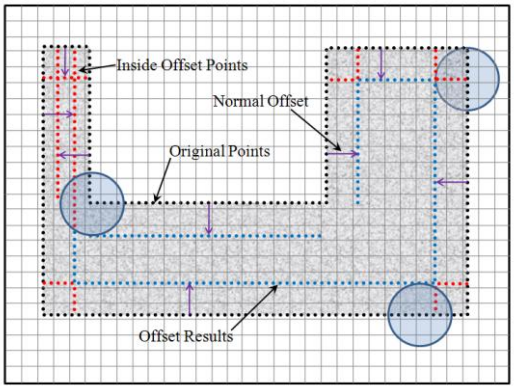


Figure 3. The illustration of offset points generation in 2D: the black points are the original points, the union of red and blue points are the set of candidate offset points shifted from the original points along the normal direction (purple arrows), the red points are removed candidate offset points, and the blue points are the remaining offset points in the offset point set. The grid cells are used to accelerate the processing of point removal.

4.3. Multi-level CSRBFs

Let us consider a set of n points $\tilde{P} = \{p_1, p_2, \ldots, p_n\}$ with unit normals $N = \{n_1, n_2, \ldots, n_n\}$. A set of offset points $\tilde{P}_\delta = \{p_{n+1}, \ldots, p_{n+m}\}$ is obtained via the offset operation in Section 4.2. We wish to find a function $f(x)$ which implicitly defines a 3D scalar field so that it satisfies the equations $f(p_i) = 0 \quad (i = 1, \ldots, n)$ and $f(p_i) = \delta \neq 0 \quad (i = n+1, \ldots, n+m)$. In order to avoid the trivial solution that f is zero everywhere when there are no offset points to be interpolated, extra off-surface points in set P_d are appended to the input data and are given values as their signed Euclidean distance to $\tilde{P}$. Experience has shown that it is better to augment a sample point in $\tilde{P}$ with two off-surface points – one on each side of the surface. However, in our implementation, we consider the offset points as an off-surface point set and just need to generate off-surface points on the opposite side of the offset surface to be constructed. To make the reconstructed surface in a good shape, n off-surface points are expected. More specifically, we move each point in $\tilde{P}$ along the normal vector with distance $+d$ or $-d$ where the sign is determined by the moving direction. In our implementation, we choose $d = \sigma / 10$. Again, the invalid off-points generated in this way will also be removed by the method of offset point selection in Section

4.2.

The interpolation problem can then be restated as follows. Considering a set of n points $\widetilde{P} = \{\mathrm{p}_1, \mathrm{p}_2, ..., \mathrm{p}_n\}$ scattered along a surface, a corresponding set of $n'(n' \le n)$ off-surface points $P_d = \{\mathrm{p}_{n+m+1}, \mathrm{p}_{n+m+2}, ..., \mathrm{p}_{n+n'+m}\}$, and a set of $m(m \le n)$ valid offset points $\widetilde{P}_\delta = \{\mathrm{p}_{n+1}, \mathrm{p}_{n+1}, ..., \mathrm{p}_{n+m}\}$, we will compute a 3D scalar field defined by $f(\mathrm{x})$ such that its zero-level-set $f(\mathrm{x}) = 0$ interpolates $\widetilde{P}$ and the δ-level-set $f(\mathrm{x}) = \delta$ interpolates $\widetilde{P}_\delta$.

We interpolate $\widetilde{P}$ and $\widetilde{P}_\delta$ by

$$f(\mathrm{x}) = \sum_{\mathrm{p}_i \in \widetilde{P}} w_i \phi_\sigma(\| \mathrm{x} - \mathrm{p}_i \|) + \sum_{\mathrm{p}_i \in P_d} w_i \phi_\sigma(\| \mathrm{x} - \mathrm{p}_i \|) + \sum_{\mathrm{p}_i \in \widetilde{P}_\delta} w_i \phi_\sigma(\| \mathrm{x} - \mathrm{p}_i \|), \tag{5}$$

where $\phi_\sigma(r)$, σ and w_i s are defined as in Eq.(2).

Thus, we determine the coefficients w_i from the interpolation conditions that

$$\begin{cases} f(\mathrm{p}_j) = 0 = \sum_{\mathrm{p}_i \in \widetilde{P}} w_i \phi_\sigma(\| \mathrm{p}_j - \mathrm{p}_i \|) + \sum_{\mathrm{p}_i \in P_d} w_i \phi_\sigma(\| \mathrm{p}_j - \mathrm{p}_i \|) + \sum_{\mathrm{p}_i \in \widetilde{P}_\delta} w_i \phi_\sigma(\| \mathrm{p}_j - \mathrm{p}_i \|), & \mathrm{p}_j \in \widetilde{P} \\ f(\mathrm{p}_j) = \delta = \sum_{\mathrm{p}_i \in \widetilde{P}} w_i \phi_\sigma(\| \mathrm{p}_j - \mathrm{p}_i \|) + \sum_{\mathrm{p}_i \in P_d} w_i \phi_\sigma(\| \mathrm{p}_j - \mathrm{p}_i \|) + \sum_{\mathrm{p}_i \in \widetilde{P}_\delta} w_i \phi_\sigma(\| \mathrm{p}_j - \mathrm{p}_i \|), & \mathrm{p}_j \in \widetilde{P}_\delta \\ f(\mathrm{p}_j) = d = \sum_{\mathrm{p}_i \in \widetilde{P}} w_i \phi_\sigma(\| \mathrm{p}_j - \mathrm{p}_i \|) + \sum_{\mathrm{p}_i \in P_d} w_i \phi_\sigma(\| \mathrm{p}_j - \mathrm{p}_i \|) + \sum_{\mathrm{p}_i \in \widetilde{P}_\delta} w_i \phi_\sigma(\| \mathrm{p}_j - \mathrm{p}_i \|), & \mathrm{p}_j \in P_d \end{cases}, \tag{6}$$

therefore Eqs.(6) lead to a sparse system of linear equations with respect to w_i. Since Wendaland's compactly supported RBFs are strictly positive definite (ref. [37]), the $(n+n'+m)\times(n+n'+m)$ interpolation matrix $\Phi = \{\phi_{ij}\}$ is positive definite unless the points in $\widetilde{P}$, $\widetilde{P}_\delta$ and P_d are co-planar, which seldom happens in practice .

Due to the local property of CSRBF, the isosurface of the δ-level-set cannot be correctly reconstructed with the single-level method. Details will be explained in next sub-section. The multilevel interpolation procedure described in the following eliminates this problem.

Borrowing the idea of Ohtake et al. [2], we build a multiscale hierarchy of point sets $\{P^1, P^2, ..., P^M = \widetilde{P}\}$ and interpolate a point set P^{m+1} in the hierarchy by shifting the interpolation function determined at the previous level that interpolates P^m. After that, according to the point set P^k in the k level, we can generate the corresponding offset point sets $P_\delta^k = \{\mathrm{p}_j \mid \min(dis(\mathrm{p}_j, \mathrm{p}_i)) = \delta, \mathrm{p}_i \in P^k\}$ and the off-surface point sets $P_d^k = \{\mathrm{p}_j \mid \min(dis(\mathrm{p}_j, \mathrm{p}_i)) = d, \mathrm{p}_i \in P^k\}$ using the method described in section 4.2. These sets also form two hierarchies: $\{P_\delta^1, P_\delta^2, ..., P_\delta^M = \widetilde{P}_\delta\}$ and $\{P_d^1, P_d^2, ..., P_d^M = P_d\}$. For those points in P^k whose corresponding normal is zero-vector, the off-surface points and the offset points are difficult to compute. Therefore, they are simply neglected. Then, our multilevel interpolation procedure works in the coarse-to-fine way to interpolate a point set P^k in the hierarchy by shifting the interpolation function defined in the previous level P^{k-1}. Figure 4 shows the support spheres of the original and offset points, the reconstructed surfaces using our multilevel interpolation approach, and the cross-sections of the implicit function.

For the k th level of the hierarchy of point sets with offset points and off-surface points, the offsetting function h^k is defined as

$$h^k(\mathrm{x}) = \sum_{\mathrm{p}_i^k \in P^k} w_i^k \phi_{\sigma^k}(\| \mathrm{x} - \mathrm{p}_i^k \|) + \sum_{\mathrm{p}_i^k \in P_\delta^k} w_i^k \phi_{\sigma^k}(\| \mathrm{x} - \mathrm{p}_i^k \|) + \sum_{\mathrm{p}_i^k \in P_d^k} w_i^k \phi_{\sigma^k}(\| \mathrm{x} - \mathrm{p}_i^k \|). \tag{7}$$

Thus, the coefficients w_i^k are found by solving the following system of linear equations

$$\begin{cases} h(\mathrm{p}_j^k) = -f^{k-1}(\mathrm{p}_j^k), & \mathrm{p}_j^k \in P^k \\ h(\mathrm{p}_j^k) = \delta - f^{k-1}(\mathrm{p}_j^k), & \mathrm{p}_j^k \in P_\delta^k \\ h(\mathrm{p}_j^k) = d - f^{k-1}(\mathrm{p}_j^k), & \mathrm{p}_j^k \in P_d^k \end{cases}. \tag{8}$$

Here, the support size σ^k and the number of subdivision levels M are determined by the method in Section 3.2.

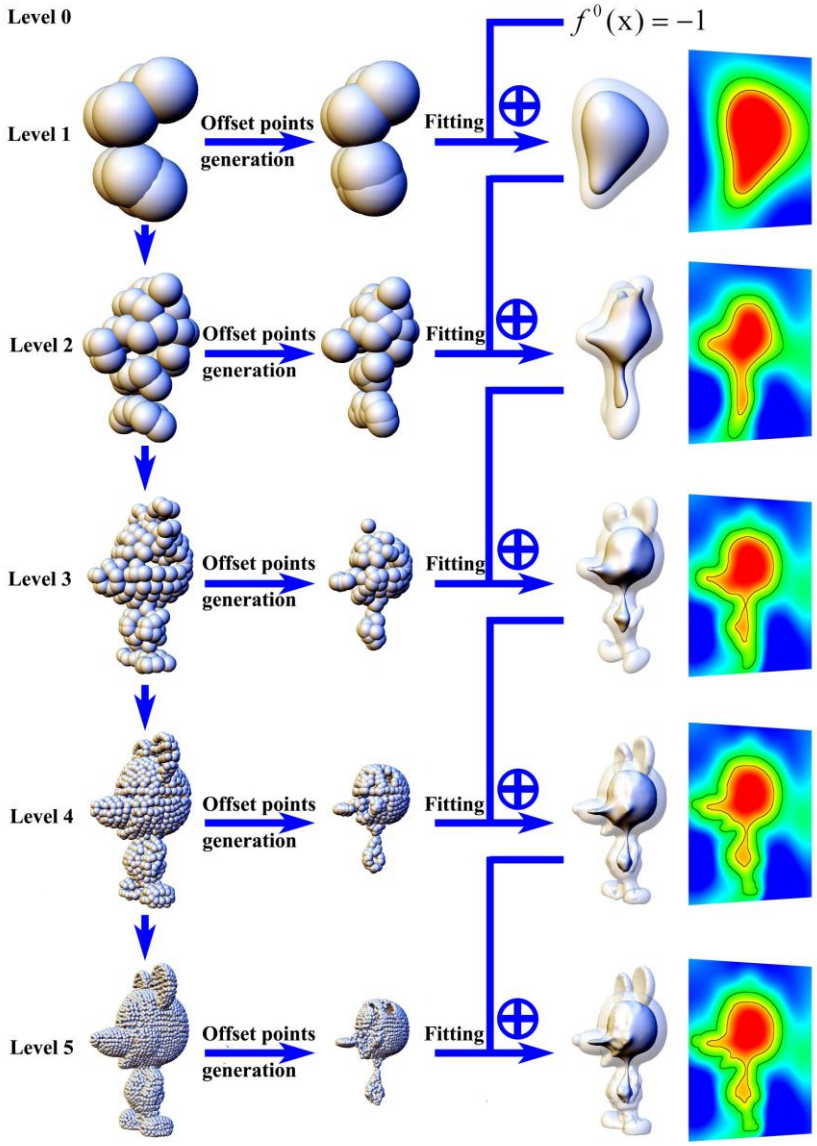
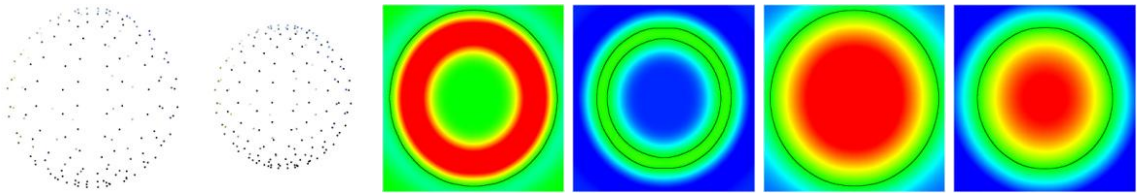


Figure 4 Multilevel interpolation with offset points (δ =0.06). The hierarchies of original points (first column) and offset points (second column) are shown with the radii of the spheres at each level k being 0.2 times of the support size of RBFs used. Reconstructed zero level-set and offset surfaces (third column) are polygonized at each level of the hierarchy. The zero level-set was rendered in transparent mode. Last column shows the cross-sections of implicit functions, where the bold black lines show the zero level sets and the δ -level sets of the functions.

4.4. Single-level vs. multi-level interpolation

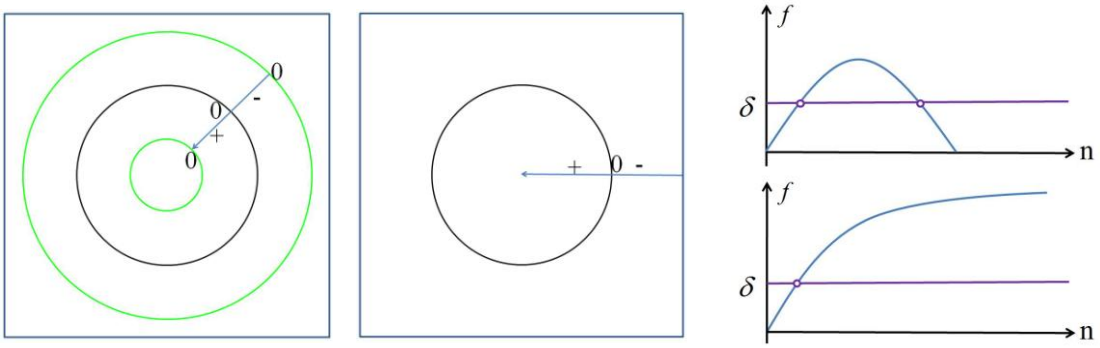
By experiments, we find that even if single-level interpolation is employed here, the reconstructed RBF function still defines a scalar distance field which interpolates the sample points and the offset points. The isosurface of the zero-level is correct. However, for the δ level-set, two parts — one correct part and one extra

part are generated. Figure 5 shows such an example. This is because the constructed implicit function *f* is not monotonic around the isovalue δ since *f* is defined by compactly supported basis functions. Using global RBF can solve this problem – Figure 6 gives an illustration for this. However, using global RBF will slow down the reconstruction process significantly when there is a large number of point data. Besides, there are another two problems as stated in [2]. Firstly, using this single-level interpolation, there is no ability to repair incomplete data – in particular interpolating irregularly sampled data and filling holes. Secondly, the constructed implicit function has a narrow band support, so it requires the grid of later polygonization program (e.g., the Marching Cubes algorithm in [43]) to be smaller than the width of the support band. This will generate too many triangles. The multilevel interpolation procedure solves these problems.



(a) Points (b) Reconstruction by single-level (c) by multi-level CSRBF

Figure 5. Single-level vs. multi-level CSRBF interpolation: original points and the offset points with δ =0.1 (a), and cross-sections of the isosurfaces reconstructed by (b) single-level and (c) multi-level CSRBF interpolation method, where the bold black lines correspond to the zero level-set and the δ level-sets of the functions.



(a) (b) (c) top – CSRBF and bottom – global RBF

Figure 6 The fields of two single-level functions which were formed with (a) CSRBFs and (b) global RBFs respectively. From (c), we can easily find that there are multiple intersections for δ level-set on CSRBF but only one intersection on global RBF.

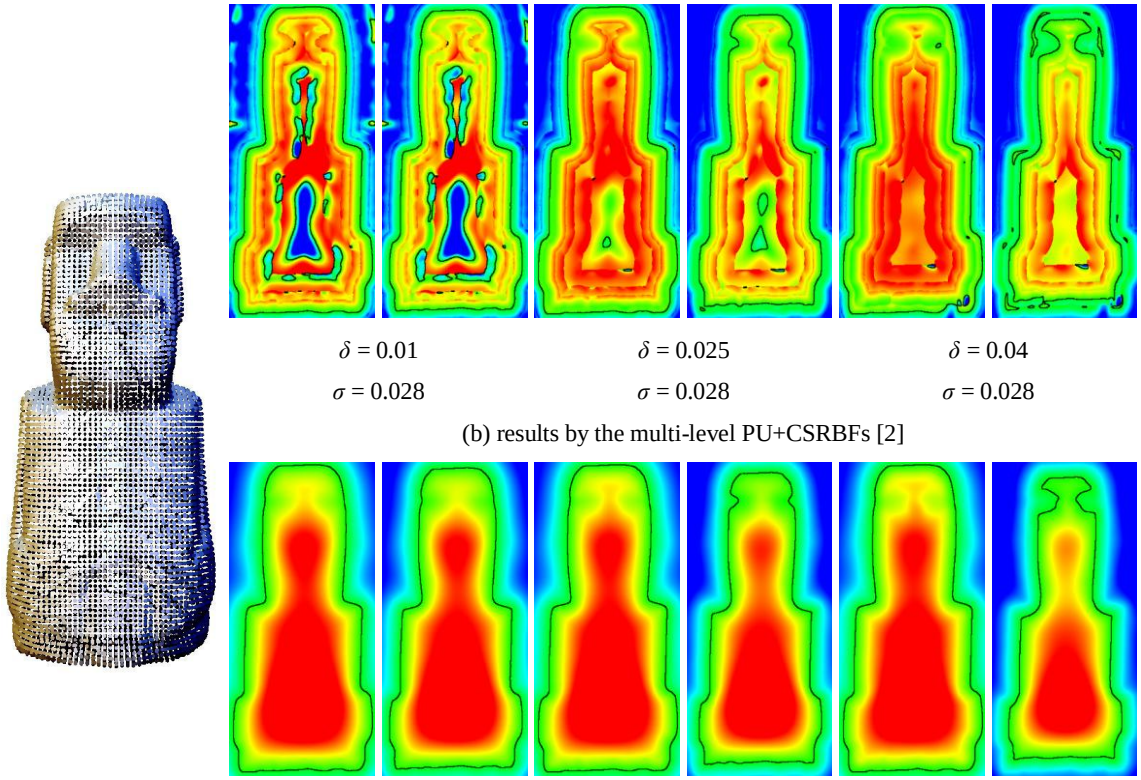
4.5. CSRBF vs. PU+CSRBF

Before using the method presented in this paper, we also tried to use Ohtake et al.'s method (ref. [2]) to reconstruct the surfaces with offset points' constraints. Ohtake et al. interpolate a given set of n scattered points $\tilde{P}$ by the equation (2). It combines partition of unity (PU) and RBF approximations together. As the offset points at the non-zero-level set are involved, the normalized version of radial basis function $\phi_\sigma(\|\mathrm{x}-\mathrm{p}_i\|)/\sum\phi_\sigma(\|\mathrm{x}-\mathrm{p}_i\|)$ are used in Eq.(2). The function can then be rewritten as

$$f(\mathrm{x}) = \sum_{\mathrm{p}_i \in P}[g_i(\mathrm{x}) + w_i]\phi_\sigma(\|\mathrm{x}-\mathrm{p}_i\|) \Big/ \sum_{\mathrm{p}_i \in P}\phi_\sigma(\|\mathrm{x}-\mathrm{p}_i\|). \tag{9}$$

Although this method does not need off-surface points for interior/exterior constraints (i.e., the size of the linear equation system is $(n+m)\times(n+m)$ instead of $(n+n'+m)\times(n+n'+m)$), we found that it can only give correct offset surface when the offset distance is less than the support size of RBF. However, enlarging the

support size will greatly slow down the fitting process. Contrary to this, our method can generate good results in which the offset distance is less constrained by the support size of RBF. With different offset distances, the results generated by multi-level normalized PU and CSRBFs, and multi-level CSRBFs are shown in Figure 7 respectively.



(b) results by the multi-level PU+CSRBFs [2]

(a) (c) results by our duplex fitting

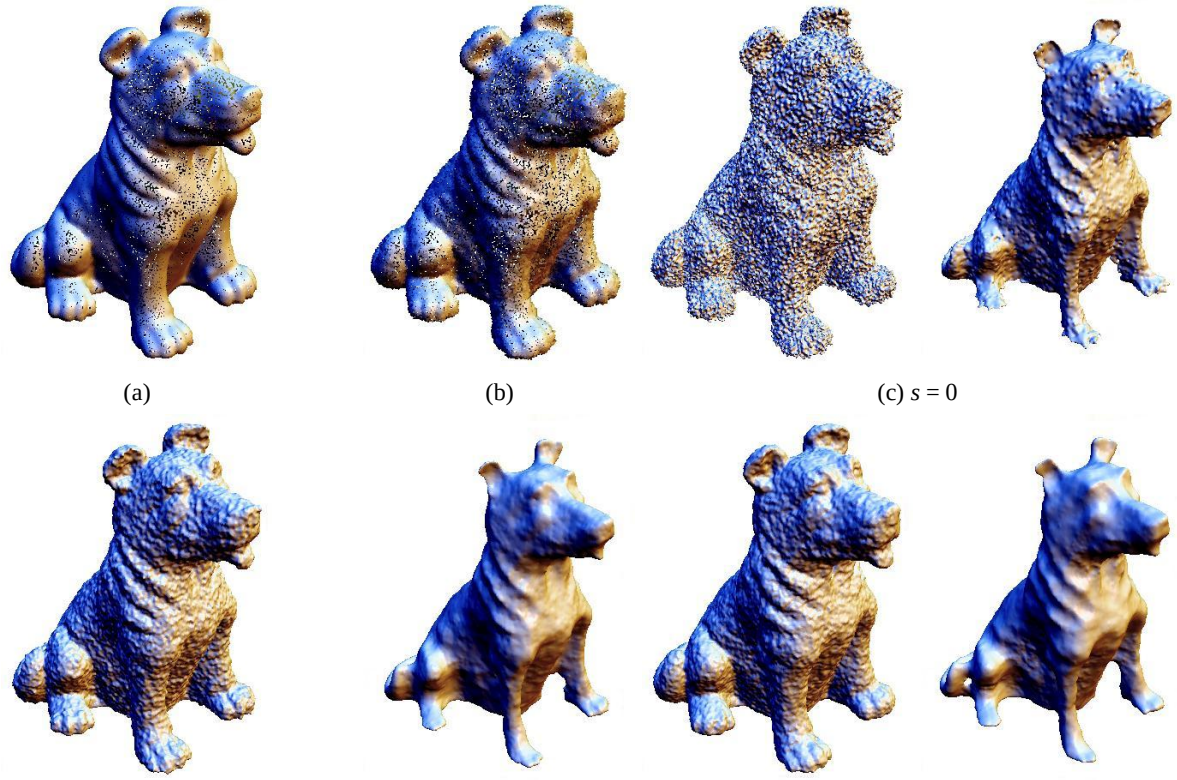
Figure 7. The offset surfaces generated by the multi-level PU+CSRBFs (ref. [2]) versus our duplex multi-level CSRBFs on (a) a given Moai model with 14.8k oriented points. (b) and (c) show three groups of the cross-sections of reconstructed implicit function respectively. Each group shows a pairs of zero-level and offset surface where black contours mean the surfaces with different offset values. δ gives the offset value, and σ is used as the support size of CSRBF.

4.6 Noisy point data

Both positions and normals of the given point samples can have noise embedded. If noises only appear in normal vectors, we can simply use the method in [16] to estimate the correct normals from cloudy points. For the data set with position noise, we tried two methods to process it: 1) approximation (as Ohtake et al. did in [2]) and 2) bilateral filtering.

Approximation: By inverting the interpolation matrix Φ in eq.(6) to a regularized matrix $\Phi + \lambda \mathrm{I}$, the reconstruction procedure will be switched from interpolation to approximation, where λ is a smoothing parameter and I is the identity matrix. In [2], Ohtake et al. used this method to fitting zero-level set from noisy scattered points. Here, we approximate the zero-level set and the offset surface from the given noisy data in the same way. The value of λ is set as $\lambda_k = s \cdot k$ where s is a float number given by users and k is the level of hierarchy. The approximating results with different λ values are shown in Figure 8. It can get satisfactory reconstruction for neither zero-level set nor offset surface. This is because the offset points generated from noisy

cloudy points actually enlarge the level of noise.



(a) (b) (c) s = 0

(d) s = 0.3 (e) s = 0.5

Figure 8. The dog model in (a) is artificially corrupted by Gaussian noise in (b). The surface approximated by our duplex surface reconstruction method with different smooth parameter are shown in (c) – (e) respectively – all are for the offset value $\delta = 0.025$.

Bilateral Filtering: The filter for mesh surface used in [45] is modified to denoise point cloud data here. For a point p in a noisy point set Q, the estimate p' is found by

$$\mathrm{p}' = \frac{1}{\omega(\mathrm{p})} \sum_{\mathrm{q} \in Q} \mathrm{K}_q(\mathrm{p}) s_q \psi(\| \mathrm{q} - \mathrm{p} \|) \zeta(\| \mathrm{K}_q(\mathrm{p}) - \mathrm{p} \|), \tag{10}$$

where $\omega(\cdot)$ is a normalization factor (sum of the weights)

$$\omega(p) = \sum_{\mathrm{q} \in Q} s_q \psi(\| \mathrm{q} - \mathrm{p} \|) \zeta(\| \mathrm{K}_q(\mathrm{p}) - \mathrm{p} \|),$$

$\mathrm{K}_q(\mathrm{p})$ is the projection of p on the tangent plane of q and s_q is the area at point q. Here, we simply compute s_q by $s_q = \sigma^2 / N(q)$ with σ being the support radius of RBF at q and $N(q)$ being the number of points in this support region. $\psi(\cdot)$ and $\zeta(\cdot)$ are the spatial weight and the influence weight Gaussian as

$$\psi(r) = e^{-\frac{r^2}{2\sigma_\psi^2}} \quad \text{and} \quad \zeta(r) = e^{-\frac{r^2}{2\sigma_\zeta^2}}.$$

The widths σ_ψ and σ_ζ control the amount of smoothing.

Simply interpolating the filtered points will not always generate good results (e.g., the one in Figure 9(b)). Therefore, we approximate the filtered points by the aforementioned method with s = 0.2. Better results can be obtained in this way as shown in Figure 9(c). In denoising, we choose $\sigma_\psi = \sigma_\zeta = 4R_{ave}$ where $R_{ave} = 1/N \sqrt{\sum_{\mathrm{q} \in Q} s_q}$.

(a) (b) (c)

Figure 9. The dog model (a) is filtered from the noisy Dog model in Figure 8(b) by our denoising method, (b) the interpolation result on filtered points, and (c) the approximation result on filtered points with s = 0.2. The offset surfaces are reconstructed with $\delta = 0.025$.

5. Results and Application

In this section, we applied the developed method to a variety of point sets. All examples were computed on a PC with Intel Core 2 Quad 2.4GHz CPU and 2.0GB RAM, and our program is written in Visual C++. As described in section 4.2, we generate offset points by a user given offset δ. The initial offset points are generated by shifting along the normal vector with the distance δ from each point. Then, we remove the invalid offset points to get the final offset point set. In our implementation, the offset points can be obtained within one second for all examples except the high resolution Fandisk model in Table 2, which is with more than 550k points and requires 3.1 seconds. After generateing the offset points, our algorithm described in section 4 is used to reconstruct the zero-level and offset surface simultaneously. After computing the implicit surface function, the mesh surfaces for zero-level-set and δ-level-set are extracted by the polygonization algorithm in [44].

5.1. Duplex multi-level CSRBFs fitting results

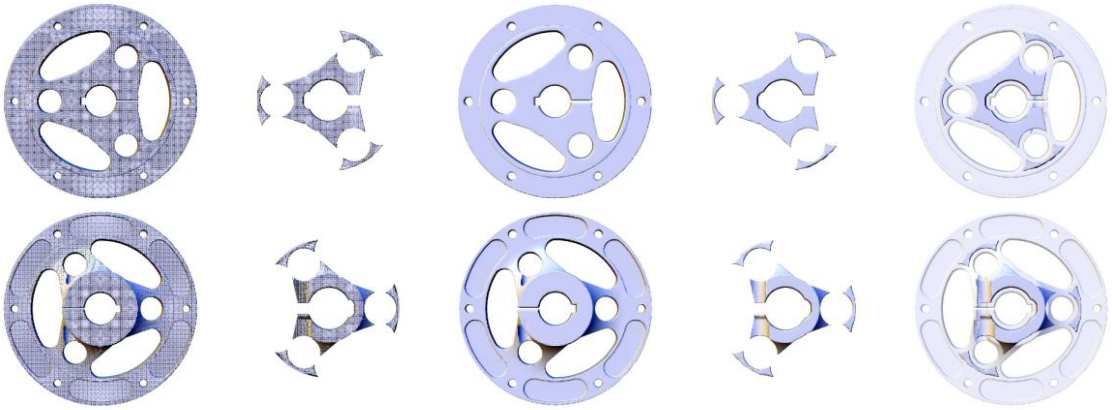
Our method puts the reconstruction of zero-level and offset surfaces in one system. This not only reconstructs a single function to define the model surface and the offset surface with different iso-values, but also improves accuracy of the offset surface for given points. Such improvement is significant especially when the generated offset points are sparse. Figure 10 shows an example for this. Figure 10(a) is the given model Fandisk with 8k points, Figure 10(b) shows the generated offset points, and Figure 10(c) shows the implicit surfaces reconstructed from the original points and the offset points separately. The result using our method is shown in Figure 10(d). The surfaces that interpolate the original points are shown in transparent. It is easy to find that our method can construct more accurate offset surface than the method which fits the offset points along. Similar conclusion can be made on the results of filigree, hub, and eight-shape models in Figure 11-13.

(a) (b) (c) (d)

Figure 10. Separate reconstruction of zero-level and offset surfaces versus our duplex fitting result: (a) the fandisk model with 8,030 points – low resolution, (b) 4,433 offset points are generated with δ = 0.025, and (c) the two surfaces generated separately from (a) and (b), and (d) the surfaces reconstructed simultaneously by our method.

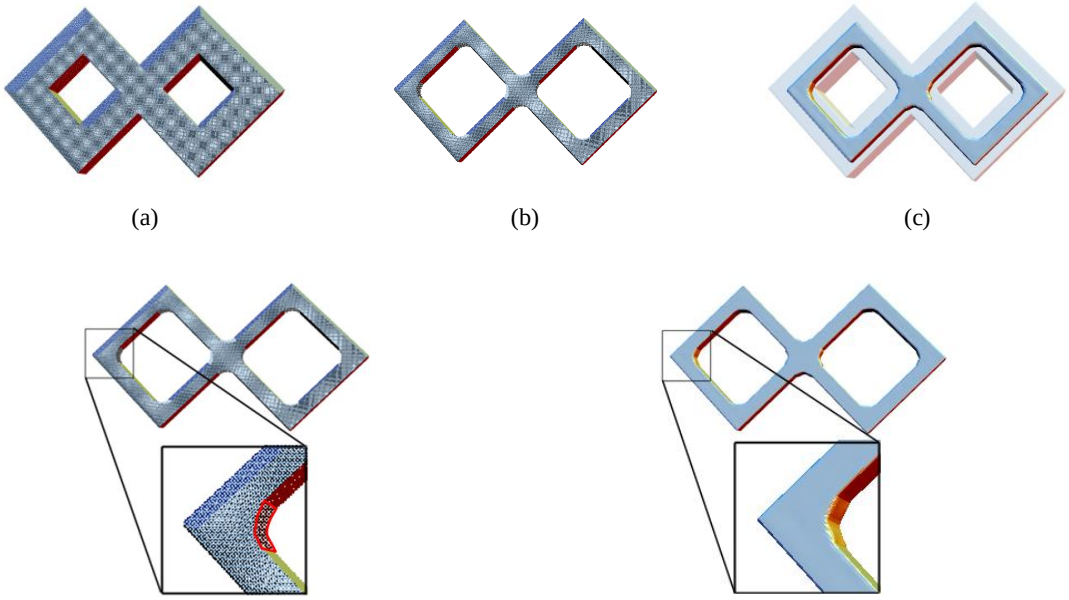
(a) (b) (c) (d) (e)

Figure 11. The duplex fitting result from irregularly sampled data of filigree with δ =0.017: (a) the given point set with 31.4k points, (b) the generated offset points, the reconstructed result – (c) the zero-level and (d) offset surfaces , and (e) superimposition of two surfaces where the zero-level is rendered in transparent.



(a) (b) (c) (d) (e)

Figure 12. The duplex fitting result from irregularly sampled data of the hub model in high resolution with δ =0.025 in front (top row) and back (bottom row) views: (a) the given point set with 118.6k points, (b) the generated offset points, the reconstructed result – (c) the zero-level and (d) offset surfaces, and (e) superimposition of two surfaces where the zero-level is rendered in transparent.



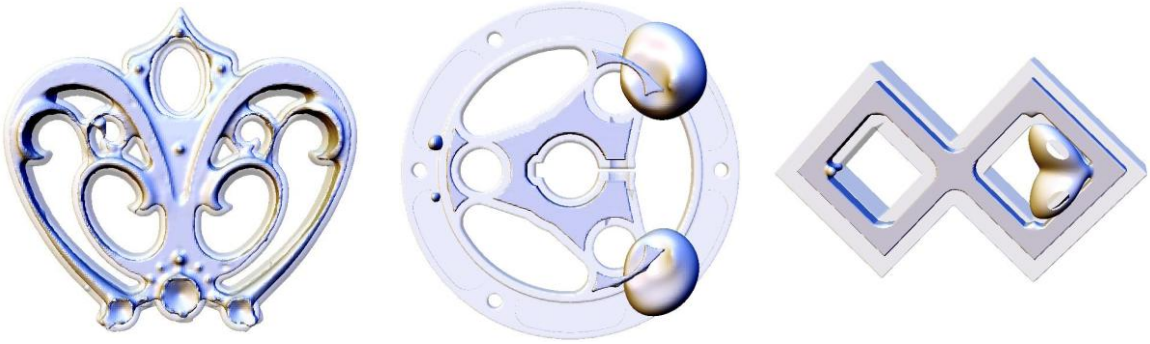
(a) (b) (c) (d) (e)

Figure 13. The reconstruction of zero-level and offset surfaces on eight-shape model with δ = 0.05: (a) the given point set with 110.6k points, (b) the generated offset points, (c) superimposition of two surfaces where the zero-level is rendered in transparent, (d) the hole in offset points which is flagged with red lines, and (e) the reconstructed offset surface with the hole filled.

The example of eight-shape model in Figure 13 demonstrates that our method can fill holes and reconstruct a shape for offset surface as accurately as possible according to the sharp features defined by the given sample points. The original concave sharp features lead to no inside offset points (see Figure 13(b) and (d)). Our method can reconstruct the missing portion of offset surface with the help of the zero-level set defined by the given point set (as shown in Figure 13(c) and (e)). A similar case happens on the gear model example in Figure 14. Figure 15 shows the results of using the method of Ohtake et al. in [2] to reconstruct the zero-level and the offset surfaces separately from given points and the offset points. It is easy to find that the quality of the resultant offset surfaces is very poor. Table 1 gives the quantitative comparison of our method versus the method in [2]. In Table 1, the fitting times for separate reconstruction are computed by the implementation of sequential code on CPU. In our duplex fitting, we speed up the computation by using GPU to solve sparse linear system [46] and the multiple-threads code on CPU for the offset points generation and the function value evaluation.

(a) (b) (c) (d) (e)

Figure 14. The duplex fitting result from irregularly sampled data of gear model with δ =0.025: (a) the given point set with 53.8k points, (b) the generated offset points, the reconstructed result – (c) the zero-level and (d) offset surfaces, and (e) superimposition of two surfaces where the zero-level is rendered in transparent.



(a) (b) (c)

Figure 15. Reconstructing zero-level set and offset surfaces separately using the method in [2]: (a) the filigree model in Figure 11, (b) the hub model in Figure 12, and (c) the eight-shape model in Figure 13 – some unwanted parts are generated on the offset surfaces.

Table 1. Comparisons of fitting times and distances from offset surfaces to original surfaces.

Models	Fig.	Inside offset	Separate reconstruction			Duplex fitting		
			Fitting time	Distance from offset surface to original surface		Fitting time	Distance from offset surface to original surface	
				Maximum	Mean		Maximum	Mean
Sphere	2	0.365	0.02s	0.404	0.371	0.3s	0.371	0.365
Fandisk (lr)	10	0.025	2s	0.042	0.245	3s	0.031	0.025
Filigree	11	0.017	24.6s	0.0265	0.0167	24.5s	0.0305	0.017
Hub (hr)	12	0.025	93.64s	0.13	0.0345	100.5s	0.0313	0.025
Eight-shape	13	0.05	28.15s	0.11	0.046	22.9s	0.052	0.05

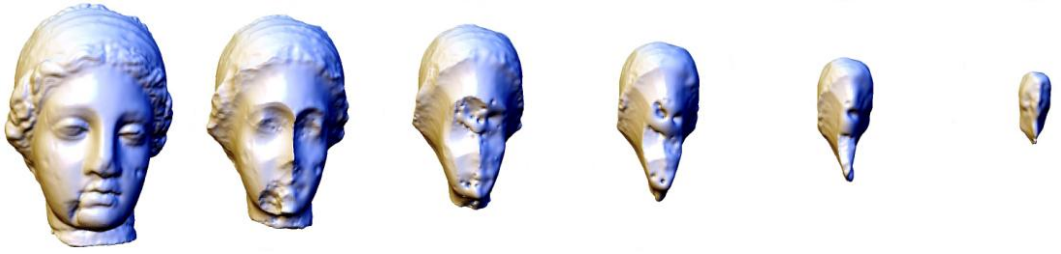
Generally, an RBF fitting will employ all input data points for interpolation and as centers of the RBF. However, such an interpolation is not needed in many applications. Using fewer centers will be able to approximate the input data within a desired accuracy. When duplex fitting the zero-level and offset surfaces using a hierarchy, we test points in the current level by the function in previous level. If the maximal distance from all points to the surfaces defined by the points in last level is less than a user defined *fitting tolerance*, the fitting procedure stops. Such center reduction often results in a shorter fitting time and is optional in our algorithm.

Reconstruction using our duplex multi-level CSRBFs is robust with respect to variations of point density – two examples are shown in Figure 16 and 17. Figure 16 shows a fitting result from a scattered point data set with non-uniform density on vase model. The given points in Figure 17(a) come from the Igea mesh where its right part was first 90% decimated and then with all connectivity information removed.

(a) (b) (c) (d) (e)

Figure 16. The duplex fitting result from a scattered point dataset of vase with non-uniform density, and the offset distance is set as δ = 0.025: (a) the given point set with 51k points, (b) the generated offset points, (c) and (d) are the reconstructed result – the zero-level and offset surfaces respectively, and (e) the superimposition of two surfaces where the zero-level is rendered in transparent.

(a) given points and generated offset points



zero-level surface δ =0.05 δ =0.1 δ =0.15 δ =0.2 δ =0.25

(b) reconstructed zero-level and offset surfaces

Figure 17. Duplex surfaces fitting with different offset distance on Igea model. (a) Original points and offset points with different distances and (b) the reconstructed surface of zero-level set (first column) and offset surfaces with different offset distances (other columns).

As stated before, when the value of offset becomes large, many of the points will be invalid and removed from $\tilde{P}_\delta$. Although our duplex fitting method can help to improve the reconstructed offset surface (e.g., in Figure 2, 10 and 13), for models with complex shape, the error can still be large. In Figure 17, we reconstruct the zero-level set and offset surface with different distances ranging from small distance (0.05) to large distance (0.25) respectively. As the offset distance becomes large, the remaining points become few. In some parts of the model, it results in large distortion on offset surfaces (e.g., the eye part on the Igea model in Figure 17).

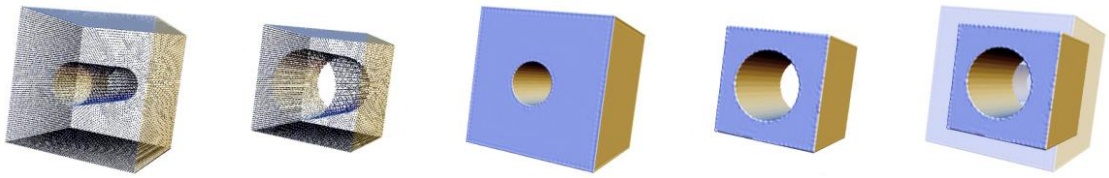
Our duplex fitting algorithm is developed for points – the nature of point representation makes it easy to be modified to a parallel algorithm. Moreover, the local supporting property of CSRBF makes the linear system sparse so that it can be solved with the help of GPU (ref. [46]). To take these two advantages, we speed up the fitting procedure by using GPU-based numerical solver and programming our code in a multi-threads mode. The comparison of computing time between with and without using this speed up is listed in Table 2. It is easy to find that even if working on huge numbers of points, the computation can be completed within two and a half minutes. From Table 2, we can also find that using multi-threads and GPU can increase the efficiency in about 2-6 times without losing the accuracy, which is measured by the maximum function error.

Table 2. Computational Statistics of duplex multi-level RBFs fitting

Models	Fig.	Number of points	Levels	CPU			Multi-threads + GPU		
				Fitting time	Maximum Function Error		Fitting time	Maximum Function Error	
					Zero level-set	Offset surface		Zero level-set	Offset surface
Fandisk(lr)	10	8,030	6	6.4s	2.8×10^{-5}	1×10^{-5}	3s	2.65×10^{-5}	8.6×10^{-5}
Fandisk(hr)	N/A	550,340	9	940.1s	3.2×10^{-4}	4.34×10^{-5}	150.8s	1.6×10^{-4}	1.5×10^{-5}
Hub(lr)	N/A	29,426	6	88.7s	2×10^{-4}	3×10^{-5}	28.5s	1.5×10^{-4}	2.13×10^{-5}
Hub(hr)	12	118,566	7	352.7s	6.14×10^{-5}	1.5×10^{-5}	100.5s	3.7×10^{-5}	1.25×10^{-5}
Moai	7	40,002	7	37.6s	4.3×10^{-7}	2.5×10^{-7}	7.12s	5.7×10^{-6}	4×10^{-6}
Filigree	11	31,400	7	44.4s	2×10^{-9}	0	8.57s	6.4×10^{-6}	5.3×10^{-6}
Gear	14	53,816	7	91.68s	4.8×10^{-6}	2.15×10^{-6}	24.5s	6.1×10^{-6}	1.14×10^{-5}
Eight-shape	13	110,628	7	81.54s	6×10^{-5}	1.5×10^{-5}	20.26s	8.7×10^{-5}	1.7×10^{-5}
Dog	1	49,998	7	119.3s	3.7×10^{-9}	0	22.9s	7.1×10^{-6}	2.1×10^{-6}
Igea	17	72,545	7	78.86s	1.6×10^{-7}	2.4×10^{-8}	13.12s	6.7×10^{-6}	4.8×10^{-6}
Vase	16	51,081	8	63.57s	8×10^{-10}	0	13.88s	2.84×10^{-6}	6×10^{-6}

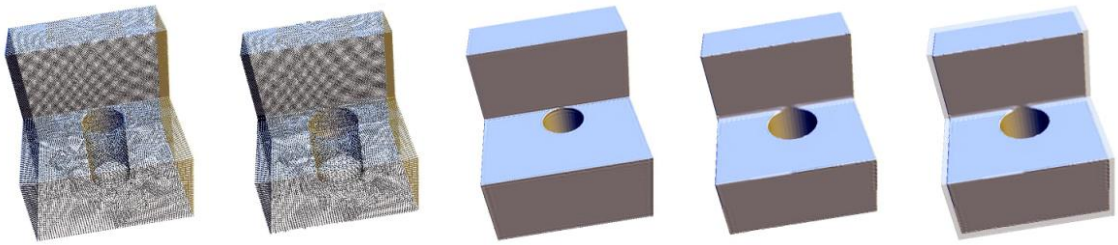
*Note that: 1) all models are scaled into a box with unit width, and 2) the maximum function error from points is generated by the iterative numerical solvers – preconditioned biconjugate gradient (PBCG) is employed in CPU solver and the Jacobi-preconditionned conjugate gradient is used in GPU solver.

To verify the accuracy of resultant offset surfaces by our method, we used two models created in SolidWorks (ref. [48]). For PART1 model in Figure 18 and PART2 in Figure 19, we generate offset surfaces with distances 0.1 and 0.025 inside it. The point clouds are generated by remeshing the original model and removing the connectivity information. The publicly available error measurement tool in [42] is employed to compare the offset results generated by our algorithm and the ideal offset results constructed in SolidWorks. The comparisons are provided in Table 3.



(a) (b) (c) (d) (e)

Figure 18. Offset points and surfaces for model PART1 (δ =0.1): (a) the input points, (b) inside offset points, (c) the reconstructed zero-level surface, (d) the reconstructed offset surface, and (e) the superimposition of two isosurfaces.



(a) (b) (c) (d) (e)

Figure 19. The reconstruction of zero-level and offset surfaces with δ =0.025 for model PART2: (a) the given point set, (b) the generated offset points, (c) and (d) are the reconstructed results of the zero-level and offset surfaces respectively, and (e) the superimposition of two surfaces where the zero-level is rendered in transparency.

Table 3. Measurement of shape error

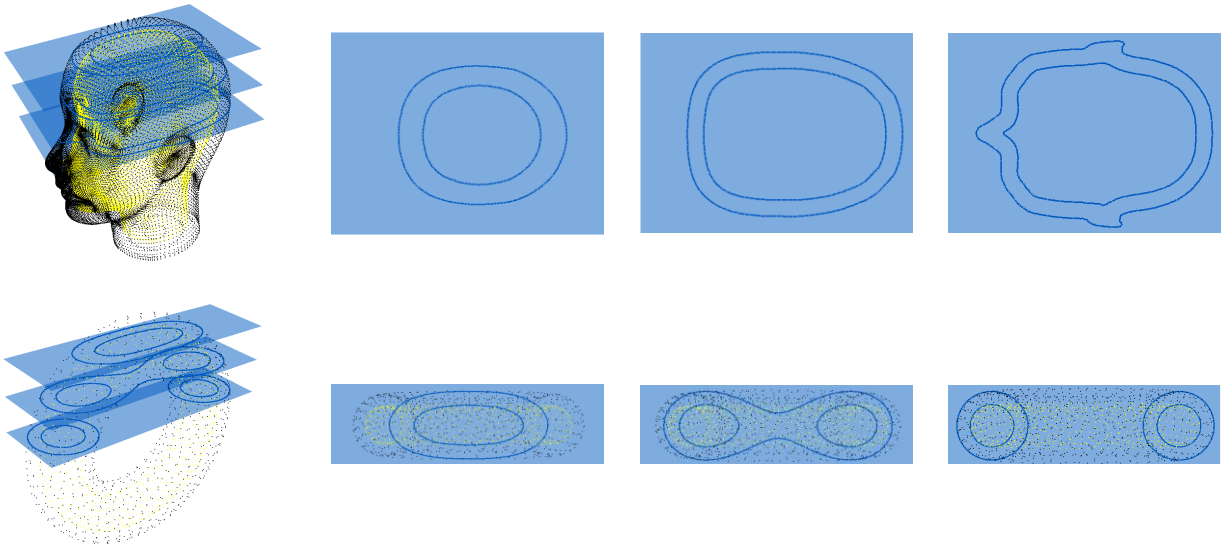
Models	Inside offset	Shape error			
		Forward distance		Backward distance	
		Maximum	Mean	Maximum	Mean
PART1	0.1	0.005	0.0002	0.009	0.00022
PART2	0.025	0.00795	0.000105	0.01115	0.000135

5.2. Application in adaptive slicing

One example application of the implicit function generated by our method is adaptive slicing in rapid prototyping. During the slicing process, we need to generate 2D contours and calculate the layer thickness, where the layer thickness is determined adaptively by the curvature of a planar curve on the reconstructed RBF surfaces.

From Section 4, we have already obtained a CSRBF implicit function $f(\mathrm{x})$ with $\mathrm{x}=(x,y,z)^T$, whose isosurfaces at zero-level and the δ-level give the boundary surfaces of hollowed models. In this application, only the curvature of a planar curve is used in determining layer thickness, which is given in Appendix A.

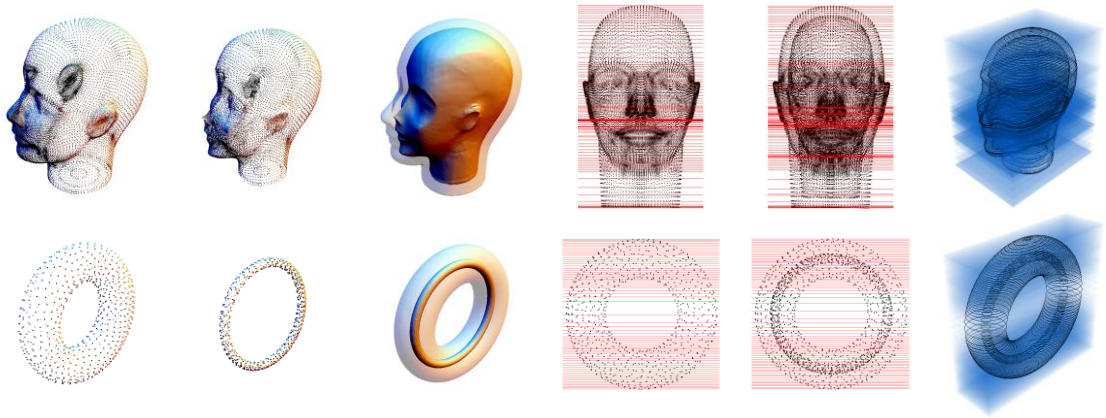
2D contours are generated by intersecting the horizontal slicing plane with the isosurface of the implicit function determined above. This is very important to the slicing procedure. In [41], the authors proposed an adaptive 2D contour generation method by a marching process, which has trouble to output loops with correct topology at the layer with critical point (i.e., the point where two loops join into one). When slicing the hollowed models, we need to generate 2D contours for both the zero level-set and the δ level-set. Here, we adopt the marching square method in [47] to generate 2D contours which can find all contours easily (e.g., Figure 20).



(a) (b) $z = 0.4$ (c) $z = 0.25$ (d) $z = 0.1$

Figure 20. Slicing of a head model (the top row) and a torus (the bottom row) with three horizontal planes (blue) on the resultant implicit function of our duplex fitting. Black points are input oriented sample points and the yellow points are offset points. Locations of the three slicing planes are shown in (a), and the 2D contours are given in (b)-(d).

We adopt the strategy presented in [40, 41] to compute adaptive thickness of layers which are determined by the cusp height and the vertical normal curvature at all points on each layer. Two slicing results are illustrated in Figure 21. In Figure 21(d) and (e), the red lines represent the slicing planes, where the slicing results in Figure 21(d) are obtained by considering the zero-level surface only whereas the results in Figure 21(d) are obtained by considering both the zero-level and the offset surfaces. Compared with Figure 21(d), it is easy to find that more layers are generated in Figure 21(e).



(a) (b) (c) (d) (e) (f)

Figure 21 Adaptive slicing results (Top row – a head model: δ =0.05 with cusp height τ =0.01, grid size 0.005; Bottom row – a torus model: δ =0.1 with cusp height τ =0.01 and grid size 0.01): (a) input point sets, (b) offset points, (c) reconstructed surfaces, (d) slicing results obtained by only considering the zero-level surface (head model: 88 layers and torus model: 67 layers), (e) slicing results obtained by considering both the zero-level and the offset surfaces (head model: 114 layers and torus model: 75 layers), and (f) the isometric views of slices in (e).

6. Conclusion

In this paper, we proposed an implicit surface reconstruction method to interpolate (or approximate) the

original points and offset points simultaneously using a single function. Given a set of oriented points $\tilde{P}$, we first generate the offset points $\tilde{P}_\delta$ by selecting the valid points shifted along the normal direction with a distance δ. Then, we use CSRBFs to form a single function and make it satisfy $f(\mathrm{x}) = 0\ (\mathrm{x} \in \tilde{P})$ and $f(\mathrm{x}) = \delta\ (\mathrm{x} \in \tilde{P}_\delta)$. The reconstructed implicit function actually gives an elegant mathematical model for the hollowed model of a given object sampled into $\tilde{P}$. A hierarchical computation method has been developed in this paper which makes the algorithm insensitive to the density of scattered data and allows us to reconstruct large parts of missing data. Lastly, one application of the reconstructed implicit function in adaptive slicing is given to demonstrate its functionality in rapid prototyping.

In our work, the offset points were generated by shifting from the original point set along the normal direction. This method makes the quality of final reconstructed offset surface dependent on the original points tightly. Furthermore, from the discrete normals coupled with the given sample points, the generated offset points will be sparser or there will be larger holes or gaps when the value of offsetting is larger. This will also downgrade the accuracy of reconstructed offset surface. How to generate a set of offset points with good quality from a point cloud will be one of our future works. Another work which is under planning is to verify the adaptive sliced objects with the computational results – however, errors could be generated from both the manufacturing and the measurement steps too. Furthermore, the theoretical analysis about why PU+CSRBF cannot generate valid results may also become a very interesting research topic in the future.

References

[1] Ohtake Y, Belyaev A, Alexa M, Turk G, and Seidel H. Multi-level partition of unity implicits. *ACM Transaction On Graphics*, 22(3), 2003: 463-470.

[2] Ohtake Y, Belyaev A, and Seidel H. 3D scattered data interpolation and approximation with multilevel compactly supported RBFs. Graphical Models, 67, 2005: 150-165.

[3] Rossignac J R and Requicha A A G. Offsetting operations in solid modelling. Computer Aided Geometry Design, 3(2), 1986: 129–148.

[4] Varadhan G and Manocha D. Accurate minkowski sum approximation of polyhedral models. In Proceedings of Pacific Graphics Conference 2004, pp. 392–401.

[5] Forsyth M. Shelling and offsetting bodies. In Proceedings of the ACM Symposium on Solid Modeling and Applications, Salt Lake City, Utah, United States, May 1995, pp. 373-381.

[6] Qu X and Stucker B. A 3d surface offset method for stl-format models. Rapid Prototyping Journal, 9(3), 2003: 133–141.

[7] Lien JM and Amato NM. Approximate Convex Decomposition of Polyhedra. In Proceedings of the ACM Symposium on Solid and Physical Modeling (SPM), Beijing, China, June 2007, pp. 121-131.

[8] Cohen J, Varshney A, Manocha D, Turk G, Weber H, Agarwal P, Brooks F, and Wright W. Simplification envelopes. In Proceedings of ACM SIGGRAPH (1996), pp. 119–128.

[9] Breen D E and Mauch S. Generating shaded offset surfaces with distance, closest-point and color voumes. In Proceedings of the International Workshop on Volume Graphics, 1999, pp. 307–320.

[10] Breen D E, Mauch S, and Whitaker R T. 3D scan conversion of CSG models into distance volumes. In Proceedings of the 1998 IEEE symposium on Volume visualization, New York, USA, 1998, ACM Press, pp. 7–14.

[11] Chen Y, Wang H, Rosen D W, and Rossignac J. Filleting and rounding using a point-based method. In DETC'05 Proceedings (2005).

[12] Chen Y, Wang H, Rosen D W, and Rossignac J. A Point-Based Offsetting Method of Polygonal Meshes. Technical report, 2005.

[13] Huang J and Crawfis R. Adaptively represented complete distance fields. Geometric Modeling for Scientific Visualization

(2002).
[14] Huang J, Li Y, Crawfis R, Lu S C, and Liou S Y. A complete distance field representation. In Proceedings of the conference on Visualization ’01 (2001), pp. 247–254.
[15] Pavic D. and Kobbelt L. High-resolution volumetric computation of offset surfaces with feature presentation. In Proceedings of Eurographics 2008.
[16] 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.
[17] Curless B and Levoy M. A volumetric method for building complex models from range images. In *Proceedings of SIGGRAPH 1996*, 1996, pp. 303-312.
[18] Whitaker R. A level-set approach to 3d reconstruction from range data. In Proceedings of *IJCV* 1998, pp. 203-231.
[19] Carr J, Beatson R, Cherrie H, Mitchel T, Fright W, Mccallum B, and Evans T. Reconstruction and representation of 3D objects with radial basis functions. In Proceedings of *ACM SIGGRAPH* 2001, pp. 67-76.
[20] Davis J, Marschner SR, Garr M, and Levoy M. Filling holes in complex surfaces using volumetric diffusion. in First International Symposium on 3D Data Processing, Visualization, and Transmission, Padua, Italy, June 2002, pp. 428–438.
[21] Ohtake Y, Belyaev A, and Seidel H. 3D scattered data approximation with adaptive compactly supported radial basis functions. In Proceedings of the Shape Modeling International 2004, pp. 31-39.
[22] Turk G and O'brien J. Modelling with implicit surfaces that interpolate. *ACM Transaction on Graphics*, 21(4), 2004: 855-873.
[23] Sethian JA. Level Set Methods and Fast Marching Methods. Second ed., Cambridge University Press, Cambridge, 1999.
[24] Turk G and Brien J O. Shape transformation using variational implicit surfaces. In Proceedings of ACM SIGGRAPH 1999, August 1999, pp. 335–342.
[25] Frisken SF, Perry RN, Rockwood A, and Jones TR. Adaptively sampled distance fields: a general representation of shape for computer graphics. In Proceedings of ACM SIGGRAPH 2000, July 2000, pp. 249–254.
[26] Zhao H and Osher S. Visualization, analysis and shape reconstruction of unorganized data sets. In: S. Osher, N. Paragios (Eds.), Geometric Level Set Methods in Imaging, Vision and Graphics, Springer, Berlin, 2002.
[27] Dinh HQ, Slabaugh G, and Turk G. Reconstructing surfaces using anisotropic basis functions. in: International Conference on Computer Vision (ICCV) 2001, Vancouver, Canada, July 2001, pp. 606–613.
[28] Dinh HQ, Turk G, and Slabaugh G. Reconstructing surfaces by volumetric regularization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24 (10), 2002: 1358–1371.
[29] Morse BS, Yoo TS, Rheingans P, Chen DT, and Subramanian KR. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In Shape Modeling International 2001, Genova, Italy, May 2001, pp. 89–98.
[30] Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, and Silva CT. Point set surfaces. In Proceedings of IEEE Visualization 2001, October 2001, pp. 21–28.
[31] Amenta N and Kil Y. Defining point-set surfaces. ACM Transactions on Graphics, 2004, 23(3): 264-270.
[32] Levin D. Mesh-independent surface interpolation. In Geometric Modeling for Scientific Visualization, Springer, Berlin, 2003.
[33] Bloomenthal J (Ed.). Introduction to Implicit Surfaces, Morgan Kaufmann, Los Altos, 1997.
[34] Dyn N, Levin D, and Rappa S. Numerical procedures for global surface fitting of scattered data by radial functions. SIAM Journal on Scientific and Statistical Computing, 7, 1986: 639-659.
[35] Schaback R. Creating surfaces from scattered data using radial basis functions. In Mathematical Methods for Curve and Surfaces, M. Daehlen, T.Lyche, and L. Schumaker (eds), Vanderbilt University Press, Nashville, 1995, 477-496.
[36] Wu Z.. Multivariate compactly supported positive definite radial functions. Advances in Computational Mathematics, 49, 1995: 283-292.
[37] Wendland H. Piecewise polynomial, positive definite and compactly supported radial basis functions of minimal degree. Advances in Computational Mathematics, 4, 1995: 389–396.
[38] Kojekine N, Hagiwara I, and Savchenko V. Software tools using CSRBFs for processing scattered data. Computers and Graphics, 27 (2), 2003: 309–317.

[39] Goldman R. Curvature formulas for implicit curves and surfaces. Computer Aided Geometric Design, 22, 2005, 632-658.

[40] Kulkarni P and Dutta D. An accurate slicing procedure for layered manufacturing, Computer Aided Design, 28(9), 1996: 683-697.

[41] Yang P. and Qian X. Adaptive slicing of moving least squares surfaces: toward direct manufacturing of point set surfaces. ASME Transactions Journal of Computing and Information Science in Engineering. Accepted.

[42] Cignoni P, Rocchini C and Scopigno R. Metro: measuring error on simplified surfaces. Computer Graphics Forum, 17(2), 1998: 167-174.

[43] Lorensen WE and Cline HE. Marching cubes: a high resolution 3D surface construction algorithm. Computer Graphics, 21(3), 1987: 163-169.

[44] Bloomenthal J. An implicit surface polygonizer. Graphics Gems IV 1 (1994), 324-349.

[45] Jones TR, Durand F, and Desbrun M. Non-iterative, feature-preserving mesh smoothing. ACM Transaction on Graphics, 21(3), 2003: 943-949.

[46] Buatois L, Caumon G, and Levy B. Concurrent number cruncher - A GPU implementation of a general sparse linear solver. International Journal of Parallel, Emergent and Distributed Systems, accepted, 2008.

[47] Marching squares. http://www.polytech.unice.fr/~lingrand/MarchingCubes/algo.html#squar.

[48] SolidWorks. http://www.solidworks.com

Appendix A Curvature on Planar Curves

Any plane curve on these isosurfaces can be defined as the intersection of the isosurface and a plane, for example

$$\{f(\mathrm{x})=0\}\cap\{g(\mathrm{x})=ax+by+cz+d=0\}.$$

For a more elegant expression of this planar curve, we can transform coordinate system so that the plane $g(\mathrm{x})=0$ is transformed to the *xy*-plane $\bar{g}(\mathrm{x})=z=0$. Then the expression for the implicit curve will be reduced to $\bar{f}(x,y)=\sum_i w_i\phi_\sigma(x,y)$ which is a function of variables *x* and *y* only.

Applying a curvature formula given in [39], we have the curvature of this implicit planar curve as

$$k=-\frac{(\mathrm{T}\cdot\nabla\bar{f})\cdot H(\bar{f})\cdot(\mathrm{T}\cdot\nabla\bar{f})^T}{|\nabla\bar{f}|^3} \tag{A1}$$

where $\mathrm{T}=\begin{bmatrix}0 & -1\\ 1 & 0\end{bmatrix}$, $\nabla\bar{f}=\left(\frac{\partial\bar{f}}{\partial x}\quad\frac{\partial\bar{f}}{\partial y}\right)^T$ is the gradient of $\bar{f}$, and $H(\bar{f})=\nabla(\nabla(\bar{f}))$ is the Hessian matrix of $\bar{f}$.

With Eq.(5), we can derive the formulas for $\nabla\bar{f}$ and $H(\bar{f})$. Here we just give the resulting formulas. The gradient of $\nabla\bar{f}$ can be expressed as

$$\nabla\bar{f}=\sum_{\mathrm{p}_i} w_i(-20(1-\|\mathrm{x}-\mathrm{p}_i\|/\sigma)_+^3/\sigma^2)(\mathrm{x}-\mathrm{p}_i)^T \tag{A2}$$

The Hessian matrix of $\bar{f}$ can be expressed as

$$\begin{aligned}H(\bar{f})&=\nabla(\nabla\bar{f})\\&=\sum_{\mathrm{p}_i} w_i(20(1-\|\mathrm{x}-\mathrm{p}_i\|/\sigma)_+^2/\sigma^2)(3(\mathrm{x}-\mathrm{p}_i)^T(\mathrm{x}-\mathrm{p}_i)/(\sigma\|\mathrm{x}-\mathrm{p}_i\|)-(1-\|\mathrm{x}-\mathrm{p}_i\|/\sigma)\mathrm{I}),\end{aligned} \tag{A3}$$

where I is the identity matrix. Notice that if the sign of the resulted curvature k is negative, the curvature vector k is opposite to the direction of normal n, or the planar curve is convex at this point; otherwise, k and n have the same direction, or the planar curve is concave. It is important to distinguish between the convex and concave properties of the planar curve since different formulas of the layer thickness will be applied to these two different cases (ref. [40, 41]).