Surface from Scattered Points

A Brief Survey of Recent Developments

Oliver Schall
Max-Planck-Institut für Informatik
schall@mpi-sb.mpg.de
http://www.mpi-sb.mpg.de/~schall

Marie Samozino
INRIA - Sophia Antipolis
Marie.Samozino@sophia.inria.fr
http://www-sop.inria.fr/geometrica/team/Marie.Samozino

Abstract

The paper delivers a brief overview of recent developments in the field of surface reconstruction from scattered point data. The focus is on computational geometry methods, implicit surface interpolation techniques, and shape learning approaches.

Keywords: surface reconstruction, scattered data

1 Introduction

In the mid-1980s the problem of surface reconstruction from scattered data was probably first addressed by Boissonnat [17]. Further attraction was lead to surface reconstruction from point cloud data by the research of Hoppe et al. [39]. Since then surface reconstruction became an active research field with two main surface reconstruction approaches.

One important research direction are Delaunay-based methods [2, 9, 11, 13, 16, 24, 27, 31] which usually involve the computation of a Delaunay complex or of dual structures using the scattered data. The surface is usually reconstructed by extraction from the previously computed Delaunay complex. Detailed surveys on computational geometry methods for surface reconstruction can be found in [22, 26]. A short survey is presented in [23].

The second important research direction are volumetric methods [14, 20, 56, 66]. Those approximate the scattered data by a 3D function. The surface can then be explicitly reconstructed by extracting it as the zero-level set of the computed function. A survey can be found in [53]. Recent research trends include Moving Least Squares (MLS) surfaces [1, 4, 5, 12] defined implicitly by the MLS projection operator.

Furthermore, different methods like zippering meshes obtained from range images [67] and statistical learning [41, 42, 73] have been developed.

Recent research in surface reconstruction from scattered data is also focused on reconstruction from scattered polygonal data [43, 64]. As point cloud data acquired from real world objects using for instance range scanners is always corrupted with noise, the reconstruction from noisy point data was recently addressed in [21, 28, 47, 58] as well as the processing of uncertain data in [12, 59]. Furthermore, point cloud data can be undersampled in regions of low visibility for the scanning device. Therefore, the problem of surface reconstruction from incomplete data is addressed in [25, 43, 63, 68].

A very comprehensive, yet preliminary, survey of surface reconstruction techniques can be found in [3]. We contribute to [3] by testing several surface reconstruction software tools and revealing their strengths and weaknesses. We also demonstrate that surface reconstruction results can be drastically improved if an appropriate preliminary filtering method [62] is applied to original scattered data.

The rest of this report is organized as follows. In Section 2 Delaunay-based methods are considered. Section 3 gives an overview of approaches computing an implicit surface. Learning-based methods are reviewed in Section 4. Finally, a filtering and denoising method is described in Section 5.

2 Delaunay-based Methods

In this section we are presenting an overview of Delaunay-based surface reconstruction approaches from scattered data. After reviewing ideas and properties of some approved methods we will present algorithms for the reconstruction of noisy point data being part of the most recent research.

Delaunay-based surface reconstruction approaches were mainly developed in the field of computational geometry. These methods are based on the idea that computing a surface from scattered point data P demands the exploration of the neighborhood of all samples in all directions to find possible neighbors for every sample point. As this is accomplished by the Delaunay triangulation it is predestinated for surface reconstruction. To find the final surface some methods compute a subset of the Delaunay triangulation. Those algorithms are usually referred to as restricted Delaunay-based methods. Two famous algorithms from this group are the Crust and the Cocone algorithm which will be described in the following section.

2.1 Crust & Cocone

The **Crust** algorithm for surface reconstruction was designed by Amenta and Bern [7] from the Crust algorithm for curve reconstruction [8]. The algorithm works as follows. Assuming we are given a point set P sampling an unknown smooth surface S the Crust algorithm first computes the Voronoi diagram $V(P)$ of P. Then it determines the set of poles Q of the Voronoi diagram.

Let V_p be the Voronoi cell of the sample point p in the Voronoi diagram. Then the positive pole p^+ is defined as the farthest Voronoi vertex of V_p from p. The pole vector $\mathbf{n}_p$ is defined as the vector pointing from the sample point p to p^+. The pole vector is a good approximation for the surface normal at p. The negative pole p^- is defined as the Voronoi vertex of V_p farthest from p in the opposite direction of the positive pole meaning the vector from p to p^- and the pole vector $\mathbf{n}_p$ enclose an angle of more than $\frac{\pi}{2}$ (see left image of Figure 2). The poles provide a good approximation for the medial axis of the surface sampled by P.

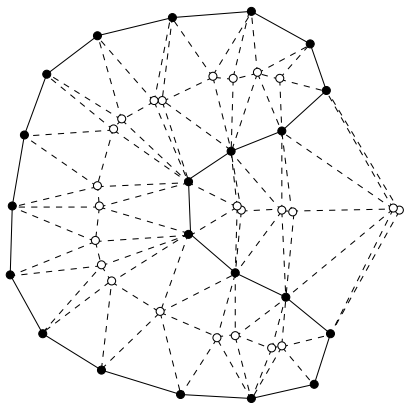

Figure 1: 2D example of the Delaunay triangulation $D(P \cup Q)$ of the sample points P (black) and their Voronoi poles Q (white). The Crust is indicated by solid lines.

After this the Crust algorithm computes the Delaunay triangulation $D(P \cup Q)$. To extract the candidate triangles of the surface that should be reconstructed from the computed Delaunay triangulation, the algorithm discards all triangles having not all vertices in P. The idea behind choosing the surface as a restriction of the Delaunay complex $D(P \cup Q)$ is that all triangles crossing the medial axis of the sampled surface S are removed (see Figure 1). The candidate triangles form in general not the resulting surface but they contain a surface that is homeomorphic to S if the sampling P is dense enough. Therefore, a manifold extraction stage is applied to find the final surface called the Crust.

The Crust algorithm was one of the first surface reconstruction methods that could provide guarantees. Amenta and Bern showed using a result of Edelsbrunner and Shah [32] that if P is an ϵ-sample for $\epsilon \leq 0.06$ then it is guaranteed that the candidate triangles contain the restricted Delaunay triangulation $D_S(P)$ which is the Delaunay triangulation $D(P)$ restricted to the sampled surface S. Although it is impossible to detect the triangles of $D_S(P)$ in the set of candidate triangles without knowing S this information is used in the manifold extraction step to compute an approximation of S.

The Crust algorithm demands the computation of two Delaunay triangulations being $D(P)$ and $D(P \cup Q)$ determining the runtime and memory complexity being $O(||P \cup Q||^2)$.

The **Cocone** algorithm was developed by Amenta, Choi, Dey and Leekha [10] from the Crust algorithm. It computes similar to the Crust algorithm a set of candidate triangles containing the restricted Delaunay triangulation $D_S(P)$ if the given sampling is dense enough. This is done by

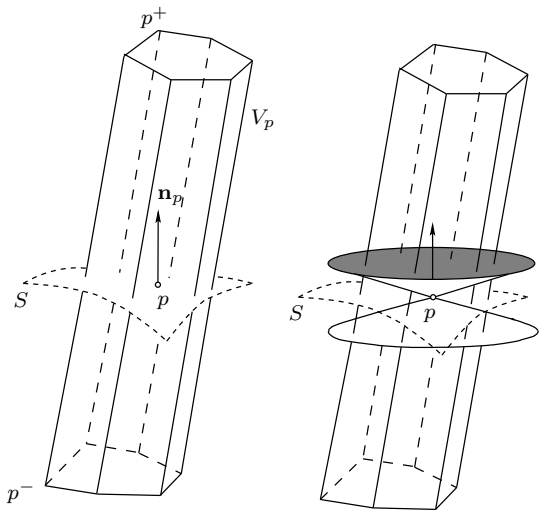

Figure 2: Left: Voronoi cell V_p of sample p intersecting surface S. The positive pole p^+, the pole vector $\mathbf{n}_p$ and the negative pole p^- are illustrated. Right: Cocone illustration. Adapted from [26].

first computing the Voronoi diagram $V(P)$ to find the pole vectors for every sample point. Then a cocone C_p for every sample point p is defined as the complement of the double cone with the apex at p, the pole vector $\mathbf{n}_p$ as axis and an opening angle of $\frac{3\pi}{4}$. The cocone C_p is clipped inside the Voronoi cell V_p. As the pole vector $\mathbf{n}_p$ approximates the surface normal at p the cocone determines a neighborhood around the tangent plane at p (see right image of Figure 2). After this the algorithm detects all Voronoi edges in V_p being intersected by the cocone C_p for every point p. The dual Delaunay triangles then build the candidate triangle set. After the whole candidate triangle set is determined the manifold extraction is performed to find the resulting surface. The cocone has the same guarantees like the Crust algorithm. It is proven that the restricted Delaunay triangulation $D_S(P)$ is part of the candidate triangle set if P is an ϵ-sample with $\epsilon \leq 0.06$.

The Cocone algorithm demands in contrast to the Crust algorithm only the computation of the Delaunay triangulation $D(P)$. Thus the runtime and memory complexity is reduced to $O(||D(P)||^2)$.

Important extensions of the presented methods are algorithms that produce watertight surfaces. These methods belong usually to the class of inside/outside labeling algorithms. They first compute a Delaunay triangulation of the scattered data. Then the resulting tetrahedra are labeled inside or outside depending whether the tetrahedron is inside the solid bounded by the scattered data or outside. After all tetrahedra are labeled the resulting surface can be extracted by retaining only triangles that are shared by one inside and one outside tetrahedron. As there is no remaining path from the inside of the bounded solid to the outside not passing a retained triangle the resulting mesh is watertight. Two important methods generating watertight reconstructions are Power Crust developed by Amenta, Choi and Kolluri [11] and Tight Cocone of Dey and Goswami [27]. Reconstructions of both algorithms are illustrated in Figure 3.

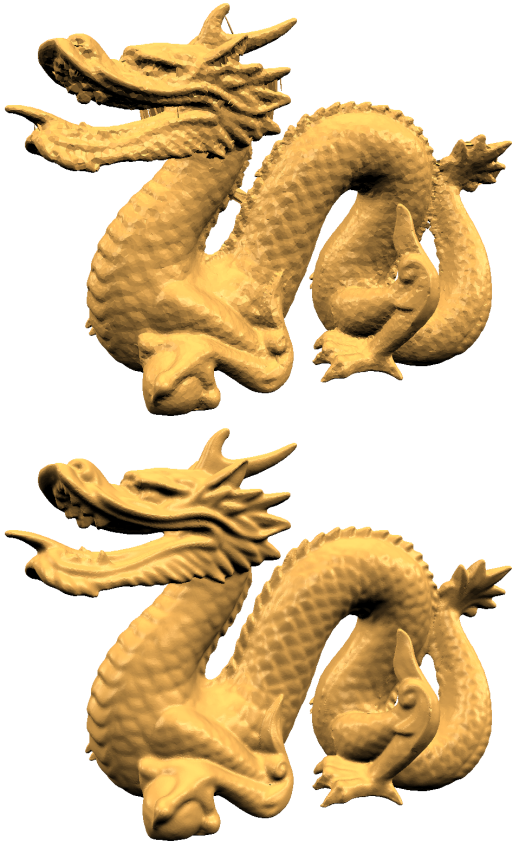

Figure 3: Power Crust (top) and Tight Cocone (bottom) reconstructions of the Dragon using samples from the VRIP files taken from the Stanford Scanning Repository.

2.2 Recent Developments

The presented computational geometry methods are supported by rigorous mathematical results and provide guarantees under the presented conditions. Unfortunately, the precondition that P is an ϵ-sampling with small ϵ does not often hold in practice. Therefore, computational geometry methods face difficulties while dealing with noisy data and undersampling (see top row of Figure 6). To overcome these problems recent research is focused on this area.

Dey and Goswami [28] present the Robust Cocone algorithm which applies an observation used

in the Power Crust algorithm. It was observed that the polar balls (balls surrounding the poles and touching the nearest samples) approximate the solid bounded by the sampled surface. If the balls of adjacent tetrahedra intersect deeply both tetrahedra belong to the same component being either inside or outside. On the other hand if balls intersect shallowly they belong to different components. Dey and Goswami showed that this might not be true for noisy regions. They also show that the polar balls in these regions have a small radius and can thus be detected. Robust Cocone only preserves the samples on the outer Delaunay balls and reconstructs a watertight surface only using them.

Kolluri et al. [47] introduce the Eigen Crust algorithm. The method computes a watertight surface reconstruction from noisy scattered data with outliers using spectral graph partitioning. To compute the surface the idea is again to label each tetrahedron of the Delaunay triangulation (or equivalently each Voronoi vertex) inside or outside. The Eigen Crust algorithm labels the Voronoi vertices in two steps. The first stage constructs a pole graph, whose nodes represent the poles. The edges of the pole graph are weighted according to the likelihood that pairs of tetrahedra are on the same side of the surface. This weighted graph is represented by the pole matrix. Finally, the smallest eigenvalue of the pole matrix is computed to divide the graph in two subgraphs containing inside and outside poles. To label also tetrahedra that are not duals of poles the Eigen Crust algorithm builds a second graph from the Voronoi vertices that are not poles. As the labeling of the non-pole vertices is more ambiguous the algorithm classifies them in order to produce a smooth surface with low genus.

3 Implicit Surface Interpolation

Given a set of points $X = \{x_i\} \subset \mathbb{R}^3$ scattered over a surface, the main idea behind most of the implicit surface interpolation techniques consists of building a function $y = f(x)$ whose zero level set $Z(f) = \{x : f(x) = 0\}$ approximates / interpolates X. Usually $y = f(x)$ is constructed as a composition (weighted sum) of simple primitives.

3.1 Radial Basis Functions (RBF)

Radial Basis Function (RBF) techniques are now standard tools for geometric data analysis [33, 53] in pattern recognition [44], statistical learning [36], and neural networks [37]. Properties of RBFs are widely studied in mathematical literature [18, 30, 40, 71]. (See also references therein.)

Given a scattered point dataset, we interpolate or approximate it by the zero level-set of a composite function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ defined as a linear combination of relatively simple primitives

$$f(x) = \sum_{i=1}^{m} \alpha_i \Phi(x, c_i) \tag{1}$$

where $\Phi(., c_i) : \mathbb{R}^3 \rightarrow \mathbb{R}$ are functions centered at c_i and α_i are the unknown weights [35].

We want to constrain the solution to be stable to translation and rotation of the point set. The functions Φ are thus given by:

$$\Phi(x, c_i) = \phi(\|x - c_i\|), \tag{2}$$

where $\|.\|$ denotes the Euclidian distance and $\phi : \mathbb{R}^+ \rightarrow \mathbb{R}$.

Relations between the RBF and variational approaches to scattered data interpolation / approximation are analyzed in [30, 37].

Reconstruction using Radial Basis Functions gives a smooth implicit interpolating or approximating surface, since both the implicit solution and its zero level set have the same continuity properties as the ones of the basis functions Φ.

Let $F = \{f_i\}$ be a set of n values of a function f at some scattered distinct points $X = \{x_i\} \in \mathbb{R}^3$. We want to find a function $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ such that $\forall i = j, \ldots, n$

$$f(x_j) = \sum_{i=1}^{m} \alpha_i \phi(\|x_j - c_i\|) = f_j \tag{3}$$

The reconstruction problem thus boils down to determining the vector $\alpha = \{\alpha_1, \ldots, \alpha_n\}$ by solving a linear system of equations given by the constraints (3). Since all constraints are located on the surface, all f_is are valued zero. In order to avoid the trivial solution $\alpha = \overrightarrow{0}$, we add interior and exterior constraints where the function is non zero and assign them the values $-d$ and d, respectively. We compute the weights $\alpha = [\alpha_i]$ using (3), and denoting $[\phi(\|x_i - x_j\|)] = A_{X,\Phi}$, we have to solve the following linear system:

$$A_{X,\Phi} \cdot \alpha = F. \tag{4}$$

In order to obtain a unique solution the matrix $A_{X,\Phi}$ is required to be invertible, at least on the subspace of $\vec{\alpha}$ vectors where the solution is searched. A common solution is to use the subspace such as:

$$\forall \alpha \in \mathbb{R}^n \qquad \sum_{i=1}^{n} \alpha_i p(x_i) = 0 \qquad \forall p \in \mathbb{P}_q \tag{5}$$

where $\mathbb{P}_q$ is the set of polynomials of order up to q. With this condition, the functions ϕ are *conditionally* positive definite [18]. Furthermore, to recover the right number of variables and unknowns, a polynomial $p \in \mathbb{P}_q$ will be added to (1)

$$f(x) = \sum_{i=1}^{m} \alpha_i \Phi(x, c_i) + p(x) \qquad (6)$$

Some conventional radial basis functions are: $\phi : \mathbb{R}^+ \rightarrow \mathbb{R}$

biharmonic RBF $\phi(r) = r$ with a linear polynomial

pseudo-cubic RBF $\phi(r) = r^3$ with a linear polynomial

triharmonic RBF $\phi(r) = r^3$ with a quadratic polynomial

thin plate RBF $\phi(r) = r^2 \log(r)$ with a linear polynomial

All functions listed above have an unbounded support. The corresponding equations lead to a dense linear system, therefore recovering a solution is tractable only for small data sets. To overcome this problem Morse et al. [55] use Gaussian as Compactly Supported RBFs to obtain a sparse interpolation matrix. More generally, a lot of other Compactly Supported RBFs (CSRBF) can be used for reconstruction as proposed in [69, 72] but they are not well-suited for reconstruction from incomplete data. To handle large and incomplete data sets two strategies have been proposed. One approach uses polyharmonic RBF (i.e. non-compactly supported functions) [19], reduces the number of centers by a greedy selection procedure and performs fast evaluation using the so-called Fast Multipole Method (FMM). Another approach consists of using locally supported functions [65], where the partition of unity is used for blending, and the function support is computed locally for all centers, as described in [58]. A multiresolution version of this approach has been proposed in [57].

We can notice that radially symmetric functions are not suited for piecewise smooth surface reconstruction. Dinh et al. [29] have presented a method using anisotropic basis functions to overcome this issue.

3.2 Partition of Unity (PU)

"Divide and conquer" is the main idea behind the Partition of Unity approach. The main idea consists of breaking the domain into smaller subdomains where the problem can be solved locally. The data is first approximated on each subdomain separately, and the local solutions are blent together using a weighted sum of local subdomain approximations. The weights are smooth functions and sum up to one everywhere on the domain.

Tobor et al. [65] combine the Partition of Unity method and the radial basis functions. Ohtake et al. [56] use weighted sums of different kinds of piecewise quadratic functions in order to capture the local shape of the surface. This way implicit surfaces from very large scattered point sets can be reconstructed.

Consider a global bounded domain Ω in an Euclidian space. Divide Ω into M mildly overlapping subdomains $\{\Omega_i\}_{i=1,\ldots,M}$ with $\Omega \subseteq \cup_i \Omega_i$. "Mildly overlapping" herein means that the intersection of two incident subdomains contains at most one data point. Together with this covering we construct a Partition of Unity, i.e. a family of non-negative continuous compactly supported functions $\{w_i\}_{i=1,\ldots,M}$ such that $Supp(w_i) \subseteq \Omega_i$ and $\sum_{i=1}^{M} w_i = 1$ everywhere. Let $X = \{x_i\} \in \mathbb{R}^3$ be a set of n points on the surface. For each cell Ω_i, a set $X_i = \{x \in X / x \in \Omega_i\}$ is built, and the surface is approximated on each subdomain by a local approximant f_i. The global function is then defined as a combination of the local functions as:

$$f(x) = \sum_{i=1}^{n} w_i(x) f_i(x). \qquad (7)$$

The condition $\sum_{i=1}^{M} w_i = 1$ can be obtained from any other set of smooth functions W_i by a normalization process:

$$w_i(x) = \frac{W_i(x)}{\sum_{j=1}^{n} W_j(x)}. \qquad (8)$$

The weighting function W_i determines the continuity of the global reconstruction function f. We can generate these functions using local geometry of the corresponding cell (distance function, center and radius of cell, etc.).

For domain decomposition, in general, space subdivision based trees are proposed (binary trees, octrees, etc.). The cells can have different shapes such as as axis-aligned bounding boxes, balls or axis-aligned ellipsoids. The degree of subdivision can be adapted to both local sample density and desired smoothness. The choice of the local fitting method is another degree of freedom: Radial Basis Functions are used in [65, 70], while quadrics are used in [56]. Furthermore, one can adapt the fitting strategy for each cell according to the number of points and distribution of associated normals. Reconstructing sharp features is this way possible as described in [56].

An interesting combination of the RBF and PU approaches was recently proposed in [58] where a

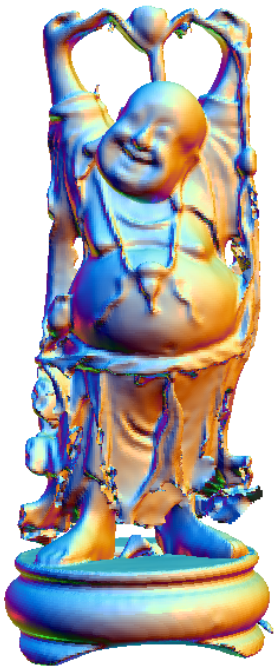

Figure 4: RBF+PU reconstruction of the Stanford Happy Buddha model from its original registered scans.

partition of unity is used to obtain an initial rough approximation of given scattered data and then RBFs are used to refine the PU approximation. An example of such PU+RBF reconstruction is shown in Figure 4.

3.3 Moving Least Squares (MLS)

Moving-least squares (MLS) surfaces have been introduced by Levin [48, 49]. The work of Alexa et al. [4, 5] first used MLS surfaces in point-based graphics. MLS surfaces are defined by a projection operator which projects points from a tubular neighborhood onto the surface. The neighborhood is usually defined by a union of balls centered at the input points $\mathbf{p}_i$. The traditional projection operator is computational expensive because of an associated non-linear optimization problem. A detailed survey of point-based techniques including a short description of the traditional projection operator can be found in [45].

Adamson and Alexa [1] propose a simpler projection technique for the definition of an implicit surface from point cloud data. They iteratively project a point $\mathbf{x}$ onto a plane defined by a weighted average of neighboring points

$$\mathbf{a}(\mathbf{x}) = \frac{\sum_j \Phi(||\mathbf{x} - \mathbf{p}_j||)\mathbf{p}_j}{\sum_j \Phi(||\mathbf{x} - \mathbf{p}_j||)}$$

and the normal $\mathbf{n}(\mathbf{x})$. If normals $\mathbf{n}_j$ for input points are given $\mathbf{n}(\mathbf{x})$ can be computed by averaging the input normals

$$\mathbf{n}(\mathbf{x}) = \frac{\sum_j \Phi(||\mathbf{x} - \mathbf{p}_j||)\mathbf{n}_j}{||\sum_j \Phi(||\mathbf{x} - \mathbf{p}_j||)\mathbf{n}_j||},$$

otherwise $\mathbf{n}(\mathbf{x})$ is determined by the normal of the weighted least-squares fitting plane of the points $\mathbf{p}_j$. After convergence the point $\mathbf{x}$ reaches a position on the MLS surface.

Recent work of Amenta and Kil [12] analyses the stability of the original projection operator for points that are not sufficiently close to the MLS surface. Furthermore, they present an explicit definition of MLS surfaces in terms of critical points of the energy function E_{MLS} along lines determined by a vector field.

4 Learning-based Methods

Learning-based methods used in surface reconstruction have their origin in the work of Kohonen [46] where Self-Organizing Maps (SOMs) were introduced. Later Fritzke [34] defined a specialized class of SOMs called the Growing Cell Structures (GCSs). Both neural networks have been used for surface reconstruction. Barhak and Fischer [15] use SOMs for grid fitting. GCSs are applied by Hoffmann and Várady [38] to free-from surface reconstruction. Yu [73] uses SOMs for surface reconstruction from scattered data. In this report we will focus on recent publications by Ivrissimtzis et al. [41, 42] using GCSs for the reconstruction of point cloud data.

The Growing Cell Structure presented by Ivrissimtzis et al. is an incrementally expanding neural network with triangle mesh connectivity (neural mesh) growing by splitting and removing vertices depending on their activity. The idea of the algorithm is to randomly sample the given input point set P and to use these samples as a training set for the initial neural mesh which is usually chosen as a tetrahedron. By processing samples the neural mesh M learns the geometry and the topology of the surface represented by the input point set. Geometry learning is accomplished by randomly choosing a sample $s \in P$ and moving the nearest vertex $v \in M$ towards s. After each update the 1-ring neighborhood of v is smoothed. To improve the connectivity and to allow growing of M the algorithm can introduce new vertices to the neural mesh by splitting the most active vertex as well as removing the least active vertex. The activity of all vertices is measured by a signal counter. The counter of the nearest vertex v to the sample s is increased by one to assign v a higher activity value. The signal counter of all other vertices is multiplied with a positive number $\alpha < 1$ to re-

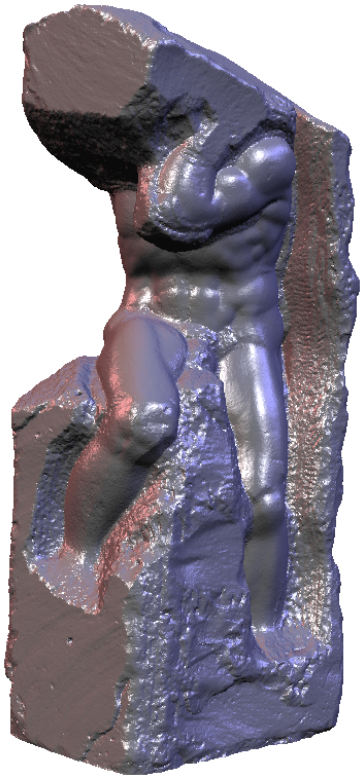

Figure 5: Neural mesh reconstruction (500K vertices) of the Atlas model with samples obtained from VRIP files in the Digital Michelangelo Archive [50].

move gradually the effect of old activity. Topology learning is achieved by triangle removal and boundary merging to allow the adaptation of M to boundaries and handles. As the growing of the neural mesh depends on the sampling density of P large triangles occur in regions of M approximating parts of the surface with a low sampling density. When the sampling density becomes insignificant such that the area of the corresponding triangles is over a defined threshold, they are removed and a boundary is created. If two boundaries are near to each other measured with an approximation of the Hausdorff distance the two boundaries are merged to create a handle.

The latest works of Ivrissimtzis et al. focus on the improvement of topological learning. One idea of [42] is to create an ensemble of neural meshes. An ensemble is a set of neural meshes with a combined output which has a smaller statistical error than any of its elements. To create a combined neural mesh from meshes with different connectivity a volumetric method is used.

Due to the sampling-based reconstruction the presented algorithm is able to process large data (see Figure 5). Furthermore, learning-based methods are well-suited for the reconstruction of noisy data. One drawback of the reviewed method is the reconstruction time complexity depending on the number of vertices of the neural mesh.

5 Filtering and Denoising

The performance of surface reconstruction approaches is usually reduced if the input data is noisy. One approach to overcome this problem is to filter and denoise the input before it is used for surface reconstruction [52, 54, 62].

Schall et al. [62] propose a method for dealing with uncertain and noisy surface scattered data. Given a set of noisy input points $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_N\}$ the method computes for every input point $\mathbf{p}_i$ a local uncertainty function measuring the contribution of $\mathbf{p}_i$ to the smooth surface S which is supposed to be reconstructed. The local functions are composed to a global uncertainty function whose local minima are determined. The computed minima are approximation centers of the sampled surface S and form a sparse set of points which are used as vertices of a mesh. The mesh can be reconstructed using an arbitrary meshing technique. The presented method robustly produces accurate smooth meshes with minimal topological noise. The top images of Figure 6 show reconstructions of the Stanford Bunny model from its original scans using Power Crust (left) and Tight Cocone (right). The bottom row of Figure 6 demonstrates reconstruction results created using the same Power Crust and Tight Cocone methods from the approximation centers generated according to [62].

6 Discussion and Conclusion

In this survey we presented several surface reconstruction methods. These methods are mainly either Delaunay-based and use a subset of a computed Delaunay complex to reconstruct the surface or represent the surface implicitly by the zero-level set of a defined function. Other techniques are learning-based or filter and denoise the input data.

The area of surface reconstruction is still a field with many open problems and research directions. Recent research trends focus on reconstruction of scattered polygonal data and noisy point cloud data. Furthermore, other methods avoid surface reconstruction but visualize the surface represented by the point cloud data directly by for instance ray-tracing Point Set Surfaces or using surface splatting techniques.

One of the recent approaches to processing scattered 3D point data consists of using point-based rendering primitives, as first suggested by Levoy and Whitted [51]. Efficient implementations of the

Figure 6: Reconstructions of the Stanford Bunny using Power Crust (left) and Tight Cocone (right) directly from the range scans (top row) and from the approximation centers (bottom row).

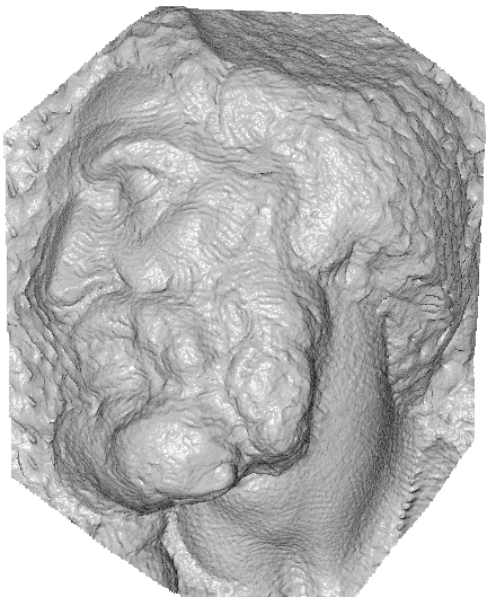

Figure 7: QSplat rendering of the St. Matthew model.

approach were presented in [60, 61] and in many subsequent works (see Figure 7). See the recent surveys [6, 45].

Acknowledgements

We would like to thank Tamal Dey, Nina Amenta and Szymon Rusinkiewicz for making their software available. Furthermore, we thank Yutaka Ohtake for providing us his software and Ioannis Ivrissimtzis for images from his papers. We are grateful to the anonymous reviewers of this paper for their constructive comments and useful suggestions. The Atlas and St. Matthew datasets are courtesy of the Digital Michelangelo Project. The Bunny, Dragon and Buddha datasets are courtesy of the Stanford 3D scanning repository. This research was supported in part by the European FP6 NoE grant 506766 (AIM@SHAPE).

References

[1] A. Adamson and M. Alexa. Approximating and intersecting surfaces from points. *Symposium on Geometry Processing 2003*, pages 245–254, 2003.

[2] U. Adamy, J. Giesen, and M. John. Surface reconstruction using umbrella filters. *Computational Geometry*, 21(1-2):63–86, 2002.

[3] Aim@Shape. Survey acquisition and reconstruction. Technical report, 2004.

[4] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Computing and rendering point set surfaces. *IEEE TVCG*, 9(1):3–15, 2003.

[5] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, and C. T. Silva. Point set surfaces. *IEEE Visualization 2001*, pages 21–28, October 2001.

[6] M. Alexa, M. Gross, M. Pauly, H. Pfister, M. Stamminger, and M. Zwicker. *Point-Based Computer Graphics*. SIGGRAPH'04 Course Notes #6, SIGGRAPH-ACM publication, 2004.

[7] N. Amenta and M. Bern. Surface reconstruction by voronoi filtering. *Discr. Comput. Geom.*, 22:481–504, 1999.

[8] N. Amenta, M. Bern, and D. Eppstein. The crust and the β-skeleton: combinatorial curve reconstruction. *Graphical Models and Image Processing*, 60:125–135, 1998.

[9] N. Amenta, M. Bern, and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. In *Proceedings of ACM SIGGRAPH 1998*, pages 415–421, 1998.

[10] N. Amenta, S. Choi, T. K. Dey, and N. Leekha. A simple algorithm for homeomorphic surface reconstruction. *Internat. J. Comput. Geom. & Appl.*, 12:125–141, 2002.

[11] N. Amenta, S. Choi, and R. Kolluri. The power crust. In *Proceedings of 6th ACM Symposium on Solid Modeling*, pages 249–260, 2001.

[12] N. Amenta and Y. J. Kil. Defining point-set surfaces. *ACM Transactions on Graphics*, 23(3):264–270, August 2004. Proceedings of SIGGRAPH 2004.

[13] M. Attene and M. Spagnuolo. Automatic surface reconstruction from point sets in space. *Computer Graphics Forum*, 19(3):457–465, 2000. Proceedings of EUROGRAPHICS 2000.

[14] C. L. Bajaj, F. Bernardini, and G. Xu. Automatic reconstruction of surfaces and scalar fields from 3D scans. In *Proceedings of ACM SIGGRAPH 95*, pages 109–118, August 1995.

[15] J. Barhak and A. Fischer. Adaptive reconstruction of freeform objects with 3D SOM neural network grids. In *Pacific Graphics 01, Conference Proceedings*, pages 97–105, 2001.

[16] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.

[17] J.-D. Boissonnat. Geometric structures for three-dimensional shape representation. *ACM Transactions on Graphics*, 3(4):266–286, October 1984.

[18] M. D. Buhman. *Radial basis functions: theory and implementations*, volume 12. Cambridge monographs on applied and computational mathematics edition, 2003.

[19] J. Carr, W. Fright, and R. Beatson. Surface interpolation with radial basis functions for medical imaging. *IEEE Transactions Med. Imag.*, 16(1), 1997.

[20] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. Reconstruction and representation of 3D objects with radial basis functions. In *Proceedings of ACM SIGGRAPH 2001*, pages 67–76, August 2001.

[21] J. C. Carr, R. K. Beatson, B. C. McCallum, W. R. Fright, T. J. McLennan, and T. J. Mitchell. Reconstruction and representation of 3D objects with radial basis functions. In *Proceedings of ACM GRAPHITE 2003*, pages 119–126, Melbourne, Australia, February 2003.

[22] F. Cazals and J. Giesen. Delaunay triangulation based surface reconstruction: Ideas and algorithms. Technical report, INRIA, November 2004. Rapport de recherche 5393.

[23] F. Cazals, J. Giesen, and M. Yvinec. Delaunay triangulation based surface reconstruction: a short survey. Technical report, INRIA, November 2004. Rapport de recherche 5394.

[24] D. Cohen-Steiner and F. Da. A greedy Delaunay-based surface reconstruction algorithm. Technical report, INRIA, September 2002. Rapport de recherche 4564.

[25] J. Davis, S. R. Marschner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion. In *First International Symposium on 3D Data Processing, Visualization, and Transmission*, Padua, Italy, June 2002.

[26] T. K. Dey. *Curve and surface reconstruction*. CRC Press, 2nd edition, 2004.

[27] T. K. Dey and S. Goswami. Tight cocone: A watertight surface reconstructor. In *Proc. 8th ACM Sympos. Solid Modeling Applications*, pages 127–134, 2003.

[28] T. K. Dey and S. Goswami. Provable surface reconstruction from noisy samples. In *Proc. 20th ACM Sympos. Comput. Geom.*, 2004.

[29] H. Q. Dinh, G. Turk, and G. Slabaugh. Reconstructing surfaces using anisotropic basis functions. In *International Conference on Computer Vision (ICCV) 2001*, volume 2, pages 606–613, 2001.

[30] J. Duchon. Spline minimizing rotation-invariant semi-norms in Sobolev spaces. In W. Schempp and K. Zeller, editors, *Constructive Theory of Functions of Several Variables*, volume 571 of *Lecture Notes in Mathematics*, pages 85–100, 1977.

[31] H. Edelsbrunner and E. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, January 1994.

[32] H. Edelsbrunner and N. Shah. Triangulating topological spaces. *Internat. J. Comput. Geom. & Appl.*, 7:365–378, 1997.

[33] R. Franke and G. Nielson. Smooth interpolation of large sets of scattered data. *International Journal for Numerical Methods in Engineering*, 15(11):1691–1704, 1980.

[34] B. Fritzke. Growing cell structures - a self-organizing network for unsupervised and supervised learning. Technical Report ICSTR-93-026, International Computer Science Institute, Berkeley, 1993.

[35] F. Girosi, M. Jones, and T. Poggio. Priors stabilizers and basis functions: From regularization to radial, tensor and additive splines. Technical Report AIM-1430, 1993.

[36] H. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[37] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, New Jersey, USA, 1999.

[38] M. Hoffmann and L. Várady. Free-form modeling surfaces for scattered data by neural networks. *Journal for Geometry and Graphics*, 1:1–6, 1998.

[39] 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.

[40] A. Iske. *Multiresolution Methods in Scattered Data Modelling*. Springer-Verlag, Heidelberg, Germany, 2004.

[41] I. Ivrissimtzis, W.-K. Jeong, and H.-P. Seidel. Using growing cell structures for surface reconstruction. In *Shape Modeling International 2003*, pages 78–86, 2003.

[42] I. Ivrissimtzis, Y. Lee, S. Lee, W.-K. Jeong, and H.-P. Seidel. Neural mesh ensembles. In Y. Aloimonos and G. Taubin, editors, *3DPTV 2004*, Los Alamitos, USA, 2004. IEEE.

[43] T. Ju. Robust repair of polygonal models. *ACM Transactions on Graphics*, 23(3), August 2004. Proceedings of SIGGRAPH 2004.

[44] M. Kirby. *Geometric Data Analysis*. John Wiley & Sons, 2001.

[45] L. Kobbelt and M. Botsch. A survey of point-based techniques in computer graphics. *Computers & Graphics*, 28(4):801–814, 2004.

[46] T. Kohonen. Self-organized formation of topologically correct feature maps. In *Biological Cybernetics 43*, pages 59–69, 1982.

[47] R. Kolluri, J. R. Shewchuk, and J. F. O'Brien. Spectral surface reconstruction from noisy point clouds. In *Symposium on Geometry Processing*, pages 11–21. ACM Press, July 2004.

[48] D. Levin. The approximation power of moving least-squares. *Math. Comput.*, 67(224):1517–1531, 1998.

[49] D. Levin. *Geometric Modeling for Scientific Visualization*, chapter Mesh-independent surface interpolation, pages 37–49. Springer-Verlag, 2003.

[50] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The Digital Michelangelo Project: 3D scanning of large statues. In *Proceedings of ACM SIGGRAPH 2000*, pages 131–144, July 2000.

[51] M. Levoy and M. Whitted. The use of points as a display primitive. Technical Report 85-022, Computer Science Department, University of North Carolina at Chapel Hill, January 1985.

[52] L. Linsen. Point cloud representation. Technical report, Fakultät für Informatik, Universität Karlsruhe, 2001. 2001-3.

[53] S. K. Lodha and R. Franke. Scattered data techniques for surfaces. In *Dagstuhl '97, Scientific Visualization*, pages 181–222. IEEE Computer Society, 1999.

[54] B. Mederos, L. Velho, and L. H. de Figueiredo. Smooth surface reconstruction from noisy clouds. *Journal of the Brazilian Computing Society*, 2004.

[55] B. S. Morse, T. S. Yoo, D. T. Chen, P. Rheingans, and K. R. Subramanian. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In *SMI '01: Proceedings of the International Conference on Shape Modeling & Applications*, pages 89–98. IEEE Computer Society, 2001.

[56] Y. Ohtake, A. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel. Multi-level partition of unity implicits. *ACM Transactions on Graphics*, 22(3):463–470, July 2003. Proceedings of SIGGRAPH 2003.

[57] Y. Ohtake, A. G. Belyaev, and H.-P. Seidel. A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions. In *Shape Modeling International 2003*, pages 153–161, Seoul, Korea, May 2003.

[58] Y. Ohtake, A. G. Belyaev, and H.-P. Seidel. 3D scattered data approximation with adaptive compactly supported radial basis functions. In *Shape Modeling International 2004*, Genova, Italy, June 2004.

[59] M. Pauly, N. J. Mitra, and L. J. Guibas. Uncertainty and variability in point cloud surface data. In *Eurographics Symposium on Point-Based Graphics*, pages 77–84, Zurich, Switzerland, June 2004.

[60] H. Pfister, M. Zwicker, J. van Baar, and M. Gross. Surfels: Surface elements as rendering primitives. In *Proceedings of ACM SIGGRAPH 2000*, pages 335–342, July 2000.

[61] S. Rusinkiewicz and M. Levoy. QSplat: a multiresolution point rendering system for large meshes. In *Proceedings of ACM SIGGRAPH 2000*, pages 343–352, 2000.

[62] O. Schall, A. G. Belyaev, and H.-P. Seidel. Sparse meshing of uncertain and noisy surface scattered data. Research Report MPI-I-2005-4-002, Max-Planck-Institut für Informatik, February 2005.

[63] A. Sharf, M. Alexa, and D. Cohen-Or. Context-based surface completion. *ACM Transactions on Graphics*, 23(3):878–887, August 2004. Proceedings of SIGGRAPH 2004.

[64] C. Shen, G. F. O'Brien, and J. R. Shewchuk. Interpolating and approximating implicit surfaces from polygon soup. *ACM Transactions on Graphics*, 23(3), August 2004. Proceedings of SIGGRAPH 2004.

[65] I. Tobor, P. Reuter, and C. Schilck. Efficient reconstruction of large scattered geometric datasets using the partition of unity and radial basis functions. *Journal of WSCG 2004*, 12:467–474, 2004.

[66] G. Turk, H. Q. Dinh, J. O'Brien, and G. Yngve. Implicit surfaces that interpolate. In *Shape Modelling International 2001*, pages 62–71, Genova, Italy, May 2001.

[67] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proceedings of ACM SIGGRAPH 1994*, pages 311–318, July 1994.

[68] J. Verdera, V. Caselles, M. Bertalmío, and G. Sapiro. Inpainting surface holes. In *IEEE International Conference on Image Processing (ICIP 2003)*, Barcelona, Spain, September 2003.

[69] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4:389–396, 1995.

[70] H. Wendland. Fast evaluation of radial basis functions: Methods based on partition of unity. *Approximation Theory X: Wavelets, Splines, and Applications*, pages 473–483, 2002.

[71] H. Wendland. *Scattered Data Approximation*. Cambridge Monographs on Applied and Computational Mathematics (No. 17). Cambridge University Press, 2004.

[72] Z. Wu. Compactly supported positive definite radial functions. *Advances in Computational Mathematics*, 4:283–292, 1995.

[73] Y. Yu. Surface reconstruction from unorganized points using self-organizing neural networks. In *Proc. of IEEE Visualization '99*, pages 61–64, 1999.