Robust Surface Reconstruction via Dictionary Learning

Shiyao Xiong[†] Juyong Zhang[†*] Jianmin Zheng[‡] Jianfei Cai[‡] Ligang Liu[†]

[†]University of Science and Technology of China [‡]Nanyang Technological University

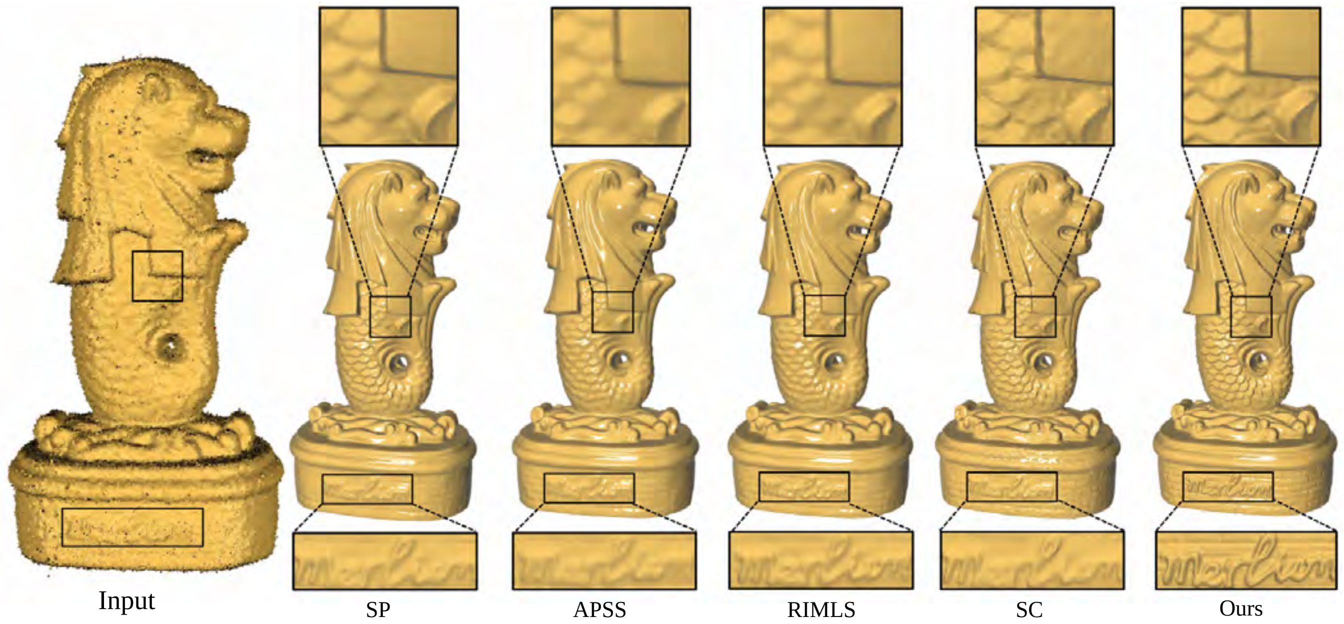


Figure 1: *Reconstruction of the Merlion model by our proposed method and state-of-the-art: SP [Kazhdan and Hoppe 2013], APSS [Guennebaud and Gross 2007], RIMLS [Öztireli et al. 2009], and SC [Dey and Wang 2013]. It can be seen that our reconstruction can recover different levels of details better.*

Abstract

Surface reconstruction from point cloud is of great practical importance in computer graphics. Existing methods often realize reconstruction via a few phases with respective goals, whose integration may not give an optimal solution. In this paper, to avoid the inherent limitations of multi-phase processing in the prior art, we propose a unified framework that treats geometry and connectivity construction as one joint optimization problem. The framework is based on dictionary learning in which the dictionary consists of the vertices of the reconstructed triangular mesh and the sparse coding matrix encodes the connectivity of the mesh. The dictionary learning is formulated as a constrained $\ell_{2,q}$-optimization ($0 < q < 1$), aiming to find the vertex position and triangulation that minimize an energy function composed of point-to-mesh metric and regularization. Our formulation takes many factors into account within the same framework, including distance metric, noise/outlier resilience, sharp feature preservation, no need to estimate normal, etc., thus providing a global and robust algorithm that is able to efficiently recover a piecewise smooth surface from dense data points with imperfections. Extensive experiments using synthetic models, real world models, and publicly available benchmark show that our method outperforms the state-of-the-art in terms of accuracy, robustness to noise and outliers, geometric feature and detail preservation, and mesh connectivity.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

Keywords: Surface reconstruction, point cloud, distance metric, $l_{2,q}$-optimization, dictionary learning, sparse coding

Links: DL PDF WEB VIDEO

1 Introduction

This paper considers the problem of surface reconstruction that takes as input a set of dense, unorganized points sampled from a subjacent, piecewise smooth surface and outputs a triangular mesh to approximate the surface. It is by nature an inverse problem, which is often ill-posed without prior assumptions. The input points usually have artifacts such as noise, outliers or missing data, which are inherent in acquisition process. The normal at points may be absent and the point sampling may not be uniform. This type of input and output is common in practice [Berger et al. 2013]. The co-existence of sharp features such as crease edges and corners and data artifacts makes the problem very challenging.

[*]Corresponding author: juyong@ustc.edu.cn (Juyong Zhang)

ACM Reference Format
Xiong, S., Zhang, J., Zheng, J., Cai, J., Liu, L. 2014. Robust Surface Reconstruction via Dictionary Learning. ACM Trans. Graph. 33, 6, Article 201 (November 2014), 12 pages. DOI = 10.1145/2661229.2661263 http://doi.acm.org/10.1145/2661229.2661263.

0730-0301/14/11-ART201 $15.00.
DOI: http://dx.doi.org/10.1145/2661229.2661263

Over the past two decades extensive work has been done on surface reconstruction [Berger et al. 2014]. Existing methods can roughly be classified into combinatorial approaches and implicit approaches. Both involve a few separate phases. For example, while combinatorial approaches may require denoising, vertex subset determination, feature detection, and triangulation, implicit approaches require normal estimation, level set function construction and iso-surfacing. However, some of these phases such as normal estimation are themselves challenging tasks. They are designed for their respective goals. As a result, the integration of them may not achieve the best performance, especially when the input data have imperfections.

Observing the inherent limitation of such multi-phase processing in the prior art, in this paper we advocate a totally different path that is to work directly on the ultimate goal of surface reconstruction, which is to find the geometry and connectivity of the reconstructed mesh best fitting the input point cloud. We propose a unified framework to jointly optimize geometry and connectivity for surface reconstruction. Specifically, the framework is based on sparse dictionary learning in which the vertices of the reconstructed triangular mesh are treated as the dictionary elements and the connectivity of the mesh is encoded in the sparse coding matrix. A key observation here is that the point cloud contains redundancy in terms of shape and each point in it can be sparsely encoded as a convex combination of triangle vertices. This sparse representation conveys connectivity information of the mesh if we impose constraints on the coding. Dictionary learning has been successfully used in sparse signal representation and reconstruction [Aharon et al. 2006; Wright et al. 2010], where signals are represented as a linear combination of only a few elements that form a dictionary.

Our dictionary learning is formulated as a constrained optimization problem that minimizes an energy function composed of point-to-mesh metric and regularization. The constraints are introduced to induce the dictionary sparsity and the mesh manifoldness. While many combinatorial methods use the distance between the reconstructed mesh vertices and the sampling points, we propose to use the distance of the sampling points to the reconstructed mesh, which is more accurate to describe the deviation of the mesh from the point cloud model. Moreover, note that the data obtained by laser range scanners or structured light depth camera typically contain spatially varying noise or uncertainty [Tošić et al. 2010], for which the Gaussian noise assumption in conventional dictionary learning is not appropriate. Therefore, we define our distance metric based on sparse norm $\ell_{2,q}, q \in (0,1)$. Our experiment shows that this sparse norm can properly handle noise and outliers and encourage non-smooth features in the solution. Since the $\ell_{2,q}$-norm is not differentiable, we present a tailored augmented Lagrangian method to efficiently solve the optimization problem.

It is also worth pointing out that the normal information is not necessary to our method, which enables our method to avoid the pitfall of normal estimation that is required in many existing surface reconstruction algorithms. Normal estimation is known to be very sensitive to noisy data and sharp features [Huang et al. 2009; Huang et al. 2013], and the unreliable normals often cause large reconstruction errors. In addition, our method can handle watertight models and open mesh models in the same way while some existing methods can only deal with watertight models.

This work offers two main contributions. First, we present a dictionary learning based framework for surface reconstruction. This framework takes all considerations into account, including distance metric, noise/outlier resilience, sharp feature preservation, no need to compute normal explicitly, etc. It thus provides a global solution with all the data and processes being jointly considered in one phase. Second, an effective algorithm for the proposed constrained $\ell_{2,q}$-optimization, $q \in (0,1)$, is designed to iteratively optimize the dictionary and sparse coding matrix, which progressively update the mesh towards the solution. Particularly, it should be stressed that in our approach topology optimization is performed as part of the reconstruction process in order to align triangle edges to features, which was not seen in the prior art. Extensive experiments demonstrate that our method is able to produce highly accurate reconstruction, preserves well different levels of geometric details and sharp features, and is robust to noise even with different scales and outliers. Fig. 1 shows such an example where our reconstruction can recover different levels of details and geometric features well while the other methods either over-smooth or lose the details.

2 Related Work

Combinatorial surface reconstruction methods. The methods in this class typically generate a triangular mesh by interpolating a subset of the input points as vertices. Classic combinatorial algorithms such as Delaunay triangulation [Kolluri et al. 2004] and alpha shapes [Edelsbrunner and Mücke 1994] are often employed for this purpose. The well-known algorithms are Cocone [Dey and Giesen 2001; Amenta et al. 2002], which takes the point cloud in 3D as input and outputs a piecewise linear approximation using Delaunay triangles, and its various extensions such as [Dey and Wang 2013]. A good survey can be found in [Cazals and Giesen 2006; Dey 2007]. The scale-space meshing method in [Digne et al. 2011] proposes to filter the sample point cloud first via mean curvature motion and then interpolate the filtered subset of the original points. Due to interpolation, these methods likely produce jaggy surfaces in the presence of noise. Post-processes are needed to improve the smoothness. If the input points are non-uniformly distributed or incomplete, they do not work well either. While our method can be considered as a combinatorial method, it optimizes the location of the mesh vertices and thus is resilient to noise and outliers.

Implicit surface reconstruction methods. Implicit methods construct implicit functions or indicator functions for the underlying surface the point cloud is sampled from, and perform iso-surfacing to generate a mesh. They are more suitable for data that are noisy, incomplete or non-uniformly distributed. [Hoppe et al. 1992] proposes to estimate the tangent plane at each point using some local methods and then computes signed distance field that defines the implicit function. The implicit function can also be defined globally, as the sum of radial basis functions centered at the points [Carr et al. 2001] or piecewise polynomial functions [Ohtake et al. 2005; Nagai et al. 2009]. Voronoi-based variational reconstruction first estimates a tensor field whose principal axes represent the normal vector and then computes an implicit function by solving a generalized eigenvalue problem such that its gradient is aligned with the normal [Alliez et al. 2007]. The Poisson surface reconstruction [Kazhdan et al. 2006] takes as input a set of oriented points and constructs an indicator function as the implicit function, which is solved using a Laplacian system discretized over a multiresolution B-spline basis. In general, implicit methods require the points to be equipped with oriented normals. However, normal estimation itself is not a trivial task in geometry processing and is very sensitive to noise [Mitra et al. 2004]. Moreover, computing a consistent normal orientation requires additional work [Mullen et al. 2010], which turns out to be surprisingly difficult [Hoppe et al. 1992; Mello et al. 2003]. By contrast, the normal information is not compulsory in our reconstruction.

Point cloud consolidation. When the input data contain a considerable level of imperfection, in general the point cloud should be consolidated before reconstruction. Various point cloud denoising or filtering methods have been proposed [Adams et al. 2009; Rosman et al. 2013; Schall et al. 2007]. Deriving a new point set from a given point cloud has also been considered in the context of defining point set surfaces. Well-known approaches include moving least squares (MLS) [Alexa et al. 2003] and extremal surfaces [Amenta and Kil 2004]. To improve the stability and performance of MLS especially in handling sparse sampling and sharp features, algebraic point set surfaces [Guennebaud and Gross 2007], implicit MLS (IMLS) combined with local kernel regression [Öztireli et al. 2009], and non-local point set surfaces [Guillemot et al. 2012] were proposed. To better deal with outliers and delicate surface structures, Lipman et al. [2007] develop an effective, parameterization-free projection operator (LOP). Huang et al. [2009] improve the LOP method with a more reliable one called weighted locally optimal projection operator (WLOP). More recent work includes l_1-sparse reconstruction of sharp point set surfaces which can recover orientation and positions of highly noisy point sets with sharp features [Avron et al. 2010], edge-aware point set resampling which can reliably resample point data that contain noise and outliers, and are incomplete and unevenly sampled [Huang et al. 2013], and continuous WLOP which can achieve interactive frame rates for moderately sized point clouds [Guillemot et al. 2012]. In our proposed mesh reconstruction, the imperfect input data points are automatically handled in the dictionary learning based $l_{2,q}$-optimization and there is no need to introduce a preprocess to consolidate them.

Sparse dictionary learning. A dictionary is a collection of elements that capture the signal structure and span the signal space. A signal can be represented by a linear combination of the dictionary elements. A sparse representation is the one that is dependent of only a few dictionary elements. Sparse representation has proven to be a very powerful tool in computer vision and signal processing, and recently it also achieves success in geometric processing, for example, in [Avron et al. 2010; He and Schaefer 2013; Wang et al. 2014]. Dictionary learning for sparse representation, which uses a dictionary of primitive elements learned from the signal and decomposes the signal into these primitive elements, has become an extremely active area of research in recent years [Wright et al. 2010; Aharon et al. 2006; Elad and Aharon 2006]. The process basically involves two steps: learning the dictionary and computing (sparse) coefficients for representing the signal using the dictionary elements. Dictionary learning has been applied in many contexts, for example, in signal classification and image de-noising. However, there has been no work on learning dictionaries for surface reconstruction except for [Gal et al. 2007; Sharf et al. 2004] that uses dictionary based learning for surface completion.

3 Formulation

3.1 Surface Reconstruction

Problem. Our problem can be stated as follows: given the point set $\mathbb{P} = \{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n\}$ sampled from a piecewise smooth surface S, we want to find a manifold triangular mesh $M = \{\mathbb{V}, \mathbb{F}\}$ with vertex set $\mathbb{V} = \{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m\}$ and triangle set $\mathbb{F} = \{f(\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k) | \mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k \in \mathbb{V}, i \neq j, i \neq k, j \neq k\}$ to approximate the underlying surface S so that the approximation error is as small as possible.

Approximation error. As S is unknown, it is impossible to measure the approximation error of mesh M to surface S directly. Considering the fact that point set $\mathbb{P}$ is sampled from S, point set $\mathbb{P}$ can be used to replace surface S to estimate the approximation error $d(S, M)$. We define the distance between point $\mathbf{p}_i$ and mesh M as $d(\mathbf{p}_i, M) = \min_{f \in \mathbb{F}} d(\mathbf{p}_i, f)$ where $d(\mathbf{p}_i, f)$ is the distance from point $\mathbf{p}_i$ to triangle f formed by three vertices $\mathbf{v}_r, \mathbf{v}_s$ and $\mathbf{v}_t$ from V, i.e.,

$$d(\mathbf{p}_i, f) = \|\mathbf{p}_i - \mathbf{p}'_i\| = \min_{\substack{\alpha+\beta+\gamma=1\\ \alpha,\beta,\gamma\geq 0}} \|\mathbf{p}_i - (\alpha\mathbf{v}_r + \beta\mathbf{v}_s + \gamma\mathbf{v}_t)\| \quad (1)$$

where $\mathbf{p}'_i = \alpha^*\mathbf{v}_r + \beta^*\mathbf{v}_s + \gamma^*\mathbf{v}_t$, with barycentric coordinates $(\alpha^*, \beta^*, \gamma^*)$ with respect to triangle f, is the point on triangle f that is closest to $\mathbf{p}_i$.

3.2 Dictionary Learning Formulation

Sparse representation. For sample point $\mathbf{p}_i$, if $f = f(\mathbf{v}_r, \mathbf{v}_s, \mathbf{v}_t)$ is the triangle closest to it over mesh M, we say $\mathbf{p}_i$ *corresponds to* f, and we can further find a point $\mathbf{p}'_i = \mathbf{b}_i^r\mathbf{v}_r + \mathbf{b}_i^s\mathbf{v}_s + \mathbf{b}_i^t\mathbf{v}_t$ on f, which is the closest point of f to $\mathbf{p}_i$. Denoting $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m] \in \mathbb{R}^{3\times m}$ as the vertex position matrix of mesh M, we can write $\mathbf{p}'_i$ as $\mathbf{p}'_i = \mathbf{V}\mathbf{b}_i$, where $\mathbf{b}_i$ is an $m \times 1$ vector with at most three nonzero elements $\mathbf{b}_i^r, \mathbf{b}_i^s$ and $\mathbf{b}_i^t$ corresponding to vertices $\mathbf{v}_r, \mathbf{v}_s$ and $\mathbf{v}_t$. Denote the displacement from $\mathbf{p}'_i$ to $\mathbf{p}_i$ by $\mathbf{z}_i = \mathbf{p}_i - \mathbf{p}'_i$. Then $\mathbf{z}_i = \mathbf{p}_i - \mathbf{V}\mathbf{b}_i$.

Let $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n] \in \mathbb{R}^{3\times n}$, $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n] \in \mathbb{R}^{m\times n}$ and $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n] \in \mathbb{R}^{3\times n}$. We have $\mathbf{P} = \mathbf{V}\mathbf{B} + \mathbf{Z}$ with a sparse coefficient matrix $\mathbf{B}$ whose column vector contains at most three non-zero elements. This gives a sparse representation of the point cloud using the vertices in $\mathbf{V}$. From the dictionary learning perspective [Aharon et al. 2006], $\mathbf{P}$ is the given signal, $\mathbf{V}$ is the dictionary, $\mathbf{B}$ is the sparse coding matrix and $\mathbf{Z}$ is the *global approximation error matrix*. As shown in Fig. 2, all the points sampled from the region approximated by a triangle can be represented as a convex combination of the same three vertices.

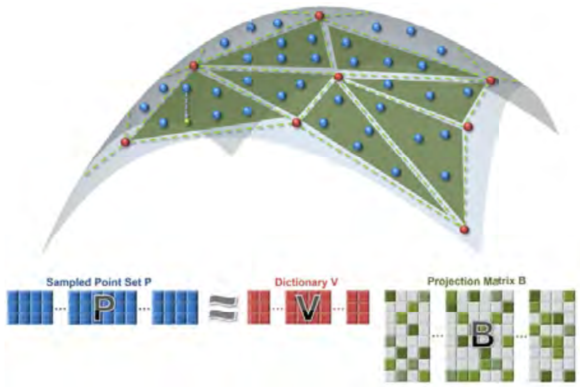


Figure 2: *(Top) An illustration of the reconstruction problem. Given point set $\mathbb{P}$ (blue) sampled from surface S, we approximate S with piecewise linear surface M with vertices $\mathbb{V}$ (red) and triangles $\mathbb{F}$. (Bottom) The reconstruction problem is formulated as a dictionary learning problem where* $\mathbf{P}$ *is the position of sample point set,* $\mathbf{V}$ *is the dictionary and* $\mathbf{B}$ *(green) is the sparse coding matrix that encodes triangles $\mathbb{F}$.*

$l_{2,q}$-optimization. Given sample point position matrix $\mathbf{P}$, surface reconstruction is to find the vertices and connectivity of the reconstructed mesh M best fitting the sample points, which corresponds to finding dictionary $\mathbf{V}$ and sparse coding matrix $\mathbf{B}$. Therefore we formulate the surface reconstruction as a *dictionary learning problem*, which is achieved by performing a constrained minimization over $\mathbf{V}$ and $\mathbf{B}$:

$$\begin{aligned} \min_{\mathbf{V},\,\mathbf{B}} \quad & E = E_{\text{appr}} + E_{\text{reg}} && (2) \\ \text{s.t.} \quad & \|\mathbf{b}_i\|_0 \leq 3, \quad \|\mathbf{b}_i\|_1 = 1, \quad \mathbf{b}_i \geq 0, && \forall i \\ & \mathbf{B} \in \mathbb{MT} \end{aligned}$$

where

$$E_{\text{appr}} = \frac{1}{n}\|\mathbf{Z}\|_{2,q} = \frac{1}{n}\|\mathbf{P} - \mathbf{V}\mathbf{B}\|_{2,q} = \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{p}_i - \mathbf{V}\mathbf{b}_i\|_2^q \quad (3)$$

measures the distance from the point set to the reconstructed mesh in $l_{2,q}$-norm; E_{reg} is a regularization term that will be described in Section 3.3; $\|\mathbf{b}_i\|_0 \leq 3$ means that the number of nonzero elements in vector $\mathbf{b}_i$ is no greater than 3, encoding the corresponding triangle in $\mathbb{F}$; and $\|\mathbf{b}_i\|_1 = 1$ is the requirement of barycentric coordinates; $\mathbf{b}_i \geq 0$ stands for element-wise nonnegativity; and $\mathbb{MT}$ is a subset of $\mathbb{R}^{m \times n}$, consisting of elements whose columns correspond to triangles that can form a manifold mesh. A mesh is a manifold if each edge of the mesh is incident to only one or two faces and the faces incident to a vertex form a closed or an open fan.

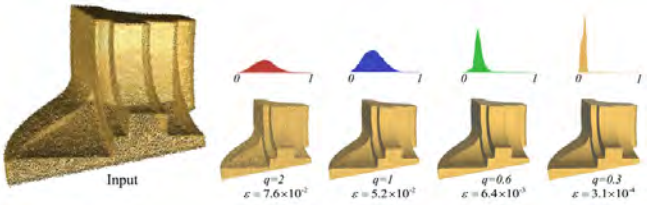


Figure 3: *(Left) The point cloud of the fandisk model is corrupted by noise and outliers. (Right) Reconstruction results of the model with different q values in Eq.* (3). *As the q value decreases, the Hausdorff distance ε between the reconstructed mesh M and the* ***ground truth*** *surface S decreases, and the features get better recovered. The distribution of the reconstruction residuals in the histograms further emphasizes that the residuals would congregate toward zero with smaller q value.*

Remark 1. The objective function in Eq. (3) is specifically designed to be in $l_{2,q}$-norm. To avoid the excessive influence of large residuals caused by noise and outliers contained in the point cloud, we choose a small $q \in (0, 1)$ value. Though the sparse $l_{2,q}$-norm leads to a nonconvex problem, it is known that $\ell_q, q \in (0, 1)$ outperforms ℓ_2 and even ℓ_1 in presence of noise, outlines and sharp features [Avron et al. 2010; Bouaziz et al. 2013]. In addition, by using $l_{2,q}$-norm, we are not solving for simple element-wise sparsity, but instead for group sparsity [Eldar et al. 2010] where all components of a residual vector $\mathbf{z}_i$ vanish simultaneously. As shown in Fig. 3, choosing $q \in (0, 1)$ significantly improves the resilience of the method to large amount of noise and outliers. With the decreasing q value, the reconstruction gets less affected by the noise and outliers contained in the input point cloud and delivers better results.

Remark 2. The sparse coding matrix $\mathbf{B}$ encodes the connectivity information of mesh M. Each column $\mathbf{b}_i$ has at most three nonzero elements $\mathbf{b}_i^r, \mathbf{b}_i^s$ and $\mathbf{b}_i^t$ indicating the sparsity of the dictionary, which also implies that point $\mathbf{p}_i$ *corresponds* to the triangle formed by vertices $\mathbf{v}_r, \mathbf{v}_s$ and $\mathbf{v}_t$. In this way, each sampling point $\mathbf{p}_i$ introduces one triangle, and point set $\mathbb{P}$ introduces at most n triangles as different sample points may correspond to the same triangle. According to Euler-Poincaré formula, the number of triangles in a triangular mesh with m vertices is about $2m$, which suggests that we choose $m \leq n/2$. It is also worth pointing out that our approach is different from classic sparse coding since in our approach sparseness itself is not an optimization objective and the number of basis elements is fixed to be three.

3.3 Regularization

The regularization term E_{reg} is to regularize the reconstructed mesh M to produce good mesh quality.

While the reconstructed mesh should cover the whole region spanned by the sample point set, we would like the mesh vertices to be distributed relatively evenly in space. Observing that over all the triangles of the same area, the one with the shortest perimeter is equilateral, we introduce the following edge length term to the regularization term:

$$E_{\text{edge}} = \frac{1}{l}\sum_{i=1}^{l}\|\mathbf{e}_i\|_2^2, \quad (4)$$

where l is the number of edges and $\mathbf{e}_i$ represents the vector of an edge of mesh M. If the two end vertices of the edge are $\mathbf{v}_s$ and $\mathbf{v}_t$, then $\mathbf{e}_i = \mathbf{v}_t - \mathbf{v}_s$. Thus the edge length term is quadratic in $\mathbf{V}$.

Moreover, in case reliable normals at points are supplied with the point set, such prior information can still be incorporated into our optimization model to guide the reconstruction. Specifically, consider point $\mathbf{p}_i$ with normal vector $\mathbf{n}_{\mathbf{p}_i}$. If $\mathbf{p}_i$ corresponds to triangle f, we align the normal of triangle f with $\mathbf{n}_{\mathbf{p}_i}$ by enforcing $\mathbf{n}_{\mathbf{p}_i} \cdot \mathbf{e}_j = 0$ for three edges $\mathbf{e}_j$ of triangle f. Thus a normal regularization term is defined:

$$E_{\text{normal}} = \frac{1}{3n}\sum_{i=1}^{n}\sum_{\mathbf{e}_j \in f}(\mathbf{n}_{\mathbf{p}_i} \cdot \mathbf{e}_j)^2, \quad (5)$$

which is also quadratic in $\mathbf{V}$.

Overall regularization term. By integrating both terms together, we obtain the overall regularization term: $E_{\text{reg}} = \omega_{\text{e}} E_{\text{reg}} + \omega_{\text{n}} E_{\text{normal}}$, where ω_{e} and ω_{n} are nonnegative tradeoff weights. Here we want to emphasize that the normal regularization is not necessarily required in our formulation, and if the input data do not supply reliable normal information, ω_{n} is simply set to zero. In our experiments, all examples except the surface reconstruction benchmark tests are computed without the normal regularization term.

4 Numerical Optimization

We have converted the problem of constructing mesh M from $\mathbb{P}$ into solving a constrained optimization problem (2), which could be very hard (NP-hard) [Vavasis 2009]. This section describes an efficient alternating direction optimization approach to solve the problem, which involves three steps: initialization, sparse coding and dictionary update. The initialization step is to initialize the dictionary $\mathbf{V}$ and sparse coding matrix $\mathbf{B}$. While sparse coding is to fix $\mathbf{V}$ and update $\mathbf{B}$, dictionary update is to fix $\mathbf{B}$ and update $\mathbf{V}$. This type of alternating directions optimization strategy is quite common and empirically successful in dictionary learning and matrix factorization [Mairal et al. 2010; Jain et al. 2013]. When the iteration converges, we can obtain a triangular mesh M in that dictionary $\mathbf{V}$ represents the positions of vertices and sparse coding matrix $\mathbf{B}$ encodes its triangulation. The whole algorithm is outlined in Alg. 1, followed by the detailed description of the three steps.

Algorithm 1 Dictionary Learning Based Surface Reconstruction.

```
Input:
   Point cloud: $\mathbf{P} = [\mathbf{p}_1, \cdots, \mathbf{p}_n] \in \mathbb{R}^{3\times n}$;
Initialize dictionary $\mathbf{V}$ and sparse coding matrix $\mathbf{B}$ from $\mathbf{P}$;
repeat
   Update matrix $\mathbf{B}$ (Sparse Coding);
   Update $\mathbf{V}$ (Dictionary Update);
until convergence
return Mesh $M(\mathbb{V}, \mathbb{F})$;
```

4.1 Initialization

First, we sample m points from point set $\mathbb{P}$ by applying Poisson-disk Sampling algorithm [Corsini et al. 2012], which gives the initial positions of the reconstructed mesh vertices.

Second, we construct the initial sparse coding matrix $\mathbf{B}$ which gives the connectivity of the mesh. We start with an empty mesh and progressively update it by checking each input point. For point $\mathbf{p}_i$, we select its k-nearest points in $\mathbb{V}$ to construct a triangle set $\mathcal{T}(\mathbf{p}_i)$, which has at most $\binom{k}{3}$ triangles. The triangle to which point $\mathbf{p}_i$ corresponds is chosen from triangle set $\mathcal{T}(\mathbf{p}_i)$ according to the following two criteria and it is then added into the current mesh. Our goal is to find the triangle that helps minimize the objective function of Eq. (2) and updates the current mesh without violating the manifold property. Once the triangle is selected, column vector $\mathbf{b}_i$ is then initialized by solving Eq. (1).

Projection energy. We define a projection energy $E(\mathbf{p}_i, f)$ for point $\mathbf{p}_i$ and triangle $f \in \mathcal{T}(\mathbf{p}_i)$:

$$E(\mathbf{p}_i, f) = E_{\text{appr}}(\mathbf{p}_i, f) + \omega_{\text{e}} E_{\text{edge}}(f) + \omega_{\text{n}} E_{\text{normal}}(\mathbf{p}_i, f) \quad (6)$$

where $E_{\text{appr}}(\mathbf{p}_i, f) = |d(\mathbf{p}_i, f)|^q$, $E_{\text{edge}}(f) = \frac{1}{3}\sum_{e_i \in f} \|\mathbf{e}_i\|_2^2$, and $E_{\text{normal}}(\mathbf{p}_i, f) = \frac{1}{3}\sum_{e_i \in f} (\mathbf{e}_i \cdot \mathbf{n}_{p_i})^2$. Eq. (6) can be considered as a special case of Eq. (2) for point $\mathbf{p}_i$ and triangle f. We compute the projection energy for each triangle in $\mathcal{T}(\mathbf{p}_i)$.

Manifold constraint. Note that in the optimization model of Eq. (2), the manifold constraint $\mathbf{B} \in \mathbb{MT}$ is a hard constraint. Thus we search for the triangle in $\mathcal{T}(\mathbf{p}_i)$ with the smallest energy value and check whether adding it to the current mesh still forms a manifold mesh. If not, we remove the triangle. This process is repeated until there is no triangle left or we meet a triangle that makes the updated mesh still a manifold when it is added to the current mesh. Then this triangle is selected.

It can be seen that this initialization process depends on the order of processing the input points. The generated initial mesh is typically not of in high quality in terms of both geometry and topology. However, it provides an initial manifold mesh. The vertex positions and connectivity will be improved later by the proposed sparse coding and dictionary update which minimize Eq. (2).

4.2 Sparse Coding

In sparse coding, the dictionary $\mathbf{V}$ is fixed and the challenge of the optimization problem lies in identifying which three dictionary elements $\mathbf{v}_r, \mathbf{v}_s, \mathbf{v}_t$ should be chosen for each point $\mathbf{p}_i$ such that the $E_{\text{appr}} + E_{\text{reg}}$ value is the smallest among all possible dictionary element combinations. Once the three optimal elements are determined, the coefficient update becomes a simple least squares problem (Eq. (1)). A greedy strategy like matching pursuit [Mallat and Zhang 1993] can be used to identify the optimal dictionary elements. However, it requires $\mathcal{O}(m)$ operations per sample point, and $\mathcal{O}(mn)$ operations for updating all the coefficients. Here, we propose an edge-based sparse coding method that is found to be very efficient in practice.

***Edge-based* sparse coding** Since we have already had a manifold mesh after the initialization, the basic idea here is to update connectivity of the current mesh rather than start from scratch. The updating process basically involves edge swap that adjusts the connectivity of four vertices to improve the triangulation (i.e., minimize the objective function in our problem), edge insertion in the boundary that will cause insertion of triangles into the mesh and thus extend the mesh so as to probably fill in an unwanted hole or gap, and triangle removal in the region where the surface does have a hole. While these are local operations, we use a recursive strategy to achieve a certain level of global effects. The first two operations are based on edges, which are outlined in Alg. 3. The whole sparse coding algorithm is to iterate the edge-based operations and after the edge-based update is done, the third process of triangle removal is performed, as illustrated in Alg. 2.

Specifically, to implement the edge-based sparse coding, a priority queue Q (in the descending order of $E(\mathbf{e}_i)$) with paired elements $(\mathbf{e}_i, E(\mathbf{e}_i))$ is initialized by adding all the edges from M. $E(\mathbf{e}_i)$ is defined as the sum of the projection energy of all sample points corresponding to the triangles containing edge e_i:

$$E(\mathbf{e}_i) = \sum_{j=1}^{s} \sum_{k=1}^{n_j} E(\mathbf{p}_k, f_j), \quad (7)$$

where s is the number of triangles that contain $\mathbf{e}_i$ and n_j is the number of sample points which correspond to triangle f_j.

In each iteration, the first element $(\mathbf{e}_i, E(\mathbf{e}_i))$ of queue $\mathbb{Q}$ that has the largest projection energy and is likely to be updated is fetched from the set. A recursive edge-update process (Alg. 3) is then applied for $\mathbf{e}_i$ according to its edge type:

- $\mathbf{e}_i$ **is an inner edge.** $\mathbf{e}_i$ will be swapped as shown in Fig. 4 if its opposing edge initially does not exist in mesh M and its projection energy $E(\mathbf{e}_i)$ decreases after edge swap operation. The opposing edge of $\mathbf{e}_i$ is the one that would replace $\mathbf{e}_i$ after the edge swap. If initially the opposing edge is already in the mesh, swapping $\mathbf{e}_i$ would result in a non-manifold edge [Dyer et al. 2007], which is thus not allowed.
- $\mathbf{e}_i$ **is a boundary edge.** We join $\mathbf{e}_i$ with its two adjacent boundary edges and obtain two *virtual* triangles (Fig. 5). If the sample points corresponding to the triangle containing $\mathbf{e}_i$ can be changed to corresponding to any of the new *virtual* triangles with less energy, the new triangle with less projection energy is added into mesh M, and the new generated edge is added to set $\mathbb{Q}$.
- If $E(\mathbf{e}_i)$ decreases, the projection energy of corresponding sample points, edges, and triangles are updated, and the neighbor edges of $\mathbf{e}_i$ (the edges except $\mathbf{e}_i$ in triangle f_1 and f_2 in Fig. 4(left) and Fig. 5(left)) are then updated similarly following the descending order of the newly updated projection energy.

It can be seen that these edge-based operations will decrease the objective energy and the local operations are propagated to other areas via the recursive strategy.

When set $\mathbb{Q}$ is unempty, it means that the edges in set $\mathbb{Q}$ have not been updated in this sparse coding stage. The algorithm will con-

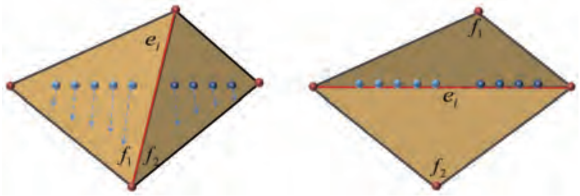


Figure 4: *Inner edge. Left: Inner edge* $\mathbf{e}_i$*(red solid line) with large projection energy* $E(\mathbf{e}_i)$ *before* edge-swap. *Right: projection energy* $E(\mathbf{e}_i)$ *decreases significantly after* edge-swap.

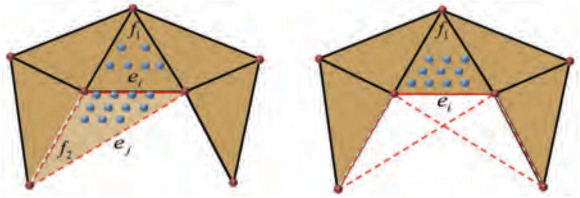


Figure 5: *Boundary edge. Left: triangle* f_2 *is added to* M *if some sample points initially corresponding to triangle* f_1 *are projected to* f_2 *with less projection energy. Right: if no sample point corresponding to triangle* f_1 *is changed to corresponding to any of the two* virtual *triangles, the* virtual *triangles will not be added.*

tinue the edge-based process until set $\mathbb{Q}$ is empty. After that, we perform triangle removal. For those triangles to which no sample points correspond, we remove them from the mesh if the removal does not violate the manifold property. These triangles do not affect the objective energy.

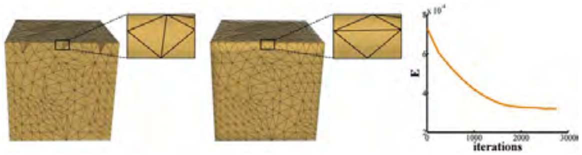


Figure 6: Sparse coding *on a cube model. Left: two-face patch with large projection energy along the edge of cube. Middle: the edge of the cube is recovered accurately with the decreased energy after* edge-swap. *Right: energy monotonously decreases with the increase of the iterations.*

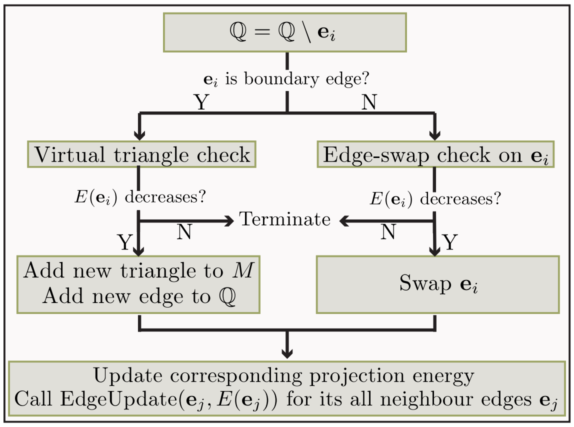
Properties of sparse coding. The proposed edge-based sparse coding algorithm has the following properties:

- **Manifoldness.** The mesh updated by the sparse coding process is still manifold. This is because edge swap and boundary triangle insertion in the sparse coding do not affect the manifold property of the input mesh.
- **Energy decreasing.** The total energy Eq. (2) would monotonously decrease with iterations. This is because the operations of sparse coding decrease the projection energy of sample points corresponding to triangles containing $\mathbf{e}_i$ with the projection energy of other sample points unchanged. As shown in the cube model in Fig. 6, the total energy monotonously decreases with the increase of the iterations, and the triangulation of the mesh after sparse coding gets better.
- **Computational efficiency.** The proposed sparse coding algorithm can efficiently optimize the coefficient weights compared to the matching pursuit method [Mallat and Zhang 1993]. Our algorithm is designed based on the observation that if the projection energy of sample points corresponding to triangle f decreases, the projection energy of these sample points may further decrease if they are projected to the neighboring triangles of f and it is unlikely to decrease the projection energy if they are projected to triangles that are far away from f. As a result, the algorithm tends to have linear running time in practice. For example, we have tested six models on the ratios of the number of EdgeUpdate operations to the edge number under different sampling rates and found that the change of the ratios is quite small, as shown in Fig. 7.

Algorithm 2 SparseCoding($M(\mathbb{V}, \mathbb{F})$).

```
Add $(\mathbf{e}_i, E(\mathbf{e}_i))$ of all edges $\mathbf{e}_i$ from $M$ into queue $\mathbb{Q}$.
while $\mathbb{Q} \neq \emptyset$ do
    Select the element with highest $E(\mathbf{e}_i)$ value from $\mathbb{Q}$.
    Call EdgeUpdate($\mathbf{e}_i$, $E(\mathbf{e}_i)$).
end while
Check and Delete triangles that have no corresponding sample points.
return $M(\mathbb{V}, \mathbb{F})$;
```

Algorithm 3 EdgeUpdate($\mathbf{e}_i$, $E(\mathbf{e}_i)$).


$\mathbb{Q} = \mathbb{Q} \setminus \mathbf{e}_i$
$\mathbf{e}_i$ is boundary edge?
Y
N
Virtual triangle check
Edge-swap check on $\mathbf{e}_i$
$E(\mathbf{e}_i)$ decreases?
$E(\mathbf{e}_i)$ decreases?
Terminate
N
N
Y
Y
Add new triangle to M
Add new edge to $\mathbb{Q}$
Swap $\mathbf{e}_i$
Update corresponding projection energy
Call EdgeUpdate($\mathbf{e}_j, E(\mathbf{e}_j)$) for its all neighbour edges $\mathbf{e}_j$

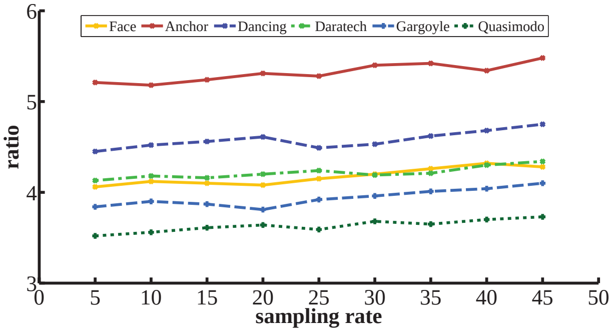


Figure 7: *The ratios of the number of EdgeUpdate operations to the edge number under various sampling rates of* m/n $(5\%, 10\%, \cdots, 45\%)$ *tested on six models.*

4.3 Dictionary Update

Dictionary update is to optimize the vertex positions $\mathbf{V}$ with the connectivity fixed, which is equivalent to solving

$$\min_{\mathbf{V}} \quad E_{\text{appr}} + E_{\text{reg}}. \tag{8}$$

However, it is hard to optimize Eq. (8) directly as the E_{appr} term contains an $\ell_{2,q}$ norm that is non-convex and non-differentiable. Based on variable splitting [Eckstein 1989], we use matrix $\mathbf{Z}$ to replace $\mathbf{P} - \mathbf{VB}$ in term E_{appr}. The dictionary update becomes

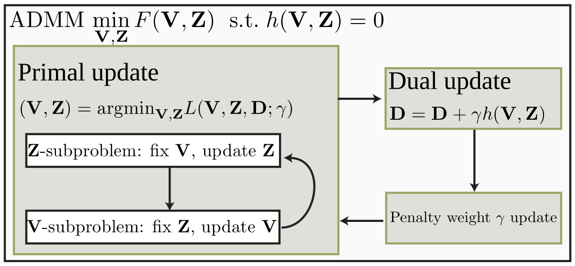
$$\begin{aligned} &\min_{\mathbf{V},\,\mathbf{Z}} \; F(\mathbf{V},\mathbf{Z}) \\ &\;s.t. \;\; h(\mathbf{V},\mathbf{Z}) = 0, \end{aligned} \tag{9}$$

where $F(\mathbf{V},\mathbf{Z}) = E_{\text{appr}} + E_{\text{reg}}$ and $h(\mathbf{V},\mathbf{Z}) = \mathbf{Z} - \mathbf{P} + \mathbf{VB}$. The augmented Langrangian function of Eq. (9) is

$$L(\mathbf{V},\mathbf{Z},\mathbf{D}) = F(\mathbf{V},\mathbf{Z}) + \frac{\gamma}{2}(\|h(\mathbf{V},\mathbf{Z}) + \frac{\mathbf{D}}{\gamma}\|_F^2 - \|\frac{\mathbf{D}}{\gamma}\|_F^2),$$

with the Lagrangian multiplier $\mathbf{D}$. The above problem can be solved by Alternating Direction Method of Multipliers (ADMM) [Boyd et al. 2011], which is efficient for sparse optimization problems with a separable objective function and linear constraints. The algorithm is given in Alg. 4 and the $\mathbf{Z}$-subproblem and $\mathbf{V}$-subproblem are explained in detail below.

Algorithm 4 Dictionary update.


ADMM $\min_{\mathbf{V},\mathbf{Z}} F(\mathbf{V},\mathbf{Z})$ s.t. $h(\mathbf{V},\mathbf{Z}) = 0$
Primal update
$(\mathbf{V},\mathbf{Z}) = \text{argmin}_{\mathbf{V},\mathbf{Z}} L(\mathbf{V},\mathbf{Z},\mathbf{D};\gamma)$
$\mathbf{Z}$-subproblem: fix $\mathbf{V}$, update $\mathbf{Z}$
$\mathbf{V}$-subproblem: fix $\mathbf{Z}$, update $\mathbf{V}$
Dual update
$\mathbf{D} = \mathbf{D} + \gamma h(\mathbf{V},\mathbf{Z})$
Penalty weight γ update

Z-subproblem. The subproblem for $\mathbf{Z}$ is

$$\min_{\mathbf{z}_i} \sum_{i=1}^{n} (\|\mathbf{z}_i\|_2^q + \frac{n\gamma}{2}\|\mathbf{z}_i - \mathbf{x}_i\|_2^2), \tag{10}$$

where $\mathbf{z}_i$ is the i^{th} column of $\mathbf{Z}$ and $\mathbf{x}_i$ is the i^{th} column of $(\mathbf{P} - \mathbf{VB} - \frac{\mathbf{D}}{\gamma})$. As Eq. (10) can be decomposed into n small problems and the optimal $\mathbf{z}_i^*$ is in the form of $\mathbf{z}_i^* = \alpha_i^* \mathbf{x}_i$ (see [Wu et al. 2011]), each sub-problem of Eq. (10) can be simplified into a scalar problem:

$$\min_{\alpha_i} \|\alpha_i \mathbf{x}_i\|_2^q + \frac{n\gamma}{2}\|\alpha_i \mathbf{x}_i - \mathbf{x}_i\|_2^2.$$

The optimal α_i^* can be solved by a few iterations [Marjanovic and Solo 2012].

V-subproblem. The subproblem for $\mathbf{V}$ is

$$\min_{\mathbf{V}} \frac{\omega_{\text{e}}}{l}\sum_{i=1}^{l}\|\mathbf{e}_i\|_2^2 + \frac{\omega_{\text{n}}}{3n}\sum_{i=1}^{n}\sum_{\mathbf{e}_j \in f}(\mathbf{n}_{\mathbf{p}_i}\cdot\mathbf{e}_j)^2 + \frac{\gamma}{2}\|\mathbf{Z} - \mathbf{P} + \mathbf{VB} + \frac{\mathbf{D}}{\gamma}\|_F^2,$$

which is a quadratic problem in $\mathbf{V}$ whose solution can be obtained by solving a linear system. The preconditioned conjugate gradient method with incomplete Cholesky factorization is used in our work to solve the linear system.

In summary, the proposed alternating direction optimization approach iteratively optimizes the triangulation and vertex positions of the reconstructed mesh. Fig. 8 shows an example where the triangulation in the region of the "eye" gets better and better as iterations increase and small holes are filled in the final iterations due to the sparse coding algorithm.

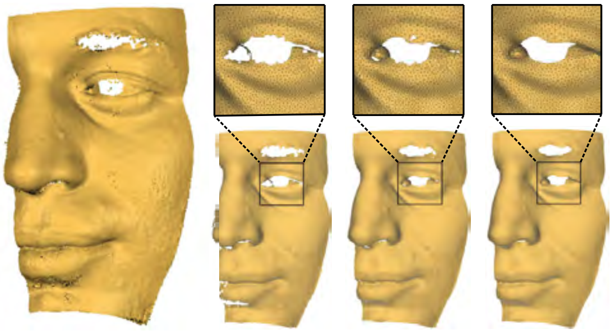


Figure 8: *Both the vertex positions and connectivity are improved (from left to right) as iterations increase.*

5 Experimental Results

In this section, we demonstrate the superior performance of the proposed reconstruction method by comparing it with state-of-the-art methods, including Screened Poisson reconstruction (SP) [Kazhdan and Hoppe 2013], Algebraic Point Set Surfaces (APSS) [Guennebaud and Gross 2007], RIMLS [Öztireli et al. 2009], and Singular Cocone reconstruction(SC) [Dey and Wang 2013]. Note that unless specified, state-of-the-art resampling method called EAR [Huang et al. 2013] is used as a pre-processing to re-sample the input point cloud of size n into a set of points with size m and estimate their corresponding normal information, which are needed for SP, APSS and RIMLS methods. Although the SC method does not require normal information, it stills needs EAR for preprocessing since SC cannot deal with noisy input. By contrast, our method does not need normal information and is robust to noise, and thus we only use the simple Poisson-disk sampling, instead of the advanced and complex EAR algorithm, as the preprocessing for our method to generate an initial dictionary.

For our reconstruction method, we fix the neighborhood size for each point p_i to be 10 for KNN search. Typically, larger m leads to smaller approximation error between the reconstructed mesh M and the ground truth surface S, as shown in Fig. 9. Considering that the input point cloud $\mathbb{P}$ is often over-dense with noise and large m will also increase computational cost, we empirically set $m = 0.4 \times n$, i.e. 40% of the input point cloud set size, for all the tested examples. For parameter q in the $l_{2,q}$ norm in (3), Fig. 3 shows the reconstructed results and the Hausdorff distance ε between the reconstructed mesh M and the *ground truth* surface S. We can see that when q decreases, the reconstructed mesh gets better and the Hausdorff distance ε gets smaller. As smaller q will need more iterations and computation time in Dictioary update and make the problem become more non-convex and prone to local minima, we set $q = 0.3$ in our model to have a good trade-off between performance and computation time. The input point cloud is normalized before processing, and the trade-off parameters ω_{e} and ω_{n} are chosen in the range of $[1, 3]$, and we empirically set $\omega_{\text{e}} = 2.5$ and $\omega_{\text{n}} = 1.6$ for all the experiments. All the tests are run on a

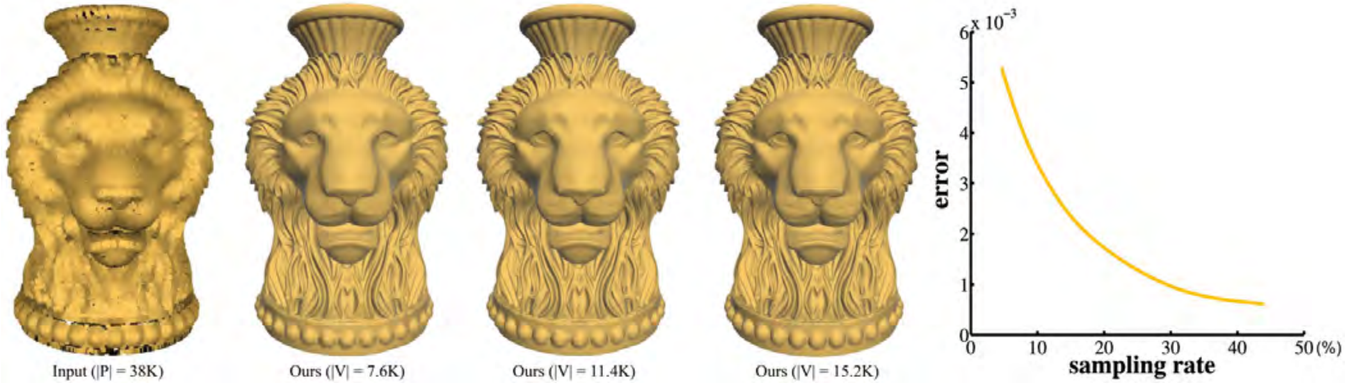


Figure 9: *The visual illustration of our reconstruction with different sampling rates m/n (20%,30%,40%) of the Vase-lion model.*

desktop with a quad-core Intel CPU i5 and 2GB RAM.

5.1 Visual Comparisons

Here, we give visual comparisons of the reconstruction results between our proposed method and state-of-the-art approaches in the following aspects. Note that, similar to other implicit methods, the SP, APSS and RIMLS methods require sufficient grid resolution in order to well preserve surface details. We experimentally find that a resolution of 350^3 provides sufficient resolution for the APSS and RIMLS methods. For the SP method, we empirically set its octree depth to 10, which achieves good reconstruction performance.

Surface detail recovering and sharp feature preserving. As described in Section 3.1, our method is to minimize the distance metric between the input points $\mathbb{P}$ and the reconstructed mesh M, which is better than the conventional explicit approaches that use the distance between $\mathbb{P}$ and the reconstructed mesh vertices $\mathbb{V}$. Besides, in many existing methods, sparsity regularization is applied to the normal difference of adjacent points from the reconstructed mesh or resampled points, which may cause the staircase effects. The $\ell_{2,q}$-norm is applied to the position residues, and thus help avoid the staircase effects. Our unified framework together with the point-to-mesh distance metric and sparse $\ell_{2,q}$ norm usually delivers higher reconstruction accuracy and better preserves surface details and geometric features. As shown in Fig. 1 where the model contains various geometric features such as sharp and semi-sharp features and different levels of surface details, our method can successfully recover them. In Fig. 10, the over-sharpened effect can be observed for SP, which is caused by the normal estimation with the EAR method. In Fig. 11, the use of least square for normal difference in SP smoothes out the sharp features.

Robust to noise & outliers. Fig. 10 shows the reconstruction results with different levels of noise and outliers applied to different regions of the input point cloud. Our method performs well for such a heavily corrupted input, while the SP method over EAR produces over-sharpened edges. Fig. 11 gives another example, where the comparative methods generate over-smoothed geometric details and sharp features, while our method preserves them well. One major reason for the limited performance of the implicit methods is that the preprocessing of the EAR algorithm cannot generate accurate normals at the sharp edges in the presence of severe noise (see the 2nd column of Fig. 11). By contrast, our proposed joint optimization framework does not need a step of normal estimation at all and the $\ell_{2,q}, q \in (0,1)$ norm used in our objective function defined in (3) also makes our method robust to noise and outliers.

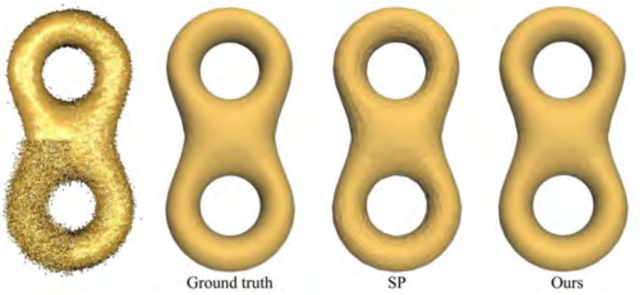


Figure 10: *Left: input point cloud of the double torus model corrupted by four different levels of noise (σ = $0.002, 0.004, 0.006, 0.012$) and outliers at top-left, top-right, bottom-right and bottom-left regions, respectively. Middle: The reconstructed result by the SP method over EAR. Right: The reconstructed result by our proposed method.*

High triangle quality. In our optimization model (2), we introduce the E_{edge} term to encourage vertices on the reconstructed mesh to be uniformly distributed, which leads to better triangle quality. A reconstructed mesh with high triangle quality means more equilateral triangles or more 60^o angles. Fig. 12 gives the angle distributions of the reconstructed Merlion meshes by SC and our methods displayed in Fig. 1. It can be seen that the angle distribution of our reconstructed mesh concentrates around 60^o, better than that of the SC method.

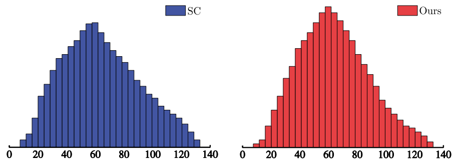


Figure 12: *The angle distributions of the reconstructed Merlion meshes by SC and our methods displayed in Fig. 1.*

Real-world scanned data. We also test the proposed method on real-world data scanned by Kinect and laser scanner, as shown in Fig. 13 and Fig. 14. Even for such challenging data, our method can still reconstruct the surface well. For this type of noisy data, our reconstruction method iteratively improves the reconstruction quality with the increase of the iterations (see the supplementary video for

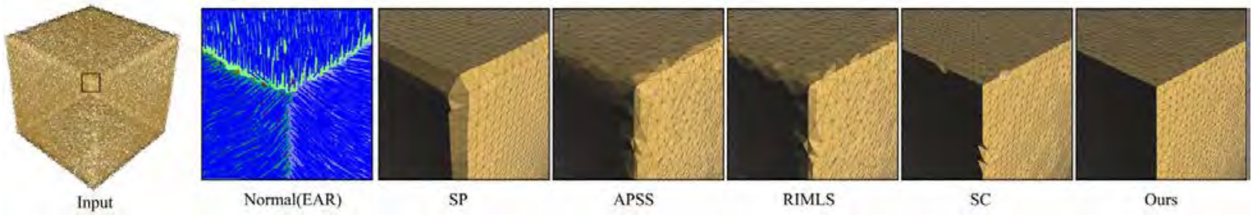


Figure 11: *Zoom-in results of the cube model with 4% noise using different reconstruction methods. The EAR algorithm is used to resample the input point cloud for the four comparative methods. The second column visualizes the normal produced by EAR.*

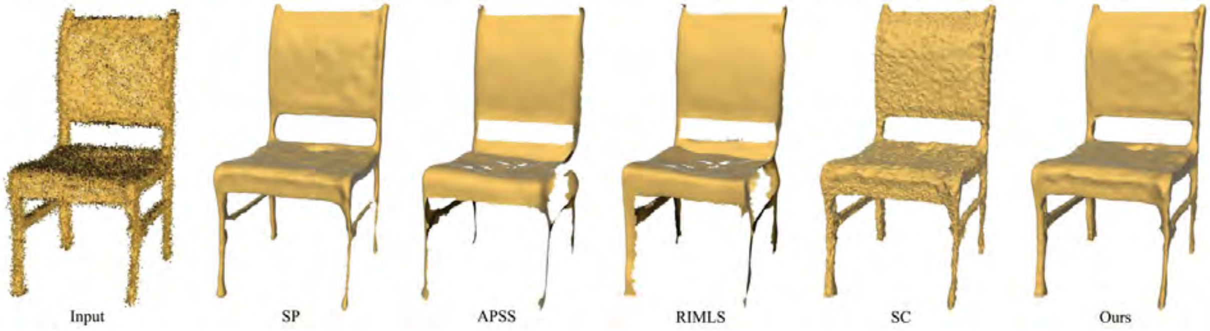


Figure 13: *Various reconstruction results of the chair model. The point cloud is scanned by Kinect. The EAR algorithm is used to resample the noisy input into a clean point set with reliable normals.*

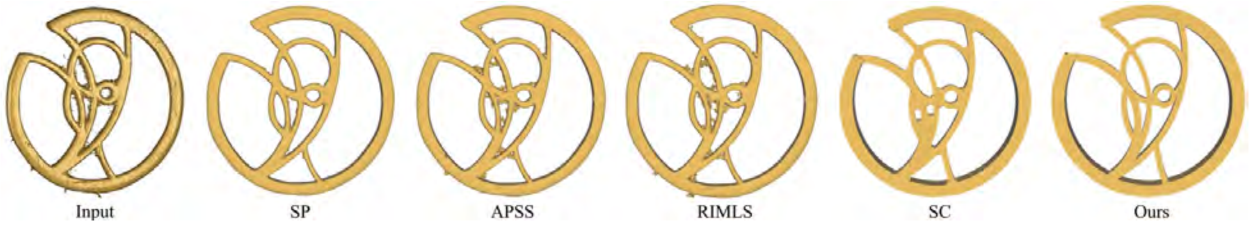


Figure 14: *Various reconstruction results of the FIFA model. The point cloud is scanned by a laser scanner. Note that here we use locally weighted PCA instead of EAR to estimate normals for the comparative methods. This is to demonstrate that the inferior performance of the comparative methods is not due to the inaccurate normal estimation of a particular preprocessing method.*

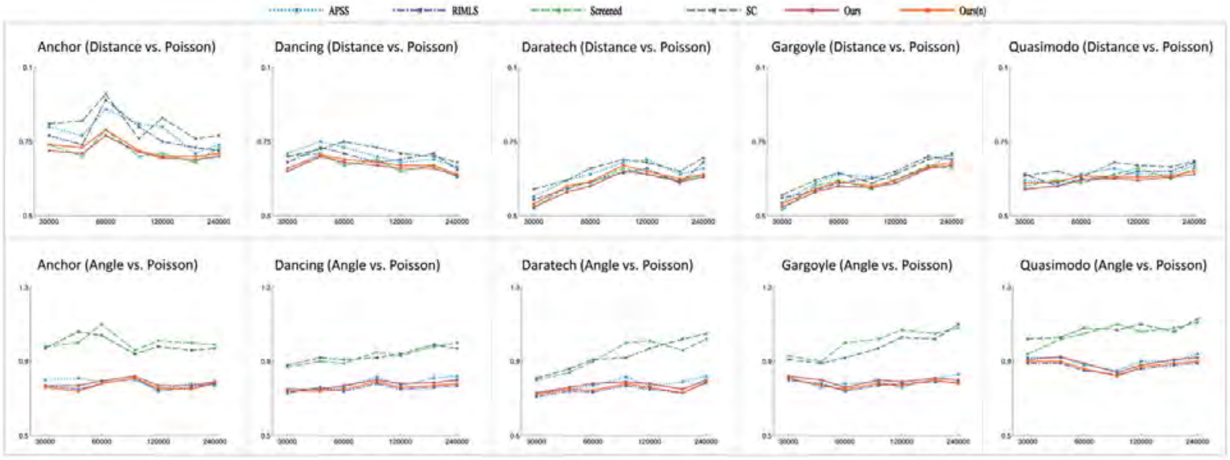


Figure 15: *Surface reconstruction accuracy measured through the benchmark tests. For each of the 5 data sets, the two plots in each column show the mean position distance (top) and the mean normal angle distance (bottom) normalized by the corresponding results of the Poisson reconstruction algorithm. The horizontal axis indicates the number of point samples in the reconstructed surfaces for quality measurement. "Ours" and "Ours(n)" refer to our methods without and with using the normal information provided by the benchmark.*

illustration). Note that in Fig. 13 and Fig. 14, we use different methods for normal estimation to show the inferior performance of the comparative methods is not due to the inaccurate normal estimation of a particular preprocessing method.

5.2 Validation on Benchmark

For quantitative evaluation, we use the surface reconstruction benchmark [Berger et al. 2013], which contains multiple virtual scans of five implicit surface models: Anchor, Dancing Children, Daratech, Gargoyle, and Quasimodo. Different virtual scans simulate different scanner artifacts including nonuniform sampling, noise, and misalignment. The surface reconstruction accuracy is measured via computing the mean position distance and the mean normal angle distance between the ground truth surface and the reconstructed surface. For their mathematical definitions, please refer to the paper [Berger et al. 2013]. The mean position distance reflects the overall closeness between the reconstructed surface and the ground truth while the mean normal angle distance describes the quality of the surface details.

Since the benchmark data sets provide points and the normal information, we do not employ the preprocess of EAR here for the comparative methods, which directly use the provided points and normal information as the input. Fig. 15 shows the reconstruction accuracy results of the benchmark evaluation under different number of point samples in the reconstructed surfaces for quality measurement. Note that all the results are normalized by the corresponding results of the original Poisson reconstruction [Kazhdan et al. 2006]. Thus, values less than one in the figure indicate improved accuracy over Poisson reconstruction, and smaller values represent better performance.

We can see from Fig. 15 that our method, “Ours” - the one without using the normal information provided by the benchmark, achieves overall best mean position distance results and competitive results for mean normal angle distance. If we incorporate the normal information by activating the E_{normal} term in (2), the proposed method, i.e., “Ours(n)”, is able to further improve the mean normal angle distance results at the cost of slightly reducing the position distance performance, as shown in Fig. 15. Fig. 16 illustrates the average reconstruction errors over the 16 noisy registered scans of the benchmark Gargoyle model, where the comparisons among the reconstruction errors by color-coding clearly demonstrate the superior performance of our method. Note that for the benchmark dataset, we have also compared our method with the four comparative methods using EAR for preprocessing and the corresponding quantitative results are provided in the supplementary document, from which similar conclusion can be made.

To evaluate the topology performance, we run the benchmark topology tests on the reconstructed surfaces of different algorithms. Table 1 summarizes the benchmark topology test results in terms of the number of connected components, the total length of boundary components, manifold, and the deviation from the true genus, averaged over all the benchmark point clouds. It can be seen that for the benchmark models SP achieves the best topology test performance and our method is in the second place. However, the SP method is designed for watertight models and it is not suitable for open mesh models. As shown in Table 1, for the non-benchmark Face model with genus three (displayed in Fig. 8), our method significantly outperforms the SP method in the topology tests. Note that for the non-benchmark Face model, since there is no normal information, the EAR algorithm is applied as preprocessing for the four comparative methods. Table 2 gives the running time performance, for which our method is comparable with the other methods. Here the time for normal computation (and denoising) is not included in the running time for other methods. In addition, Table 2 also shows that the number of vertices of the reconstructed meshes by our method is the smallest compared to other methods, which implies that our method produces better reconstruction quality with the same or even lower resolution.

Table 1: *Results of benchmark topology tests on benchmark models and the non-benchmark Face model (displayed in Fig. 8). Here,* comps *refers to the absolute error of the number of connected components,* bndry *is the absolute error of the length of boundary components,* manifold *is whether or not a mesh is manifold,* 1 *being manifold and* 0 *otherwise, and* genus *refers to the amount that deviates from the actual genus. Note that the errors of benchmark models are averaged over the five test models.*

Alg	Benchmark models / Face model			
	comps	bndry	manifold	genus
SP	**0.61**/0.03	**0.32**/11.52	**1.00/1.00**	**0.31**/3.00
APSS	35.43/14.21	141.62/10.63	0.52/0.00	1.82/1.00
RIMLS	32.16/12.86	153.81/8.92	0.73/0.00	1.68/1.00
SC	1.81/0.57	196.73/24.63	**1.00/1.00**	5.34/3.00
Ours	0.78/**0.00**	0.89/**1.46**	**1.00/1.00**	0.63/**0.00**

Table 2: *Running time statistics and the resolution information of the reconstructed results on benchmark models. Here, $|P|$ refers to the number of vertices of the input model an $|V|$ is the number of vertices of the output.*

Model	$\|P\|$	$\|V\|$					Time(seconds)				
		SP	APSS	RIMLS	SC	Ours	SP	APSS	RIMLS	SC	Ours
Anchor	263 k	174 k	360 k	352 k	246 k	**105 k**	**28.54**	31.42	30.61	39.84	32.65
Dancing	468 k	358 k	509 k	511 k	410 k	**187 k**	**38.17**	45.28	41.24	56.18	53.14
Daratech	246 k	181 k	314 k	321 k	219 k	**98 k**	**24.62**	30.14	29.58	36.75	31.08
Gargoyle	481 k	384 k	531 k	529 k	422 k	**193 k**	**42.14**	47.08	46.68	58.49	55.27
Quasimodo	350 k	310 k	365 k	385 k	330 k	**140 k**	**32.56**	39.27	38.19	45.62	44.32

5.3 Limitations

First, our optimization model is nonconvex, which makes it difficult for the solver to theoretically guarantee convergence, and the local minimum is not guaranteed even the algorithm converges. Second, if the point cloud has large holes caused by missing data, our method may fail to fill them since we do not have corresponding sampling points in the hole regions. Fig. 17 gives examples of such failure cases. For these cases, additional hole filling process such as [Sharf et al. 2004] is needed.

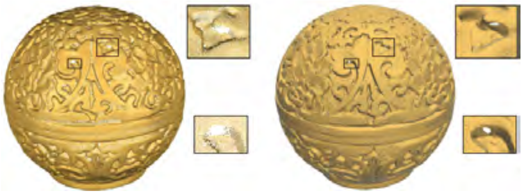


Figure 17: *Examples of the failure cases with holes existing in the reconstruction. Left: input point set of the ball model with zoom-in of the missing data regions. Right: our reconstruction.*

6 Conclusion

We have presented a unified framework using dictionary learning for reconstructing a triangular surface mesh from a dense point cloud. The core of the framework is a constrained $l_{2,q}$-optimization. The algorithm jointly optimizes both geometry and connectivity of the reconstructed mesh in one phase, which overcomes the inherent

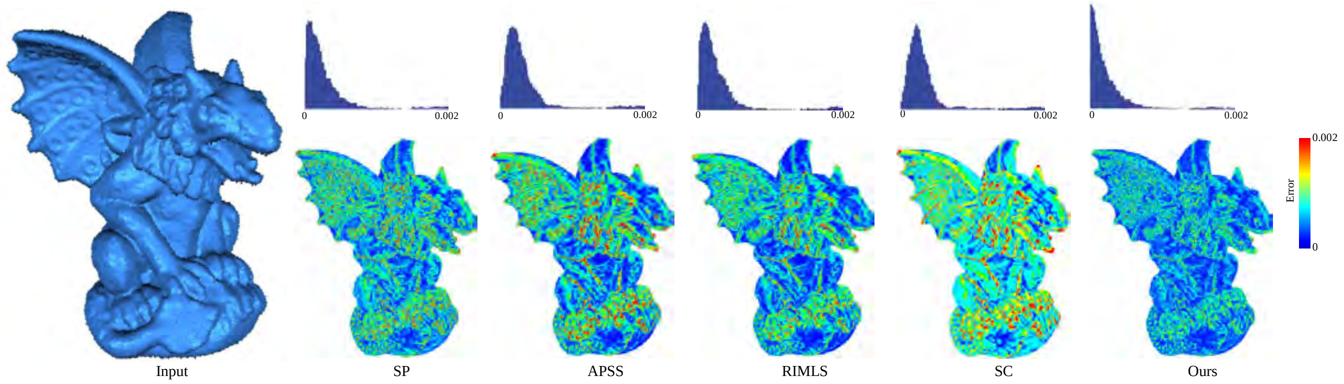


Figure 16: *Reconstruction errors on the noisy scan of the Gargoyle model. The reconstruction errors are measured by the Hausdorff distance between the reconstructed mesh and the ground truth and visualized by the color-coding, where the errors are normalized by the diagonal of the bounding box of the ground truth model. The histograms of the errors are also displayed on the top row, which show that our method results in an error distribution closest to the zero.*

limitations of multi-phase process in the prior art. The presented algorithm has several nice properties: (i) it does not require normal information from point cloud and works for both watertight models and open models in the same way; (ii) it produces high quality reconstruction in terms of both geometric accuracy and mesh connectivity and is thus able to recover various levels of surface details and preserve sharp features; and (iii) it is consistently robust to outliers and noise with different scales.

Acknowledgements

The authors thank Hao (Richard) Zhang, Daniel Cohen-Or and the reviewers for their valuable comments. We also thank Hao Li for his help to prepare the video. This work was supported by the NSF of China (Nos. 61303148, 61222206), NSF of AnHui Province, China (No. 1408085QF119), Specialized Research Fund for the Doctoral Program of Higher Education under contract (No. 20133402120002), One Hundred Talent Project of the Chinese Academy of Sciences, Singapore MoE AcRF Tier-1 Grant RG30/11, Multi-plAtform Game Innovation Centre (MAGIC) and BeingThere Centre funded by the Singapore National Research Foundation under its IDM Futures Funding Initiative and administered by the Interactive & Digital Media Programme Office, Media Development Authority.

References

Adams, A., Gelfand, N., Dolson, J., and Levoy, M. 2009. Gaussian KD-trees for fast high-dimensional filtering. *ACM Trans. Graph. 28*, 3, 21:1–21:12.

Aharon, M., Elad, M., and Bruckstein, A. 2006. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Sig. Proc. 54*, 11, 4311–4322.

Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., and Silva, C. T. 2003. Computing and rendering point set surfaces. *IEEE Trans. Vis. Comput. Graph. 9*, 1, 3–15.

Alliez, P., Cohen-Steiner, D., Tong, Y., and Desbrun, M. 2007. Voronoi-based variational reconstruction of unoriented point sets. In *Symposium on Geometry Processing*, 39–48.

Amenta, N., and Kil, Y. J. 2004. Defining point-set surfaces. *ACM Trans. Graph. 23*, 3, 264–270.

Amenta, N., Choi, S., Dey, T. K., and Leekha, N. 2002. A simple algorithm for homeomorphic surface reconstruction. *Int. J. Comput. Geometry Appl. 12*, 1-2, 125–141.

Avron, H., Sharf, A., Greif, C., and Cohen-Or, D. 2010. l_1-sparse reconstruction of sharp point set surfaces. *ACM Trans. Graph. 29*, 5, 135:1–135:12.

Berger, M., Levine, J. A., Nonato, L. G., Taubin, G., and Silva, C. T. 2013. A benchmark for surface reconstruction. *ACM Trans. Graph. 32*, 2, 20:1–20:17.

Berger, M., Tagliasacchi, A., Seversky, L. M., Alliez, P., Levine, J. A., Sharf, A., and Silva, C. T. 2014. State of the art in surface reconstruction from point clouds. *Eurographics STAR (Proc. of EG'14)*, 161–185.

Bouaziz, S., Tagliasacchi, A., and Pauly, M. 2013. Sparse iterative closest point. *Comput. Graph. Forum 32*, 5, 113–123.

Boyd, S. P., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning 3*, 1, 1–122.

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

Cazals, F., and Giesen, J. 2006. Delaunay triangulation based surface reconstruction: Ideas and algorithms. In *Effective Computational Geometry for Curves and Surfaces*, Springer.

Corsini, M., Cignoni, P., and Scopigno, R. 2012. Efficient and flexible sampling with blue noise properties of triangular meshes. *IEEE Trans. Vis. Comput. Graph. 18*, 6, 914–924.

Dey, T. K., and Giesen, J. 2001. Detecting undersampling in surface reconstruction. In *Symposium on Computational Geometry*, 257–263.

Dey, T. K., and Wang, L. 2013. Voronoi-based feature curves extraction for sampled singular surfaces. *Comput. Graph. 37*, 6 (Oct.), 659–668.

Dey, T. K. 2007. *Curve and Surface Reconstruction: Algorithms with Mathematical Analysis*. Cambridge University Press.

DIGNE, J., MOREL, J., SOUZANI, C., AND LARTIGUE, C. 2011. Scale space meshing of raw data point sets. *Comput. Graph. Forum 30*, 6, 1630–1642.

DYER, R., ZHANG, H., AND MÖLLER, T. 2007. Delaunay mesh construction. In *Symposium on Geometry Processing*, 273–282.

ECKSTEIN, J. 1989. *Splitting methods for monotone operators with applications to parallel optimization*. PhD thesis, MIT.

EDELSBRUNNER, H., AND MÜCKE, E. P. 1994. Three-dimensional alpha shapes. *ACM Trans. Graph. 13*, 1, 43–72.

ELAD, M., AND AHARON, M. 2006. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Img. Proc. 15*, 12, 3736–3745.

ELDAR, Y. C., KUPPINGER, P., AND BÖLCSKEI, H. 2010. Block-sparse signals: uncertainty relations and efficient recovery. *IEEE Trans. Sig. Proc. 58*, 6, 3042–3054.

GAL, R., SHAMIR, A., HASSNER, T., PAULY, M., AND COHEN-OR, D. 2007. Surface reconstruction using local shape priors. In *Symposium on Geometry Processing*, 253–262.

GUENNEBAUD, G., AND GROSS, M. H. 2007. Algebraic point set surfaces. *ACM Trans. Graph. 26*, 3, 23:1–23:10.

GUILLEMOT, T., ALMANSA, A., AND BOUBEKEUR, T. 2012. Non local point set surfaces. In *3DIMPVT*, 324–331.

HE, L., AND SCHAEFER, S. 2013. Mesh denoising via l_0 minimization. *ACM Trans. Graph. 32*, 4, 64:1–64:8.

HOPPE, H., DEROSE, T., DUCHAMP, T., MCDONALD, J. A., AND STUETZLE, W. 1992. Surface reconstruction from unorganized points. In *SIGGRAPH*, 71–78.

HUANG, H., LI, D., ZHANG, H., ASCHER, U. M., AND COHEN-OR, D. 2009. Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. Graph. 28*, 5, 176:1–176:8.

HUANG, H., WU, S., GONG, M., COHEN-OR, D., ASCHER, U., AND ZHANG, H. R. 2013. Edge-aware point set resampling. *ACM Trans. Graph. 32*, 1 (Feb.), 9:1–9:12.

JAIN, P., NETRAPALLI, P., AND SANGHAVI, S. 2013. Low-rank matrix completion using alternating minimization. In *STOC*, 665–674.

KAZHDAN, M. M., AND HOPPE, H. 2013. Screened Poisson surface reconstruction. *ACM Trans. Graph. 32*, 3, 29:1–29:13.

KAZHDAN, M. M., BOLITHO, M., AND HOPPE, H. 2006. Poisson surface reconstruction. In *Symposium on Geometry Processing*, 61–70.

KOLLURI, R. K., SHEWCHUK, J. R., AND O'BRIEN, J. F. 2004. Spectral surface reconstruction from noisy point clouds. In *Symposium on Geometry Processing*, 11–21.

LIPMAN, Y., COHEN-OR, D., LEVIN, D., AND TAL-EZER, H. 2007. Parameterization-free projection for geometry reconstruction. *ACM Trans. Graph. 26*, 3, 22:1–22:6.

MAIRAL, J., BACH, F., PONCE, J., AND SAPIRO, G. 2010. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research 11*, 19–60.

MALLAT, S., AND ZHANG, Z. 1993. Matching pursuits with time-frequency dictionaries. *IEEE Trans. Sig. Proc. 41*, 12, 3397–3415.

MARJANOVIC, G., AND SOLO, V. 2012. On l_q optimization and matrix completion. *IEEE Trans. Sig. Proc. 60*, 11, 5714–5724.

MELLO, V., VELHO, L., AND TAUBIN, G. 2003. Estimating the in/out function of a surface represented by points. In *Symposium on Solid Modeling and Applications*, 108–114.

MITRA, N. J., NGUYEN, A., AND GUIBAS, L. J. 2004. Estimating surface normals in noisy point cloud data. *Int. J. Comput. Geometry Appl. 14*, 4-5, 261–276.

MULLEN, P., DE GOES, F., DESBRUN, M., COHEN-STEINER, D., AND ALLIEZ, P. 2010. Signing the unsigned: Robust surface reconstruction from raw pointsets. *Comput. Graph. Forum 29*, 5, 1733–1741.

NAGAI, Y., OHTAKE, Y., AND SUZUKI, H. 2009. Smoothing of partition of unity implicit surfaces for noise robust surface reconstruction. *Comput. Graph. Forum 28*, 5, 1339–1348.

OHTAKE, Y., BELYAEV, A. G., AND ALEXA, M. 2005. Sparse low-degree implicits with applications to high quality rendering, feature extraction, and smoothing. In *Symposium on Geometry Processing*, 149–158.

ÖZTIRELI, A. C., GUENNEBAUD, G., AND GROSS, M. H. 2009. Feature preserving point set surfaces based on non-linear kernel regression. *Comput. Graph. Forum 28*, 2, 493–501.

ROSMAN, G., DUBROVINA, A., AND KIMMEL, R. 2013. Patch-collaborative spectral point-cloud denoising. *Comput. Graph. Forum 32*, 8, 1–12.

SCHALL, O., BELYAEV, A. G., AND SEIDEL, H.-P. 2007. Feature-preserving non-local denoising of static and time-varying range data. In *Symposium on Solid and Physical Modeling*, 217–222.

SHARF, A., ALEXA, M., AND COHEN-OR, D. 2004. Context-based surface completion. *ACM Trans. Graph. 23*, 3, 878–887.

TOŠIĆ, I., OLSHAUSEN, B. A., AND CULPEPPER, B. J. 2010. Learning sparse representations of depth. Tech. Rep. arXiv:1011.6656, Dec.

VAVASIS, S. A. 2009. On the complexity of nonnegative matrix factorization. *SIAM J. on Optimization 20*, 3, 1364–1377.

WANG, R., YANG, Z., LIU, L., DENG, J., AND CHEN, F. 2014. Decoupling noise and features via weighted l_1-analysis compressed sensing. *ACM Trans. Graph. 33*, 2, 18:1–18:12.

WRIGHT, J., MA, Y., MAIRAL, J., SAPIRO, G., HUANG, T. S., AND YAN, S. 2010. Sparse representation for computer vision and pattern recognition. *Proceedings of IEEE 98*, 6, 1031–1044.

WU, C., ZHANG, J., AND TAI, X.-C. 2011. Augmented Lagrangian method for total variation restoration with non-quadratic fidelity. *Inverse Problems and Imaging 5*, 1, 237–261.