

Quasi-Interpolation for Surface Reconstruction from Scattered Data with Radial Basis Function

Shengjun Liu¹, Charlie C.L. Wang²

¹ School of Mathematical Science and Computing Technology, Central South University, China

² Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong, China

Abstract

Radial Basis Function (RBF) has been used in surface reconstruction methods to interpolate or approximate scattered data points, which involves solving a large linear system. The linear systems for determining coefficients of RBF may be ill-conditioned when processing a large point set, which leads to unstable numerical results. We introduce a quasi-interpolation framework based on compactly supported RBF to solve this problem. In this framework, implicit surfaces can be reconstructed without solving a large linear system. With the help of an adaptive space partitioning technique, our approach is robust and can successfully reconstruct surfaces on non-uniform and noisy point sets. Moreover, as the computation of quasi-interpolation is localized, it can be easily parallelized on multi-core CPUs.

1 Introduction

Surface reconstruction from points sampled from a three-dimensional object has played an important role in geometric modeling for a few decades. Its aim is to construct a geometric structure on a set of points in order to make explicit the proximity relationships between points on the surface of an object [8]. A lot of research work has been devoted to developing reconstruction methods for applications in computer graphics, robotics, and computer-aided design and manufacturing (CAD/CAM). Implicit representation is good at repairing incomplete data as it can handle topology change easily and is robust for noises. Therefore, approaches based on fitting implicit functions to input point sets become one of the main trends in surface reconstruction, where several approaches in literature used RBF based interpolation or approximation [11, 28]. However, when the number of interpolation points is very large, the interpolation matrix may become ill-conditioned. In this case, more than having a high computational cost, the numerical computation could become unstable. Changing interpolation into approximation by adding an approximation efficient on the diagonal of the matrix can improve the stability; however, the stability still pretty much relies on the performance of numerical solvers. Different from an exact interpolation approach, the quasi-interpolation method presented in this paper does not require solving large linear systems, and can generate satisfactory results with small shape approximation errors.

Most work about quasi-interpolation in literature [18, 36] is developed for the computation on regular grid points; here, we generalize it to work for unorganized point sets and the surface reconstruction problem can be solved thereafter. Specifically, a hierarchical quasi-interpolation method based on RBFs is proposed in this paper for reconstructing an implicit surface from scattered points.

The technical contribution of this approach is threefold. Firstly, we introduce a quasi-interpolation method for fitting *Radial Basis Functions* (RBFs) onto scattered points. This method avoids solving large linear systems

which may be ill-conditioned. In the quasi-interpolation method, the RBF on each point is computed locally and independently; therefore, the reconstruction can be easily parallelized. Secondly, a multi-level quasi-interpolation method based on *Compactly Supported Radial Basis Functions* (CSRBFs) is presented. How shape parameters influence the quality of reconstructed surface is also analyzed. Lastly, an adaptive scheme for choosing shape parameters and compact supports in CSRBF based fitting is proposed for reconstructing surfaces from imperfect data. As a result, for those almost uniform points, no parameter needs to be specified. For highly non-uniform points, only a parameter δ needs to be given by users for determining adaptive support size.

The rest of this paper is organized as follows. After reviewing some related work in Section 2, we present the details of RBF-based quasi-interpolation in Section 3. Section 4 discusses the function of shape parameters and support sizes on reconstruction results, and introduces an adaptive scheme for determining their values. Experimental results are given in Section 5, and our paper ends with the conclusion section.

2 Related Work

The problem of reconstructing surfaces from point clouds has been widely studied for many years (ref. [9, 30]). A large number of reconstruction algorithms have been proposed. Basically, they can be classified into two major groups: direct methods and indirect methods.

Direct methods usually involve the construction of Voronoi diagram and reconstruct a mesh surface by linking points directly, which need dense samples with little noise [3, 14, 15]. Although a number of algorithms that are robust to noises have been presented recently [4, 16, 22], this type of methods is generally more sensitive to noise than indirect methods. Moreover, the cost to compute Voronoi diagram is high in both memory and time.

Indirect methods attempt to create a signed implicit function, which divides the space into inside and outside of an object, from a set of oriented points. Mesh surface can then be extracted from the zero level set of this implicit function by contouring methods (e.g., [7, 20, 25]). Surface reconstruction algorithms in this group can be further classified according to whether the computation is taken locally or globally.

For local methods, the main advantages include fast computation and ability to handle large data sets. Algorithms in this sub-group began from Hoppe [19]. He approximated a manifold with piecewise linear surfaces from unorganized points. After that, a volumetric method was presented which provides high-resolution surfaces from range images but struggles to handle misaligned surfaces [12]. Other examples of local approaches include the one based on moving least squares [1, 17] and its variants [5, 29]. The *Multi-level Partition of Unity Implicits*

(MPU) method [27] proposed by Ohtake et al. excels in the above advantages. However, these methods have difficulty in handling points with poor quality, such as those contain noises, outliers, holes, and have high non-uniformity.

Global function based approaches have the advantage of being able to handle data sets with the aforementioned defects. Examples of such methods include RBF-based approaches [11, 28, 34], integration of Voronoi diagrams and variational method [2], Poisson surface reconstruction technique [21], and smooth signed distance method [10]. These methods reduce the surface reconstruction problem to a numerical optimization problem, and often need to find the solution of a large linear system. Unfortunately, if there is a large number of points, the linear system may become ill-conditioned and its numerical computation can be unstable.

Our surface reconstruction method presented in this paper is global function based. A quasi-interpolation scheme is introduced to speed up the reconstruction of surfaces from scattered points. It is stable and does not need to solve the large linear system that could be ill-conditioned. The adaptive multi-level quasi-interpolation framework is robust to points with non-uniformities and noises, and can successfully reconstruct surfaces in high quality.

Quasi-interpolation is a kind of approximation methods. It possesses some advantages, such as less computation time and stable computation. In [13, 23, 26], some quasi-interpolation methods have been discussed for approximation of a function with a neural network the activation function of which is sigmoidal. The univariate quasi-interpolants were discussed in [36]. Han and Hou discussed quasi-interpolation by radial basis functions and provided a suitable value of the shape parameter in [18], which shows a constructive method for obtaining a family of approximate interpolations. The surface reconstruction method in this paper is inspired by it. Here, the quasi-interpolation is applied to scattered points instead of regularly sampled grid points.

3 RBF-based Quasi-interpolation

Given a set of $(n + 1)$ data pairs $(\mathbf{v}_i, f_i), i = 0, 1, \dots, n$, where $\mathbf{v}_i \in \mathbb{R}^3$ and $f_i \in \mathbb{R}$ are 3D points and their associated function values, we consider an interpolant $g(\mathbf{x})$ as

$$g(\mathbf{x}) = \sum_{i=0}^n c_i \varphi_i(\mathbf{x}), \quad (1)$$

where $\phi_\rho(r) = \phi(r/\rho)$, $\phi(r) = (1 - r)_+^4(4r + 1)$ is the Wendland's CSRBF [35], ρ is the support size, and

$$\varphi_i(\mathbf{x}) = \frac{\phi_\rho(\lambda_i \|\mathbf{x} - \mathbf{v}_i\|)}{\sum_{j=0}^n \phi_\rho(\lambda_j \|\mathbf{x} - \mathbf{v}_j\|)} \quad (2)$$

is the normalized radial basis function. The value of λ_i is assigned as $\sqrt{\mu}/\eta_i$, where μ is a shape parameter and $\eta_i = \inf_{k \neq i} \|\mathbf{v}_k - \mathbf{v}_i\|$ reflects the impact of the density of points. The appropriate value of the shape parameter μ has been suggested for the computation on uniformly sampled grid points with global RBF in terms of Gaussian by Han and Hou [18]. However, their result cannot be applied to scattered data. Moreover, for CSRBF, the value of μ also affects the support size of each basis function. Based on our experimental tests, we can always choose $\mu = 0.1\rho^2$ to obtain satisfactory results.

According to [31], a surface interpolating a given data set can be represented by a weighted average of the values at the data points. Shepard constructed a group of weighting functions to interpolate two-dimensional data in [31].

Here, we assign the coefficients, c_i , with the values, f_i , at data points and use the normalized radial basis functions, φ_i , as weighting functions. In other words,

$$c_i = f_i, \quad (i = 0, 1, \dots, n), \quad (3)$$

and the surface function $g(\mathbf{x})$ is approximated by

$$g'(\mathbf{x}) = \sum_{i=0}^n f_i \varphi_i(\mathbf{x}). \quad (4)$$

However, different from two-dimensional interpolation in [31], the values, f_i , at 3D data points are unknown for surface reconstruction. In this section, methods are introduced to estimate the values of f_i s for fast RBF-based surface reconstruction.

3.1 Single-level quasi-interpolation

In this subsection, we demonstrate how our quasi-interpolation scheme works at a single level. Considering a set of scattered points $V = \{\mathbf{v}_i\}$ on a surface S , we assume that all the points are equipped with unit normals \mathbf{n}_i defining an inward-pointing orientation. We are going to generate a 3D scalar field $g'(\mathbf{x})$ whose zero level-set $g'(\mathbf{x}) = 0$ approximates S . In order to avoid adding off-points that lead to a bigger system of linear equations with interior and exterior constraints (ref. [11]), an interpolation function similar to the method in [28] is constructed as

$$g'(\mathbf{x}) \rightarrow g(\mathbf{x}) = \sum_{i=0}^n (c_i + h_i(\mathbf{x})) \varphi_i(\mathbf{x}) = 0, \quad (5)$$

where $h_i(\mathbf{x})$ is a local quadratic approximation of S in a small vicinity of \mathbf{v}_i , c_i and $\varphi_i(\mathbf{x})$ are defined as Eqs.3 and 2. The support size ρ is determined by the density of V . We organize the points in V into an octree in which each leaf cell contains no more than eight points. ρ is equal to $3/4$ of the average diagonal length of the leaf cells. In order to get $h_i(\mathbf{x})$, we create a local orthogonal coordinate system (u, v, w) at each point \mathbf{v}_i by setting the direction of \mathbf{n}_i as the positive direction of w . A quadric polynomial $w = w(u, v) \equiv au^2 + 2buv + cv^2$ is used to fitting the local surface of V in a vicinity of \mathbf{v}_i , where the coefficient a , b , and c can be determined by solving a small linear system (usually in the least-square form). Then, $h_i(\mathbf{x})$ can be set as $h_i(\mathbf{x}) = w - w(u, v)$. More details can be found in [28]. For exact interpolation, the coefficients c_i are determined by solving the system of linear equations 5. Here, we rewrite Eq.5 as

$$\psi(\mathbf{x}) = \sum_{i=0}^n c_i \varphi_i(\mathbf{x}) = - \sum_{i=0}^n h_i(\mathbf{x}) \varphi_i(\mathbf{x}). \quad (6)$$

The surface reconstruction problem with local quadric approximations then becomes a standard RBF fitting problem based on the function value constraints that $t_i = \psi(\mathbf{v}_i)$ with t_i denoting the right-hand side of Eq.6. With the above analysis of quasi-interpolation, the quasi-solution of this problem (i.e., the values of the coefficients c_i) can be found by

$$c_i = - \sum_{i=0}^n h_i(\mathbf{v}_i) \varphi_i(\mathbf{v}_i) = t_i \quad (7)$$

and $g'(\mathbf{x})$ for quasi-interpolating the given points can then be constructed as follows.

$$g'(\mathbf{x}) = \sum_{i=0}^n (- \sum_{j=0}^n h_j(\mathbf{v}_i) \varphi_j(\mathbf{v}_i) + h_i(\mathbf{x})) \varphi_i(\mathbf{x}). \quad (8)$$

The computation of the above quasi-interpolation is very fast as the evaluation for the coefficients c_i and h_i are independent of each other. However, for ones using compactly supported radial basis functions, it is weak in repairing incomplete regions in the sampled data sets. Although using adaptive support size as described in Section 4.2 can somewhat overcome this drawback, enlarging the support size always slows down the reconstruction. Moreover, an implicit surface reconstructed by single-level CSRBFs only has valid function values defined in a narrow band around the surface S . Therefore, the grids used for polygonization need to be smaller than the support size. This can further reduce the speed of mesh surface reconstruction.

3.2 Multi-level quasi-interpolation

A multi-level quasi-interpolation method is presented here to overcome problems mentioned above. Similar to [28], we construct a multi-scale hierarchy of point sets $\{V^1, V^2, \dots, V^M = V\}$ and refine fitting results by progressively adding basis functions according to the points at different levels. A base function $g^0(\mathbf{x}) = 0$ is defined at the coarsest level, and then recursively determines the set of approximating functions $g^k(\mathbf{x}) = g^{k-1}(\mathbf{x}) + \delta^k(\mathbf{x})$ ($k = 1, 2, \dots, M$) where $g^k(\mathbf{x})$ approximates a surface by the points in V^k . The shifting function $\delta^k(\mathbf{x})$ is defined by a form introduced in the aforementioned single-level quasi-interpolation as $\delta^k(\mathbf{x}) = \sum_{i=0}^{n^k} (c_i^k + h_i^k(\mathbf{x})) \varphi_i^k(\mathbf{x})$, where n^k is the number of points in V^k . Again, $h_i^k(\mathbf{x})$ s are local quadratic approximations formulated by applying a least square fitting to V^k . The exact solution of the coefficients can be determined by solving the system of linear equations $g^k(\mathbf{x}) = g^{k-1}(\mathbf{x}) + \delta^k(\mathbf{x}) = 0$. Here, they are computed in the manner of quasi-interpolation.

As in [28], we rewrite the linear system as

$$\sum_{i=0}^{n^k} c_i^k \varphi_i^k(\mathbf{x}) = -g^{k-1}(\mathbf{x}) - \sum_{i=0}^{n^k} h_i^k(\mathbf{x}) \varphi_i^k(\mathbf{x}). \quad (9)$$

For any point \mathbf{v}_i^k , we approximate its corresponding coefficient c_i^k by

$$c_i^k = -g^{k-1}(\mathbf{v}_i) - \sum_{j=0}^{n^k} h_j^k(\mathbf{v}_i) \varphi_j^k(\mathbf{v}_i). \quad (10)$$

Here, we adopt the strategy in [28] to compute the support size ρ^k and the subdivision levels M . The support size ρ^k at level k is recursively defined by $\rho^k = \rho^{k-1}/2$ and $\rho^1 = \alpha L$, where L is the diagonal length of the bounding box of V , and the parameter $\alpha = 0.75$ is chosen so that an octant of the bounding box is always covered by a ball of radius ρ^1 centered somewhere in the octant. The shape parameter μ^k is defined by $\mu^k = 4\mu^{k+1}$, and μ^M with M being the number of subdivision levels is specified by users. We found that a smooth surface can be reconstructed when μ^M is a value of about $\frac{1}{10}(\rho^M)^2$. The value of M can be determined by ρ^1 and $\hat{\rho}$, where $\hat{\rho}$ is equal to 3/4 of the average diagonal length of the leaf cells that contain no more than eight points of V . The equation $M = \lceil -\log_2(\hat{\rho}/(2\rho^1)) \rceil$ provided in [28] is used to determine the number of levels.

4 Shape Parameter and Support Size

There are two free parameters, the shape parameter μ and the support size ρ , in Eq.2 to control variational fitting results. As shown in Figs.1 and 2, reconstructed surface blurs the local fits on the given points when enlarging the shape parameter μ . For scattered points representing a complex shape, using a fixed shape parameter μ for all points is not

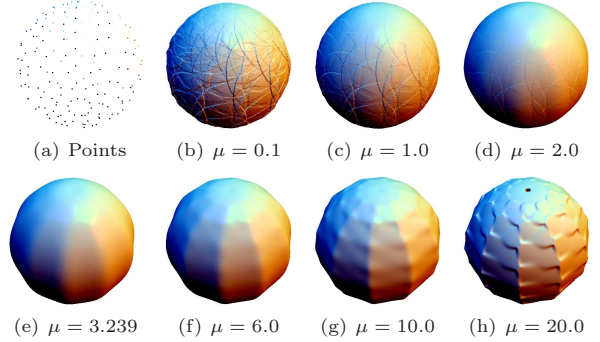


Figure 1: Single-level quasi-interpolation with different shape parameter μ . In (e), we set $\mu = 0.1\rho^2$

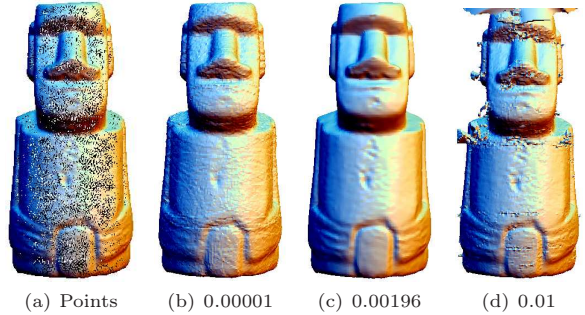


Figure 2: Multi-level quasi-interpolation with different shape parameters μ^M . In (c), we set $\mu^M = 0.1(\rho^M)^2$

appropriate. In fact, points with non-uniformities or holes are quite common in practice. Using a fixed support size ρ for all points may result in a failed reconstruction (see Figs.5(d) and 5(g) for examples). In order to solve these problems, we develop a scheme in this section to compute these parameters in an adaptive way.

4.1 Shape parameter

Reconstructed surfaces are different when different values of the shape parameter μ are used. We try to assign different value to the shape parameter at each point \mathbf{v}_i and let its value be determined by the intensity of a small vicinity of \mathbf{v}_i . By setting $\mu_i = \eta_i^2$, we can have $\lambda_i = 1$ at a point \mathbf{v}_i . Then, the normalized radial basis function in Eq.2 has a simpler form as

$$\varphi_i(\mathbf{x}) = \frac{\phi_\rho(\lambda_i \|\mathbf{x} - \mathbf{v}_i\|)}{\sum_{j=0}^n \phi_\rho(\lambda_j \|\mathbf{x} - \mathbf{v}_j\|)} = \frac{\phi_\rho(\|\mathbf{x} - \mathbf{v}_i\|)}{\sum_{j=0}^n \phi_\rho(\|\mathbf{x} - \mathbf{v}_j\|)}. \quad (11)$$

As shown in Figs.3 and 4, the quality of surface reconstructed in this way is similar to the exact RBF interpolation when working on uniformly sampled points.

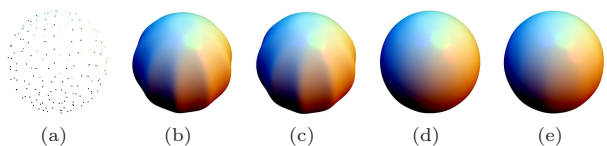
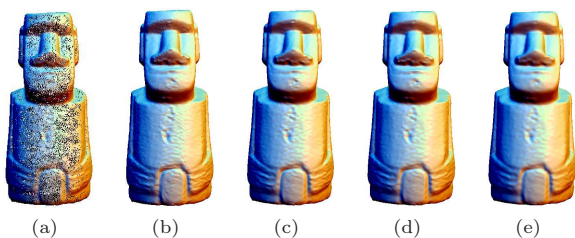
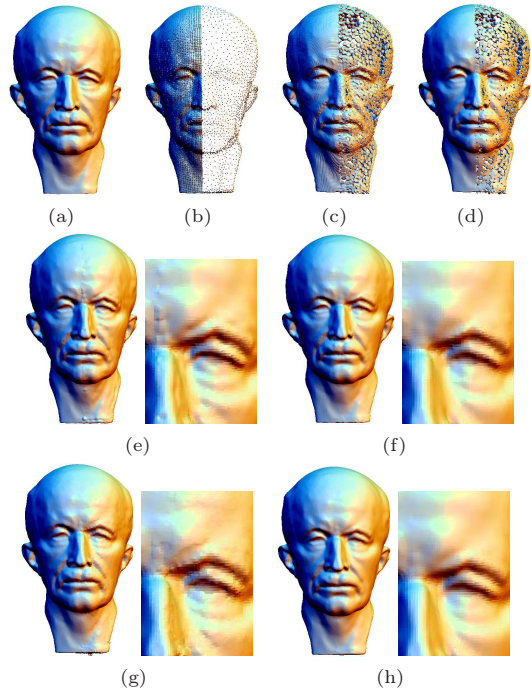


Figure 3: Comparisons between reconstructions by interpolation and quasi-interpolation from the points of a sphere model: (a) original points, (b) single-level CSRBF interpolation [28], (c) single-level quasi-interpolation with adaptive shape parameters, (d) multi-level CSRBF interpolation [28], and (e) multi-level quasi-interpolation with adaptive shape parameters.

Table 1: Time statistics for models with large scale points.

Fig.	Model	Number of Points	Multi-Level Interpolation [28]	Multi-Level Quasi-Interpolation	
			Time (Sec.)	Time (Sec.)	
				One-core	Eight-cores
6	Ramesses	570k	199.5	46.9	11
7	Budda	1,305k	1,004	110.8	25.3
8	Dancing children	2,174k	916	157	30.7
9	Statuette	5,000k	failed	341.5	62.1
10	Dragon	7,219k	failed	557.4	102.6

**Figure 4:** Comparisons between reconstructions by interpolation and quasi-interpolation from points of Moai model: (a) original points, (b) single-level CSRBF interpolation [28], (c) single-level quasi-interpolation with adaptive shape parameters, (d) multi-level CSRBF interpolation [28], and (e) multi-level quasi-interpolation with adaptive shape parameters.**Figure 5:** Reconstruction results by multi-level RBFs interpolation and quasi-interpolation: (a) original mesh, (b) a non-uniformly sampled point set from (a), (c) single-level CSRBF interpolation [28], (d) single-level quasi-interpolation with adaptive shape parameters, (e) single-level quasi-interpolation with adaptive shape parameters and adaptive support sizes ($\delta = 16$), (f) multi-level CSRBF interpolation [28], (g) multi-level quasi-interpolation, and (h) multi-level quasi-interpolation with adaptive shape parameters and adaptive support sizes ($\delta = 16$).

4.2 Adaptive support size

For points with high non-uniformity, as shown in Fig.5(b), reconstruction fails when conducting the single-level quasi-interpolation with a fixed support size for all points. Although the multi-level quasi-interpolation method with adaptive shape parameters can reconstruct surface successfully, the surface is not smooth in regions with sparse points (as shown in Fig.5(g)). This is because that the support sizes at those sparse points are too small to have enough points in its supporting field. Enlarging the support sizes globally is not reasonable as there will be too many neighbors for the points in dense regions, which leads to an expensive computation. In order to solve this problem, adaptive support sizes are adopted.

Based on the initial support size calculated by the method described in Section 3.2, we adjust the support size by considering the number of neighbors falling in a point’s local support. Letting δ be the average number of points in the initial support for every point (or setting the value of δ by users), if the number of points falling in the support $\rho_{\mathbf{v}}$ of a sample \mathbf{v} is smaller than δ , we enlarge the support size $\rho_{\mathbf{v}}$ by $\rho_{\mathbf{v}} = 1.1\rho_{\mathbf{v}}$ and check the number of points in the support again. The checking and enlarging are repeated until there are not less than δ points in the local support $\rho_{\mathbf{v}}$ of a sample point \mathbf{v} . With adaptive support sizes, single-level quasi-interpolation is immune to non-uniform sample points while multi-level quasi-interpolation can reconstruct smooth surfaces that successfully overcome the problem of high non-uniformity in the given point set (see the example shown in Fig.5). Note that, δ is the only parameter needs to be specified by users in our approach. For highly non-uniform noisy points, we always use $\delta = 16$.

5 Results and Discusses

We have implemented the proposed method with Microsoft Visual C++ and OpenGL. Point sets having up to several millions of points are used to test the method. Mesh surfaces are generated from reconstructed implicit surfaces by Bloomenthal’s method in [7]. All statistics pre-

sented in this paper are obtained by tests running on a PC with two Intel Xeon Quad E5440 CPUs at 2.83GHz plus 4GBytes RAM.

Figures 6, 7, 8 show the reconstruction results obtained by our method proposed in this paper and the results of exact RBF interpolation obtained by the method in [28]. We can hardly observe any visual difference between the reconstructed surfaces. The numbers of points in these examples are around 0.57 million, 1.3 million, and 2.2 million respectively. The statistics of computing times are listed in Table 1. It is easy to find that our method is much faster than the exact interpolation method in [28]. Moreover, as the method presented in this paper can be parallelized, the computing time can be further reduced when running on a PC with multi-cores (see Table 1). Figures 9, 10 are two other examples. There are around 5 million and 7.2 million points respectively which illustrate that our method can address large data. The computing times are 56.2 seconds and 102.6 seconds respectively while the method in [28] is failed for them.



Figure 8: Reconstructions from the points of a dancing-children model: (left) input point cloud, (middle) surface reconstructed by the interpolation method in [28], and (right) surface reconstructed by our quasi-interpolation method.

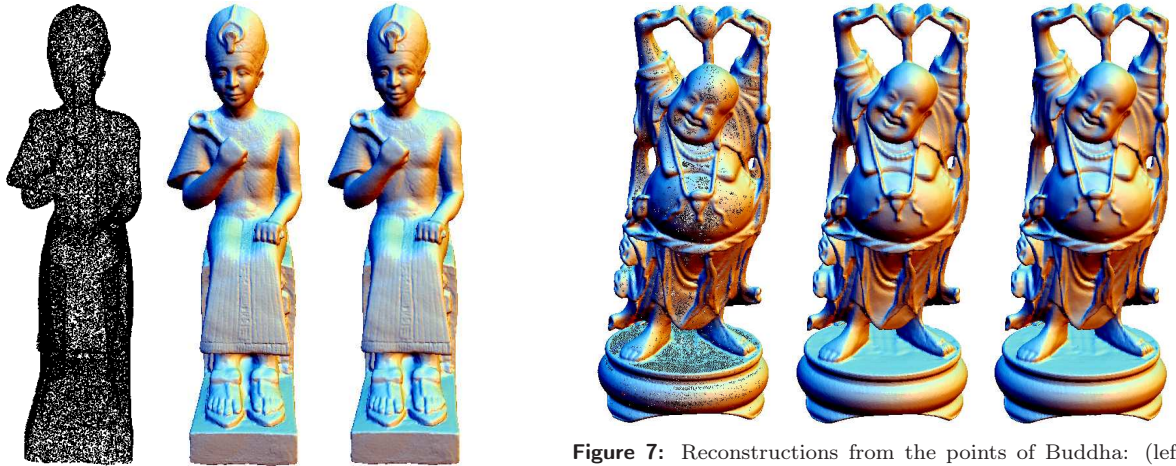


Figure 7: Reconstructions from the points of Buddha: (left) input point cloud, (middle) surface reconstructed by the interpolation method in [28], and (right) surface reconstructed by our quasi-interpolation method.

Figure 6: Reconstructions from the points of Ramesses: (left) input point cloud with 570k points, (middle) surface reconstructed by the interpolation method in [28], and (right) surface reconstructed by our quasi-interpolation method. For the illustration purpose, only 1/10 points are displayed in (left).

5.1 Quasi-interpolation with a given shape parameter

Figure 1 presents the result of applying our quasi-interpolation method (single-level) to a small and simple point set by using different shape parameters. When having a smaller μ , there are obvious bump-like artifacts. It is mainly due to the local approximation nature of this approach. Along with the increase of the shape parameter, the surface becomes smoother and smoother. However, a too large shape parameter makes the influence of a RBF center abruptly decrease when the distance value is greater than the support size. This results in a failed reconstruction. Figures 2 and 11 also demonstrate the same facts appeared in multi-level quasi-interpolation. In our implementation, a good reconstruction surface was obtained when the shape parameter was set as around 1/10 of the square of the support size for a model, such as Fig.1(e), Fig.2(c), and Fig.11(d). In practice, by the method presented in Section 4.1, the shape parameter can be hidden for the end-users of our surface reconstruction method.

5.2 Comparison with the exact CSRBF interpolation

To compare our quasi-interpolation method with the exact interpolation method in [28], we apply both methods to a few models (see Figs.3, 4, 5 and 12). The related statistics are listed in Table 2. Figures 3, 4, 12 show the reconstructions by single-level CSRBF interpolation [28], multi-level CSRBF interpolation [28], single-level quasi-interpolation with adaptive shape parameters, and multi-level quasi-interpolation with adaptive shape pa-

rameters. No obvious visual difference among them can be observed. As shown in Table 2 where the shape approximation errors are generated by the method presented in [33], the reconstruction results obtained by [28] and ours have shape errors in similar levels. However, our method is much faster. Figure 5 shows the reconstruction results from non-uniform points by multi-level quasi-interpolation with adaptive shape parameters or/and adaptive support sizes. If only adaptive shape parameters are adopted, the quasi-interpolation method will generate a surface that is not smooth in regions with highly sparse points. However, a smooth surface can be obtained by using adaptive support sizes in our quasi-interpolation approach. Figure 5 also shows the results from single-level CSRBF interpolation (Fig.5(b)) and single-level quasi-interpolation with adaptive support sizes (Fig.5(c)), where the reconstruction with a fixed support size is failed.

From the statistics of computing time listed in Table 2, quasi-interpolation is about 2-4 times faster than exact interpolation [28] even after using adaptive support sizes (which can slow down the computation). In fact, parallelizing the quasi-interpolation procedure on multi-cores can further improve its efficiency – see the statistics shown in Table 2.

5.3 Processing noisy point data

In order to illustrate the robustness of our method, we tested it with a non-uniform and noisy point set shown in Fig.13. To process noisy data, we extended our quasi-interpolation scheme by introducing a regularization parameter T_k , the value of which depends on the hierarchy level k . With this parameter T_k , the coefficients c_i of our

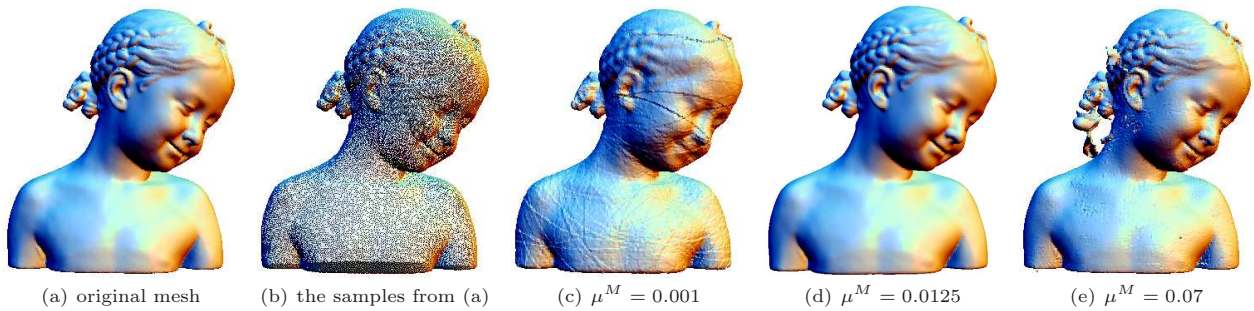


Figure 11: Reconstructions of a Bimba model by multi-level quasi-interpolation with different shape parameters μ . In d, we set $\mu^M = 0.1(\rho^M)^2$

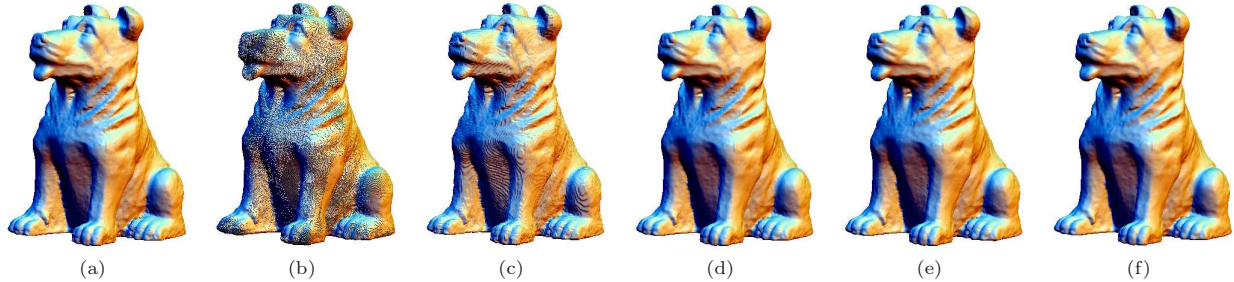


Figure 12: Comparisons between reconstructions by interpolation and quasi-interpolation from points of a dog model: (a) original mesh, (b) the sampling points from (a), (c) single-level CSRBF interpolation [28], (d) single-level quasi-interpolation with adaptive shape parameters, (e) multi-level CSRBF interpolation [28], and (f) multi-level quasi-interpolation with adaptive shape parameters.

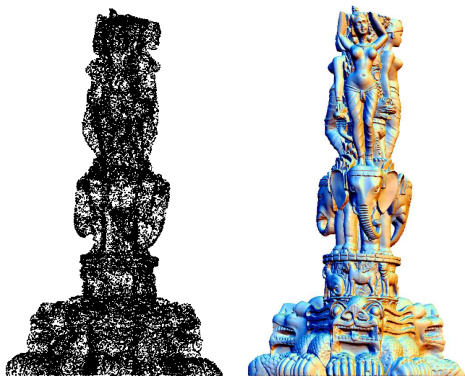


Figure 9: Reconstructions from the points of a statuette model: (left) input point cloud with 5M points, and (right) surface reconstructed by our quasi-interpolation method. For illustration, only 1/100 points are displayed in (left).

quasi-interpolation are determined by $c_i = f_i / (1 + T_k)$. From experimental tests, we found that good reconstruction results are obtained if setting $T_k = 0.5k$. For a noisy and non-uniform point set, the method in [28] reconstructs the surface (Fig. 13(b)) in 8.5 seconds. We apply our adaptive multi-level quasi-interpolation to this data. The threshold value for the adaptive support size was set as 10 (Fig. 13(c)) and 16 (Fig. 13(d)). The corresponding quasi-interpolation is slower than the method in [28] on single core (taking 9.5 and 12.8 seconds) but faster on eight cores (with 2.3 and 2.9 seconds respectively). The larger the threshold value is, the smoother the reconstructed surface is and the longer reconstruction time is needed.

For a point set with large noises, a preprocessing procedure (e.g., [24]) can be applied to remove the outliers. The resultant set retains only small noisy. A surface can be reconstructed from it by our method proposed in this

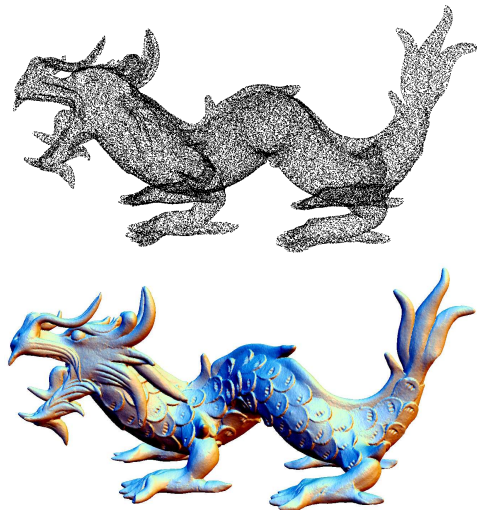


Figure 10: Reconstructions from the points of a dragon model: (top) input point cloud with 7.2M points, and (bottom) surface reconstructed by our quasi-interpolation method. For illustration, only 1/100 points are displayed in (left).

paper. An example is shown in Fig.14.

5.4 Comparison with Poisson reconstruction and MPU

We also compared our results with Poisson reconstruction [21] and MPU method [27] on an Igea model (Fig.15) with highly non-uniform points. As shown in Fig.15(c), Poisson method can reconstruct very smooth surfaces. However, the details can not be preserved well in sparse regions. Some time the surface reconstructed by Poisson reconstruction needs to be further process by mesh processing techniques. In the implementation of Poisson reconstruction provided by authors, they adapt the octree to

Table 2: The statistics for interpolation and quasi-interpolation methods.

Fig.	Model	Scale	Number of Points	Interpolation	Quasi-interpolation				Shape Approximation Error [†]	
					Time (Sec.)	Time(sec.)		μ	δ	Average
				One core		Eight cores				
5(e)	Max-Planck	3.12	67.4k		5.8	1	Adaptive	16	0.0011	0.027
5(f)				18				0.0011	0.023	
5(g)				3	0.7	Adaptive		0.00099	0.023	
5(h)				11	1.9	Adaptive	16	0.0011	0.027	
11(b)	Bimba	10	74.8k		3.7	0.8	0.001		0.0076	0.12
11(c)				3.6	0.8	0.01		0.0017	0.067	
11(d)				3.2	0.7	0.07		0.015	1.19	
12(c)	Dog	1	195.6k	28					0.00067	0.013
12(d)					2.2	0.5	Adaptive		0.00018	0.0063
12(e)				31					0.00022	0.011
12(f)				8.1	1.7	Adaptive		0.00021	0.011	

[†]Note that the errors between the reconstructed results and the original mesh are generated by the method in [33].

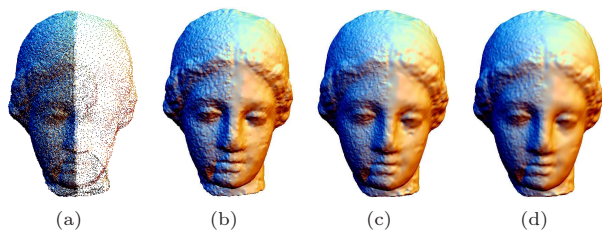


Figure 13: Reconstruction results on a noisy and non-uniform point set: (a) original noisy points, (b) multi-level CSRBF fitting result [28], (c) and (d) results from multi-level quasi-interpolation with an adaptive shape parameter and adaptive support sizes with different factors $\delta = 10$ and $\delta = 16$ respectively.



Figure 14: Reconstruction results on a large noisy point set: (left) original noisy points, (middle) the resultant point set after removing outliers [24], (right) result from multi-level quasi-interpolation with an adaptive shape parameter.

the sampling density. Therefore, the refinement stops automatically when there are few points in a cell of the tree. Based on this reason, the result of Poisson reconstruction shows no significant difference even if the required levels of refinement are assigned large. The results shown in this paper are generated by the program provided by Kazhdan et al. [21]. The reconstruction result of MPU method is good but not as good as ours – see the color map of shape approximation error generated by [33]. The statistics are shown in Table 3. The surfaces reconstructed by Poisson method and MPU method have already been the best results by choosing quite a few different parameters. During contouring, we try to generate mesh surfaces with a similar number of triangles.

Our method is similar to MPU method as both methods are based on local computation. A hierarchy of space

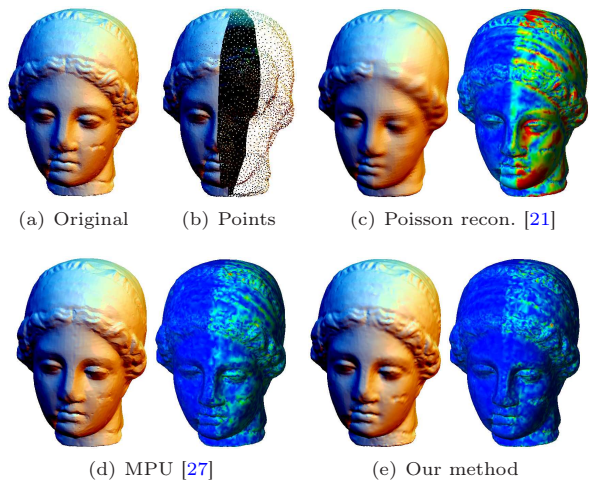


Figure 15: Reconstruction results on the points non-uniformly sampled from an Igea model. The color maps show the shape approximation errors on the results w.r.t. the original model. The blue color is for zero error, the red color is for the errors larger than 0.0067, and other colors denote the errors from 0.0 to 0.0067.

subdivision, partitions of unity, and local shape functions are the common characteristics of these two methods. There are also some differences between MPU method and ours.

- MPU method is only a partition of unity method while our method combines partition of unity method (providing a basic approximation) with RBF fitting (presenting more local details).
- MPU method adopts an adaptive subdivision based on an octree hierarchy. It computes local approximations only at the leaf cells of the octree; therefore, it is a local method. However, our method constructs a group of multi-scale functions from coarse to fine levels, and computes approximations at all levels which lead to a global fitting result. An implicit function is constructed in a bounding volume of a given point set, rather than just being in a narrow band around the surface as MPU. In Fig.16, there are cross sections of three implicit functions created from points, Bimba in Fig.11(d), Dog in Fig.12(f), and Igea in Fig.15(e).

6 Conclusion

In this paper, we present a CSRBF based quasi-interpolation approach for reconstructing an implicit sur-

Table 3: The statistics for three methods on an Igea model.

Fig.	Model	Scale	Number of Points	Method	Number of Resultant Triangles	Time [†] (Sec.)	Shape Approximation Error [‡]	
							Average	Maximal
15	Igea	1	100.4k	Poisson	751,498	39.1 (16.8)	0.0022	0.019
				MPU	751,416	total 14	0.00081	0.0098
				Our method	750,120	2.1 (42.8)	0.00062	0.0067

[†]The numbers in bracket refer to the time taken for the mesh generation from implicit surfaces. A method similar to the adaptations of the Marching Cubes [25] to octree representations is used in Poisson reconstruction while the method [7] is used in MPU and our reconstruction method. The mesh generation time for MPU is combined in its total reconstruction time.

[‡]Note that the errors between the reconstructed results and the original mesh are generated by the method in [33].

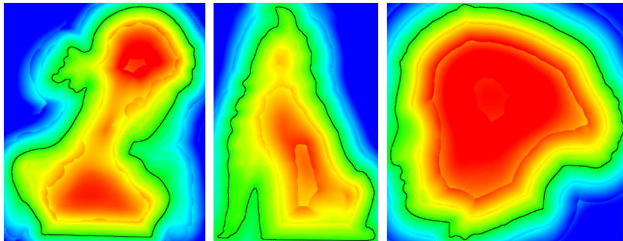


Figure 16: The cross sections of three signed distance fields: (left) Fig.11(d) in the front view, (middle) Fig.12(f) in the front view, and (right) Fig.15(e) in the right view. The colors show different field values where blue denotes negative maximum and red represents positive maximum.

face from scattered points. Our method is simple and stable as it does not need to solve large linear equation systems, which is a common step for almost all variational computation based surface reconstruction methods. Quasi-interpolation based surface reconstruction method demonstrates good performance on point sets with non-uniformity and noises. The quasi-interpolation on every data point can be computed independently so that the surface reconstruction procedure can be easily parallelized and run on a modern PC with multi-cores.

The limitations of our method are discussed below, which relates to the plan of our future work. Our method needs to construct a multi-scale hierarchy point sets which consumes a lot of memory. As a result, the scalability of our approach is not as good as MLS-based methods (e.g., [1, 5, 17]). From Fig.15 and Tab.3, considering the aspects of computation time, accuracy and recovery of sharp features, our method has not obvious advantage comparing with the MPU method. When applying the current implementation of our method to a point set with sharp features, all the sharp features are blurred on the reconstructed surfaces – therefore introducing large shape approximation errors in the relevant regions. How to reconstruct more accurate surfaces when the scattered data points contain sharp features will be one of our near future works. The strategy of prior work in [6] and [32] will be exploited. Another possible work is to implement this reconstruction method on the highly parallel architecture of GPUs. For those data with large noises, our method can deal with it after applying a preprocessing step (like [24]) on the input data.

Acknowledgments

The research presented in this paper was partially supported by the Hong Kong Research Grants Council (RGC) General Research Fund (GRF): CUHK/417508 and CUHK/417109 as well as Natural Science Foundation of China (NSFC) (Ref. No.: 61003125 and 61173119).

References

- [1] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. Silva. Point set surfaces. In *Proceedings of the conference on Visualization'01*, pages 21–28, 2001.
- [2] P. Alliez, D. Cohen-Steiner, Y. Tong, and M. Desbrun. Voronoi-based variational reconstruction of unoriented point sets. In *SGP'07: Proceedings of the fifth Eurographics symposium on Geometry Processing*, pages 39–48, 2007.
- [3] N. Amenta, M. Bern, and M. Kamvysseis. A new voronoi-based surface reconstruction algorithm. In *Proceedings of ACM SIGGRAPH 1998*, pages 415–421, 1998.
- [4] N. Amenta, S. Choi, and R. Kolluri. The power crust. In *Proceedings of 6th ACM symposium on solid modeling*, pages 249–260, 2001.
- [5] N. Amenta and Y. Kil. Defining point-set surfaces. *ACM Transactions on Graphics*, 23(3):264–270, 2004.
- [6] H. Avron, A. Sharf, C. Greif, and D. Cohen-Or. L1-sparse reconstruction of sharp point set surfaces. *ACM Transaction on Graphics*, 29(5):135:1–135:20, 2010.
- [7] J. Bloomenthal. Polygonization of implicit surfaces. *Computer Aided Geometric Design*, 5(4):341–355, 1988.
- [8] J.-D. Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Transactions on Graphics*, 3(4):266–286, 1984.
- [9] R. Bolle and B. Vemuri. On three-dimensional surface reconstruction methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–13, 1991.
- [10] F. Calakli and G. Taubin. Ssd: Smooth signed distance surface reconstruction. *Computer Graphics Forum*, 30(7):493–501, 2011.
- [11] J. Carr, R. Beatson, J. Cherrie, T. Mitchell, W. Fright, B. McCallum, and T. Evans. Reconstruction and representation of 3d objects with radial basis function. In *SIGGRAPH'2001: Proceedings of the 28th annual conference on computer graphics and interactive techniques*, pages 67–76, 2001.
- [12] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of ACM Siggraph 1996*, pages 303–312, 1996.
- [13] C. Debao. Degree of approximation by superpositions of a sigmoidal function. *Approx. Theory & its Appl.*, 9:17–28, 1993.
- [14] T. Dey, R. Dyer, and L. Wang. Localized cocone surface reconstruction. *Computers & Graphics*, 35(3):483–491, 2011.
- [15] T. Dey and S. Goswami. Tight cocone: a water-tight surface reconstructor. In *Proceedings of the 8th ACM symposium solid modeling applications*, pages 127–134, 2003.
- [16] T. Dey and S. Goswami. Provable surface reconstruction from noisy samples. *Comput. Geom. Theory Appl.*, 35(1):124–141, 2006.
- [17] G. Guennebaud and M. Gross. Algebraic point set surface. In *Proceedings of ACM Siggraph 2007*, pages 23.1–23.9, 2007.
- [18] X. Han and M. Hou. Quasi-interpolation for data fitting by the radial basis functions. In *GMP'08: Proceedings of Geometric Modeling and Processing*, pages 541–547, 2008.

- [19] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of ACM SIGGRAPH 1992*, pages 71–78, 1992.
- [20] T. Ju, F. Losasso, S. Schaefer, and J. Warren. Dual contouring of hermite data. In *SIGGRAPH'02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 339–346, 2002.
- [21] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *SGP'06: Proceedings of the fourth Eurographics Symposium on Geometry Processing*, pages 61–70, 2006.
- [22] R. Kolluri, J. Shewchuk, and J. O'Brien. Spectral surface reconstruction from noisy point clouds. In *Proceedings of the 2004 Eurographics/ACM Siggraph Symposium on Geometry Processing*, pages 11–21, 2004.
- [23] B. Lianas and F. Sainz. Constructive approximate interpolation by neural networks. *J. Comput. Appl. Math.*, 188:283–308, 2006.
- [24] S. Liu, K.-C. Chan, and C. Wang. Iterative consolidation of unorganized point clouds. *IEEE Computer Graphics and Applications*, page on line, 2012.
- [25] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163–169, 1987.
- [26] H. Mhaskar and C. Michelli. Approximation by superposition of sigmoidal andradial basis functions. *Adv. Appl. Math.*, 13:350–373, 1992.
- [27] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H. Seidel. Multi-level partition of unity implicits. *ACM Transaction on Graphics*, 22(3):463–470, 2003.
- [28] Y. Ohtake, A. Belyaev, and H. Seidel. 3d scattered data interpolation and approximation with multilevel compactly supported rbfs. *Graphical Models*, 67:150–165, 2005.
- [29] C. Oztireli, G. Guennebaud, and M. Gross. Feature preserving point set surfaces based on non-linear kernel regression. *Computer Graphics Forum*, 28(2):493–501, 2009.
- [30] O. Schall and M. Samozino. Surface from scattered points: A brief survey of recent developments. In B. Falcidieno and N. Magnenat-Thalmann, editors, *1st International Workshop on Semantic Virtual Environment*, pages 138–147, 2005.
- [31] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 ACM National Conference*, pages 517–524, 1968.
- [32] H. Sheung and C. Wang. Robust mesh reconstruction from unoriented noisy points. In *SPM'09: ACM Symposium on Solid and Physical Modeling 2009*, pages 13–24, Oct. 2009.
- [33] S. Silva, J. Madeira, and B. Santos. Polymeco - a polygonal mesh comparison tool. In *IV'05*, pages 842–847, 2005.
- [34] I. Tobor, P. Reuter, and C. Schlick. Reconstructing multi-scale variational partition of unity implicit surfaces with attributes. *Graphical Models*, 68(1):25–41, 2006.
- [35] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial basis functions of minimal degree. *Advances in Computational Mathematics*, 4:389–396, 1995.
- [36] W. Zhang and Z. Wu. Shape-preserving mq-b-splines quasi-interpolation. In *GMP'2004: Proceedings of Geometric Modeling and Processing*, pages 85–92, 2004.