Computer Vision and Image Understanding 169 (2018) 108–118

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Surface reconstruction from unorganized points with l_0 gradient minimization

Huibin Li[a], Yibao Li[1,a], Ruixuan Yu[a], Jian Sun[a], Junseok Kim[*,b]

[a] School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China
[b] Department of Mathematics, Korea University, Seoul, 02841 Republic of Korea

ARTICLE INFO

Keywords:
Surface reconstruction
Point cloud
l_0 gradient minimization
l_0 sparsity
fast Fourier transform

ABSTRACT

To reconstruct surface from unorganized points in three-dimensional Euclidean space, we propose a novel efficient and fast method by using l_0 gradient minimization, which can directly measure the sparsity of a solution and produce sharper surfaces. Therefore, the proposed method is particularly effective for sharpening major edges and removing noise. Unlike the Poisson surface reconstruction approach and its extensions, our method does not depend on the accurate directions of normal vectors of the unorganized points. The resulting algorithm is developed using a half-quadratic splitting method and is based on decoupled iterations that are alternating over a smoothing step realized by a Poisson approach and an edge-preserving step through an optimization formulation. This iterative algorithm is easy to implement. Various tests are presented to demonstrate that our method is robust to point noise, normal noise and data holes, and thus produces good surface reconstruction results.

1. Introduction

Given a set of unorganized three-dimensional (3D) points, the purpose of surface reconstruction is to restore the surface of the original object where the points are scanned. The industry of 3D printing has emerged in the past few years, thus 3D surface reconstruction to visualize objects in space has become one of the main issues in the fields of both applied mathematics (Mancosu et al., 2005) and computer science (Bellocchio et al., 2013). In the past few decades, many algorithms have been developed to solve the surface reconstruction problem (Avron et al., 2010; Berger et al., 2017; Bödis-Szomorü et al., 2017; Calakli and Taubin, 2011; Carlini and Ferretti, 2017; Huang et al., 2009; Kazhdan et al., 2006; Kazhdan and Hoppe, 2013; Khatamian and Arabnia, 2016; Kolluri et al., 2004; Li and Kim, 2015; Li et al., 2014; Lipman et al., 2007; Liu and Wang, 2012; Liu et al., 2016; Ohtake et al., 2005; Reinhold et al., 2014; Xiong et al., 2014; Zagorchev and Goshtasby, 2012; Zhao et al., 2001, 1998). However, it is still a very challenging task due to the missing information of point orders, orientations, connections, as well as complex surface topologies. In general, existing surface reconstruction techniques can be classified into two types: explicit mesh-based reconstruction and implicit volume-based reconstruction.

The explicit mesh-based reconstruction techniques use the unorganized points directly to form a triangular mesh. Kolluri et al. (2004) introduced a noise-resistant method for reconstructing a watertight surface from point cloud data. Lipman et al. (2007) presented a parameterization-free projection algorithm that does not need the local parameters. This approach was extended in Huang et al. (2009) and Reinhold et al. (2014). Xiong et al. (2014) proposed a unified method that treats connectivity construction and geometry as one joint optimization problem. Avron et al. (2010) presented an l_1-sparse approach for reconstruction of point set surfaces with sharp features. All these mesh-based reconstruction schemes are precise, but they have difficulties in dealing with noise, complex topologies, and especially holes in data.

The implicit volume-based reconstruction approaches generally construct an implicit volume-function from the input points, and then obtain the restored surface from the iso-surface of the volume-function. Liu et al. (2016) and Liu and Wang (2012) introduced a method to fit the points with radial basis functions, then the surface is defined as the zero level-set of those radial basis functions. Carlini and Ferretti (2017) proposed a semi-Lagrangian method coupled with radial basis function interpolation for computing a curvature-related level set model. Ohtake et al. (2005) suggested an implicit surface representation using

* Corresponding author.
E-mail addresses: yibaoli@xjtu.edu.cn (Y. Li), cfdkim@korea.ac.kr, tinayoyo@korea.ac.kr (J. Kim).
URLS: http://www.gr.xjtu.edu.cn/web/yibaoli (Y. Li), http://www.math.korea.ac.kr/~cfdkim (J. Kim).
[1] Yibao Li contributed equally to this work with Huibin Li, and is the co-first author for this paper.

https://doi.org/10.1016/j.cviu.2018.01.009
Received 28 March 2017; Received in revised form 9 January 2018; Accepted 24 January 2018
Available online 31 January 2018
1077-3142/ © 2018 Elsevier Inc. All rights reserved.

multi-level partition of a unity. The local shape was approximated by a weighted piecewise quadratic function. Zhao et al. (2001, 1998) proposed the construction of a stopping function that acts to stop the evolution when the contour reaches the surface data points. The main advantage of these methods is that it can easily reconstruct surface with complex topologies, and can also be implemented concisely. A curvature-adaptive implicit surface reconstruction for irregularly spaced points in 3D space was introduced in Zagorchev and Goshtasby (2012). One of the best-known techniques is Poisson surface reconstruction Kazhdan et al. (2006), in which the implicit function is used as the indicator function of the volume bounded by the surface. This function is obtained by solving a Poisson equation and is identically equal to one inside, zero outside, and discontinuous on the reconstructed surface. The main disadvantage of this method was improved by Kazhdan and Hoppe (2013) adding constraints on the points to avoid over-smoothing of the reconstructed surface. Furthermore, Calakli and Taubin (2011) suggested adding a higher-order regularization term and introduced the Hessian matrix of the indicator function. Recently, we presented a novel fast and accurate phase field model for surface embedding narrow volume reconstruction from an unorganized surface data set. The methods proposed in Li et al. (2014) and Li and Kim (2015) were based on the Allen–Cahn (AC) equation (Allen and Cahn, 1979), which has the motion by the surface mean curvature and can be applied to image processing problems (Li and Kim, 2011, 2012). We choose the AC equation because an accurate and fast hybrid numerical solver is available (Li et al., 2010). The phase field model can be directly used to reconstruct a surface from the point cloud.

All these implicit methods used the l_1- or l_2-norm in their proposed minimization or energy term. However, sometimes they suffer from a tendency to oversmooth the data. To produce sharper surfaces than either the l_1- or l_2-norm, in this paper, we present a novel accurate and fast method by using l_0 gradient minimization, which can directly measure the sparsity of a solution and produce a sharper surface. Therefore, the proposed method is particularly effective for sharpening major edges and removing noise. To the best of the authors' knowledge, there are no existing methods for surface reconstruction using the l_0-norm, which can produce the sparsest solutions. Unlike the Poisson surface reconstruction approach and its extended approaches, our method does not depend on accurate directions of normal vectors of the unorganized points. The resulting algorithm is developed by using a half-quadratic splitting method (He et al., 2014) and is based on decoupled iterations that are alternating over a smoothing step by a Poisson approach and an edge-preserving step using an optimization formulation. This iterative technique is fast, simple, and easy to implement. Various numerical tests are presented to demonstrate that our method is robust to point noise, normal noise, and data holes, and thus produces good surface reconstruction results.

Our paper is organized as follows. In Section 2, the proposed method for surface reconstruction is given. We describe the proposed optimization method in Section 3. In Section 4, experimental results and comparisons are given. We draw the conclusions in Section 5. In Appendix B, we present the numerical solver.

2. Description of the proposed model

The implicit method uses the data set to define a signed distance function on Cartesian grids and the reconstructed surface is defined as the zero iso-surface of the signed distance function. Let us briefly review the definition of the signed distance function. At a point $\mathbf{x}$ in the domain Ω, we denote by $\overline{\mathbf{X}}$ the data point that is closest to $\mathbf{x}$. Then, we define the signed distance function as

$$d(\mathbf{x}) = s(\mathbf{x})\overline{d}(\mathbf{x}). \tag{1}$$

Here $\overline{d}(\mathbf{x})$ is the unsigned distance function, which is defined as $\overline{d}(\mathbf{x}) = \|\mathbf{x} - \overline{\mathbf{X}}\|$ and $s(\mathbf{x})$ is the sign of the signed distance function $d(\mathbf{x})$, which is defined as

$$s(\mathbf{x}) = \text{sign}((\mathbf{x} - \overline{\mathbf{X}})\cdot\mathbf{N}(\overline{\mathbf{X}})). \tag{2}$$

Here, sign() is the sign function, which is defined as 1 or − 1 for positive or for negative arguments, respectively and $\mathbf{N}$ is the outward normal vector at the surface point. The inner product field is defined as the signed distance of a grid point to the tangent plane of the closest surface point. However, even the normal vectors are given with high accuracy, if the vectors $\mathbf{x} - \overline{\mathbf{X}}$ and $\mathbf{N}(\overline{\mathbf{X}})$ are nearly perpendicular, i.e., $(\mathbf{x} - \overline{\mathbf{X}})\cdot\mathbf{N}(\overline{\mathbf{X}}) \approx 0$, or $\mathbf{N}(\overline{\mathbf{X}})$ is with noise, $s(\mathbf{x})$ will be with noise. In practice, there are outliers or conflicting points in an unorganized point cloud. The normal vectors for the point cloud may be absent in practice. However, the direction vectors exist when the points are scanned from a 3D machine. Note that the normal information is not necessary for our proposed method. Although $s(\mathbf{x})$ is with noise, the numerical tests in Section 4 indicate that the proposed method can accurately remove noise and produce good results. Let us consider the following l_0 gradient regularization version:

$$\min_{\phi} \int_{\Omega} g(\mathbf{x}) \, \|\nabla\phi\|_0 \, d\mathbf{x}, \tag{3}$$

where

$$g(\mathbf{x}) = \tanh(\overline{d}(\mathbf{x})/(\sqrt{2}\xi)). \tag{4}$$

Here, ξ is related to the interface transition thickness. $g(\mathbf{x})$ is a weighted function (see Fig. 1(b) and (d)), which is almost zero near the data set and is non-negative in the other regions. The gradient $\nabla\phi(\mathbf{x}) = (\partial_x\phi, \partial_y\phi, \partial_z\phi)$ for each grid $\mathbf{x}$ denotes the vector differential operator along the x-, y-, and z-directions. The l_0-norm of a vector $\nabla\phi$, i.e., $\|\nabla\phi\|_0 = |\partial_x\phi|^0 + |\partial_y\phi|^0 + |\partial_z\phi|^0$, which directly measures the sparsity and enforces the surface to be sharper. Here, we define $0^0 = 0$. The initial surface $\phi^0(\mathbf{x})$ (see Fig. 1(c) and (d)) is chosen as

$$\phi^0(\mathbf{x}) = \tanh(d(\mathbf{x})/(\sqrt{2}\xi)), \tag{5}$$

which is defined by $\phi^0(\mathbf{x}) \approx 1$ in the interior region and $\phi^0(\mathbf{x}) \approx -1$ in the exterior region. The reconstructed surface is defined by $\phi^0(\mathbf{x}) = 0$. The initial surface is usually not bad because most of the given data points are located at the reconstructed surface. Note that our approach does not require all exterior (interior) grid points to be identified correctly. We identify as many correct exterior grid points as possible because a good initial implicit surface can reduce the computational cost significantly.

The following points should be noted:

(1) Since the function $g(\mathbf{x})$ is almost zero near the data set and is non-negative in the other regions, $g(\mathbf{x})$ also acts to stop the evolution when the contour reaches the surface data points.
(2) The l_0-norm minimization for $\nabla\phi(\mathbf{x})$ can produce sharper surface edges than those of either l_1-norm or l_2-norm.
(3) Combining the above two advantages, the reconstructed surface is close to the original point clouds and has sharper surfaces with high quality.

We need to define the value of the function along the domain boundary for Eq. (3). If we assume that the reconstructed surface is away from the boundary of the domain, we can use the Dirichlet boundary condition, $\phi = -1$. Meanwhile, the zero Neumann boundary condition enforces that the normal derivative is zero along the boundary. A periodic boundary condition can also be used because the triply periodic minimal surfaces exist (Li and Guo, 2017; Li et al., 2016; Torquato and Donev, 2004).

3. Optimization of the proposed model

The minimization problem (3) is difficult to optimize due to the

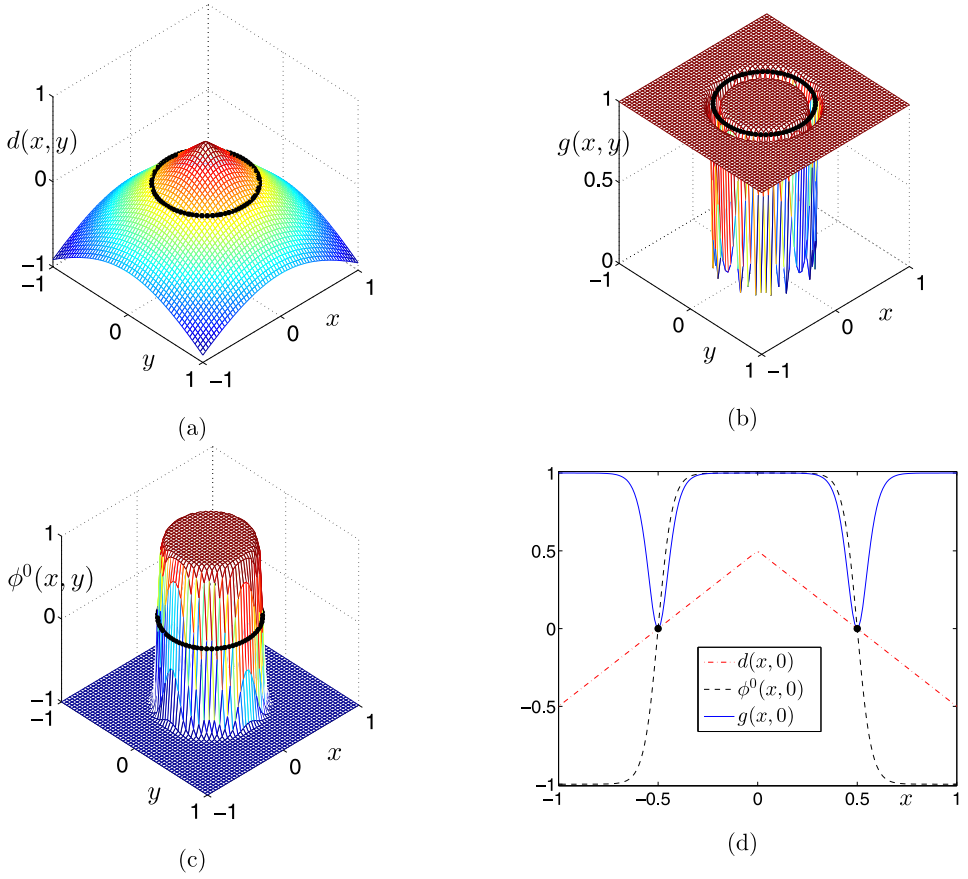

Fig. 1. (a) Signed distance function $d(x, y) = 0.5 - \sqrt{x^2 + y^2}$, (b) the initial order parameter $\phi^0(x, y)$, (c) $g(x, y)$, and (d) the slice plots of $d(x, 0)$, $\phi^0(x, 0)$, and $g(x, 0)$ at $y = 0$.

combinatorial nature of l_0 minimization. Recently, an algorithm for directly optimizing the l_0-norm was proposed in the context of image processing (Xu et al., 2011) and surface denoising (He and Schaefer, 2013). Our work extends the l_0 minimization concept to surface reconstruction. By introducing a set of auxiliary variables $\psi = (\psi_x, \psi_y, \psi_z)$, the minimization problem (3) becomes

$$\min_{\phi,\psi} \int_\Omega [g(\mathbf{x})\|\psi\|_0 + \lambda \|\nabla\phi - \psi\|_2^2] d\mathbf{x}, \tag{6}$$

where λ is a value that controls the significance of $\|\nabla\phi - \psi\|_2$ and $||\cdot||_2$ is the Euclidean norm. The minimization problem (6) generally can be minimized by an alternating minimization method in the following manner:

(1) For fixed ϕ,

$$\min_{\psi} \int_\Omega [g(\mathbf{x})\|\psi\|_0 + \lambda \|\nabla\phi - \psi\|_2^2] d\mathbf{x}. \tag{7}$$

(2) For fixed ψ,

$$\min_{\phi} \int_\Omega \|\nabla\phi - \psi\|_2^2 d\mathbf{x}. \tag{8}$$

To eventually force $\nabla\phi$ to match ψ, both of these optimizations alternate until convergence is reached with increasing λ at each iteration. The idea is summarized as follows.

Step 1. By considering the independence of $\mathbf{x}$, we can rewrite Eq. (7) as

$$\int_\Omega \min_{\psi} [g(\mathbf{x})\|\psi\|_0 + \lambda \|\nabla\phi - \psi\|_2^2] d\mathbf{x}. \tag{9}$$

For each point $\mathbf{x}$, we need to minimize $F(\psi) = g(\mathbf{x})\|\psi\|_0 + \lambda \|\nabla\phi - \psi\|_2^2$. As shown in Appendix A, we can obtain the following condition:

$$\psi = \begin{cases} \nabla\phi & \text{if } g(\mathbf{x}) = 0 \text{ or } \|\nabla\phi\|_2^2 \geq g(\mathbf{x})/\lambda, \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{10}$$

Step 2. The expression (8) is quadratic in ϕ and trivial to minimize. Following the Euler–Lagrange formulation, ϕ minimizes Eq. (8) as

$$\nabla\cdot(\nabla\phi - \psi) = 0. \tag{11}$$

Therefore, to obtain the solution of the minimization problem (3), two optimizations (Eqs. (10) and (11)) alternate until convergence is reached with increasing λ at each iteration.

4. Numerical results

We present computational results using the proposed numerical method on various synthetic and real data sets. To show the basic mechanism of the algorithm, we start with the data of a bunny. The evolution of surface reconstruction is shown in Fig. 2. From left to right, they are the give data set, the numerical results at zero, two, and seven iterations, respectively. Observing these results, we can see that our method performs well in reconstructing the surface. The first reason is that the zero level of the initial condition is much similar with the original surface. The second reason is that each iteration converges very quickly.

We compare our approach with l_1- and l_2-norm minimizations in Eq. (3). The comparisons for the cube with the side length 1 are given in Fig. 3. From left to right, they are the reconstructed surfaces using l_2-norm minimization, l_1-norm minimization, and our proposed l_0-norm minimization, respectively. We can observe that the surfaces can be well reconstructed with these three mentioned norm minimizations.

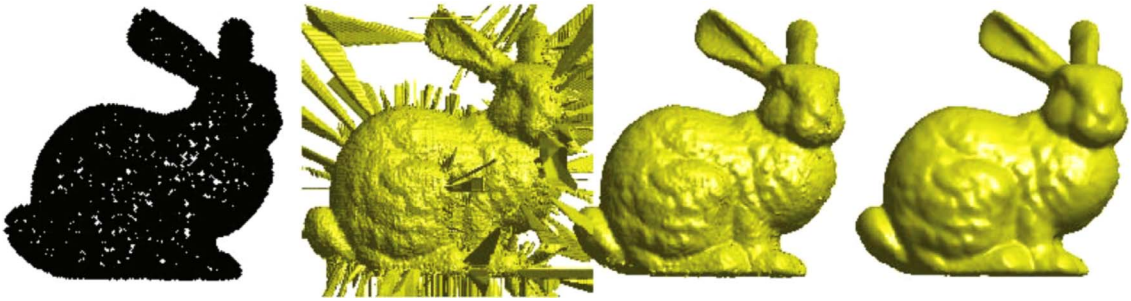

Fig. 2. Evolution of surface reconstruction for the data of a bunny. From left to right, they are the given data set, the numerical results at zero, two, and seven iterations, respectively.

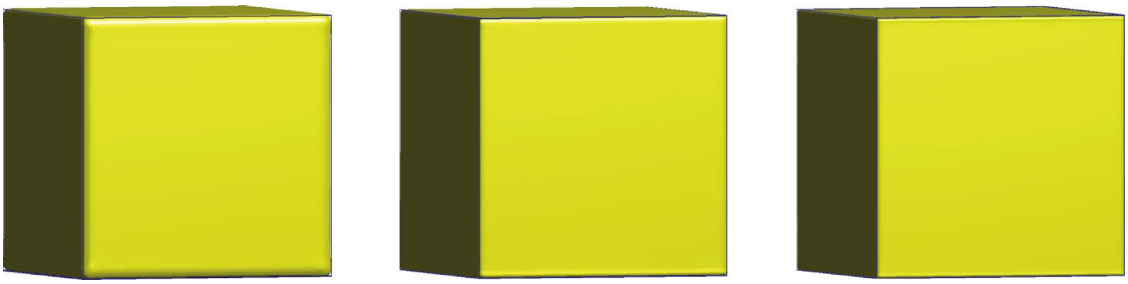

Fig. 3. Recovering sharp features for a cube surface. From left to right, they are reconstructed surfaces using l_2-norm minimization, l_1-norm minimization, and our proposed method, respectively.

However, we consider the error measured using the distances from the obtained points to the reconstructed surface. The errors are $5.756e-3$, $2.243e-3$, and $1.402e-3$, for l_2-norm, l_1-norm, and l_0-norm minimizations, respectively. It can be seen that the result obtained with l_0-norm minimization is quantitatively in good agreement with the ground truth.

Fig. 4(a)–(d) show the ground truth of a sharp sphere surface, and reconstructed surfaces using l_2-norm, l_1-norm, and l_0-norm minimization, respectively. Here a mesh grid $212 \times 212 \times 212$ is used. The methods using l_1 and l_2-norm minimization can reconstruct the surface, but also blur the sharp features. The sharp features are correctly recovered using our proposed approach. Note that the accuracy of our results depends on the mesh grid used. The more mesh grids, the better the reconstructed surface is.

Fig. 5 shows our reconstruction results for the dragon surface with different densities of input data points. From left to right, they are initial input data, reconstructed surfaces based on a coarse grid ($226 \times 164 \times 110$), and reconstructed surfaces based on a fine grid ($411 \times 293 \times 191$). Note that for the purposes of better visualization, we displayed the points more sparsely than the original density. From these results, we can observe that with low density of input data points, our proposed method can still reconstruct the surface with much finer detail and features. As the mesh grid is increased, the scales of the dragon become more sharply pronounced. Meanwhile, the agreement between the results obtained by different sampling densities suggests that our method can successfully reduce the sampling density.

Our method can process the surface reconstruction for the open boundary. Fig. 6 shows a family of the reconstructed surfaces for the triply periodic constant mean curvature surfaces. The left two figures are the given data points and the right two figures are the reconstructed

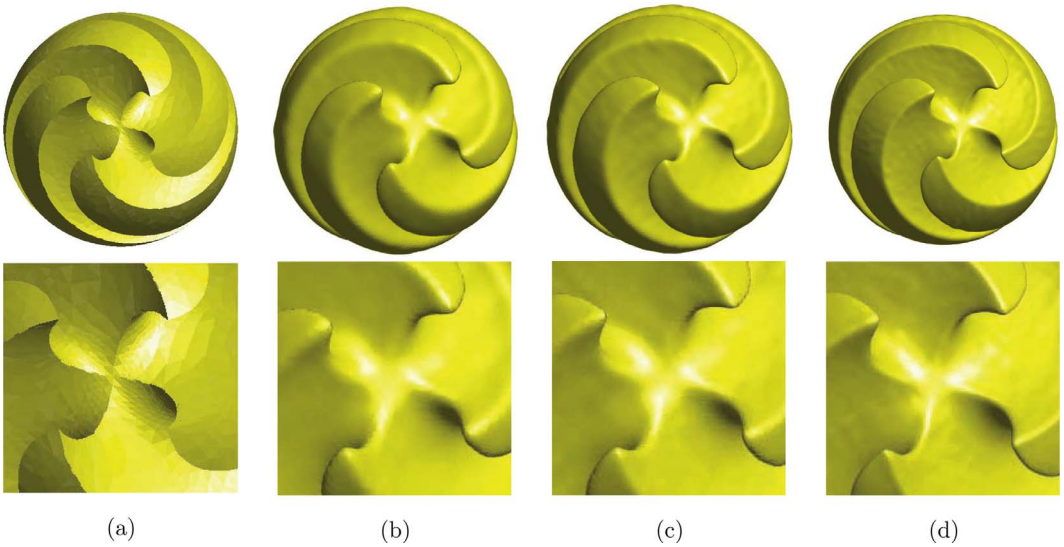

Fig. 4. Recovering sharp features for a sharp sphere surface: (a) ground truth; (b) reconstructed surface using l_2-norm minimization; (c) reconstructed surface using l_1-norm minimization; and (d) reconstructed surface using l_0-norm minimization. Here a mesh grid $212 \times 212 \times 212$ is used.

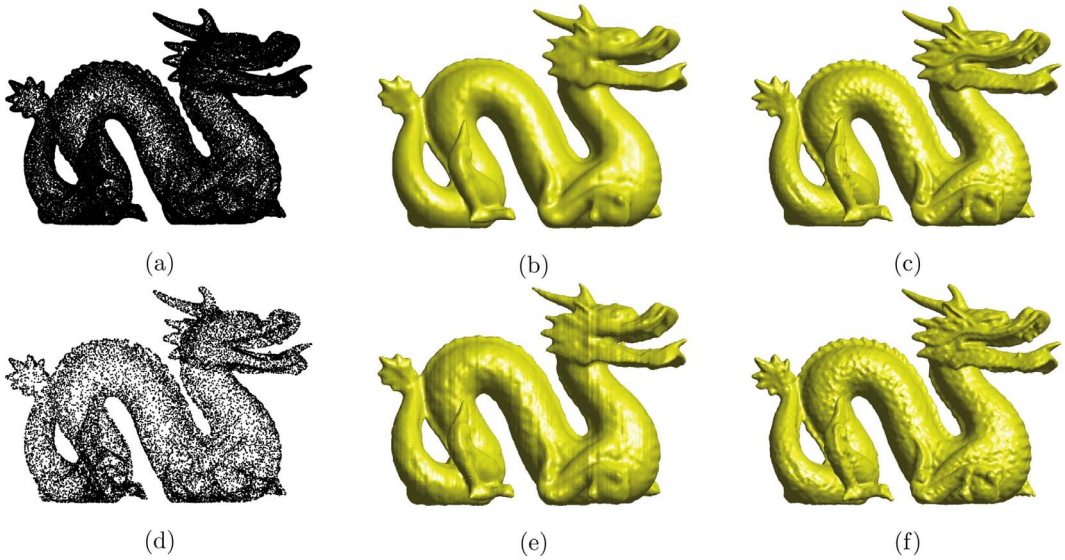

Fig. 5. Surface reconstructions generated with different sampling densities: (a) and (d) initial input data; (b) and (e) reconstructed surfaces based on a coarse grid (226 × 164 × 110); (c) and (f) reconstructed surfaces based on a fine grid (416 × 298 × 196). Note that for the purposes of better visualization, the points are displayed more sparsely than the real density.

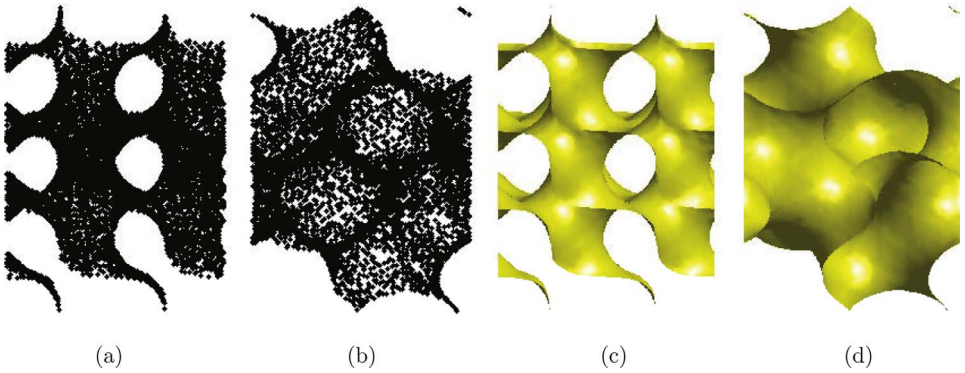

Fig. 6. Surface reconstruction for an open surface: (a) and (b) the given data sets; (c) and (d) the reconstructed surfaces from (a) and (b), respectively.

surfaces, respectively. These simulations demonstrate that our proposed method can perform well for open surface reconstruction.

Fig. 7 shows the surface reconstructions with different boundary conditions. From left to right, the first two results are with Dirichlet boundary condition having a value of -1 along the boundary. The second two results are with periodic boundary conditions. In Fig. 7(a) and (c), we use $l = 5$. In the last two results, we use $l = 1$. Observing the top figures, we can find that imposing Dirichlet constraints enforce the surfaces to be closed. While using periodic boundary condition allows the surface to extend out to the boundary of the domain. If the original surface is closed, even the surface is much closer to the boundary of the domain, we can obtain a clean surface with Dirichlet boundary conditions. On the other side, due to the periodic boundary condition, the artificial surface is created (see Fig. 7(d)).

The holes often arise in the surface reconstruction. Fig. 8(a)–(c) show surface reconstructions of an angel, bunny, and dragon, respectively. The top figures are the reconstructed surfaces and the bottom figures are the reconstructed surfaces with the data points superimposed. Note that we use Dirichlet boundary condition $\phi = -1$ for Fig. 8(a) and use periodic boundary condition for the other two tests. As can be seen, the holes and missing parts can be filled successfully.

In practice, there are outliers or conflicting points into an unorganized point cloud. In Fig. 9(c) and (d), we show surface reconstructions with 10% and 20% random noise. Fig. 9(a) and (b) are the given data points and the reconstructed surface without noise, respectively. These results indicate that our proposed algorithm can successfully reconstruct surfaces with noise. It should be noted that the reconstructed surface with the higher noise level is different from the original one.

When the noise level is much higher, l_2 minimization may do better than l_1 and l_0 minimizations due to its averaging properties. On the other hand, l_0 minimization is more effective for sharpening major edges than the other minimizations. It is certainly difficult to find a suitable minimization that simultaneously keeps the sharp edges and removes the high noise. Therefore, when the noise level is not high, we suggest to use the l_0-norm minimization.

Unlike Poisson surface reconstruction approach or some extended approaches, our method does not depend on the accuracy of the normal vectors of the oriented points. To show that, we add random noise to the normal vectors of a clean happy buddha point cloud. Fig. 10(a)–(e) are the happy buddha point cloud, the reconstructed surface without noise, and the reconstructed surfaces with 50%, 100%, and 300% random noisy normal vector, respectively. The clear results suggest that our proposed method can perform well for the surface reconstruction. It should be noted that if we first use the tangent plane estimation method (Hoppe et al., 1992) to remove the noisy normal vectors and then put them to the Poisson surface reconstruction approach, a clean happy buddha surface can also be reconstructed. However, in that case, the comparison will be somewhat unfair, because the solution in our approach does not depend on the accuracy of normal vectors.

Next, Table 1 lists the number of data points, the iteration numbers, and the CPU times. The computations are performed in *MATLAB* on a 3.4 GHz PC with 16GB of RAM. The CPU times are measured in seconds. 'Pt size' denotes the number of data points. 'CPU(ini)' is the time

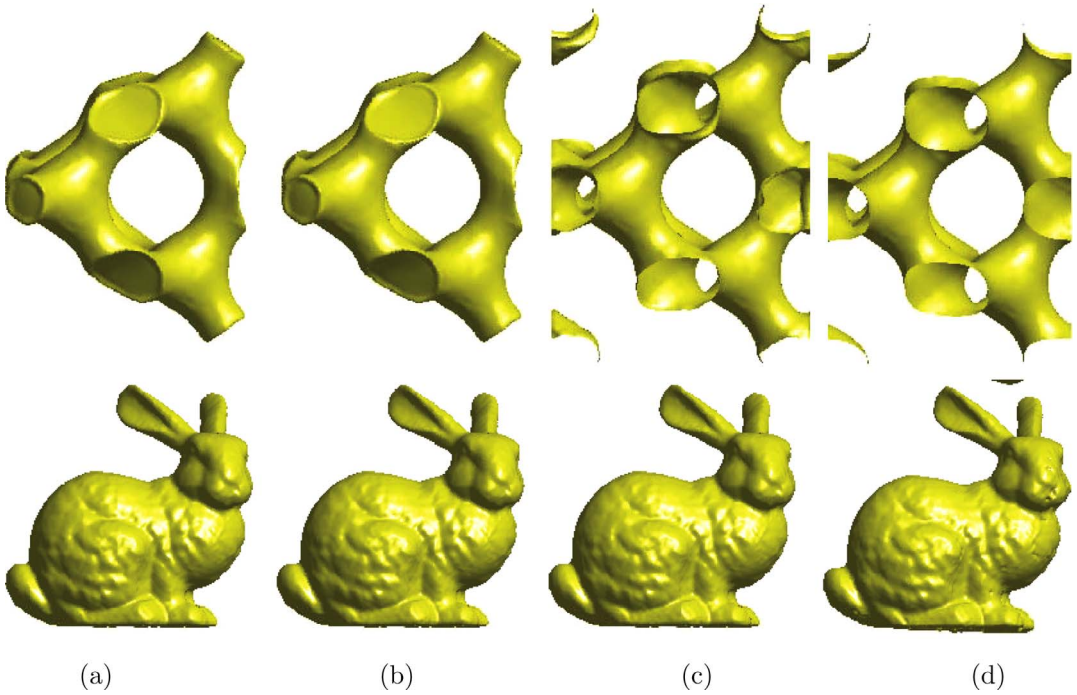

Fig. 7. Surface reconstructions with different boundary conditions. (a) Dirichlet boundary condition having a value of -1 along the boundary with $l = 5$. (b) Dirichlet boundary condition having a value of -1 along the boundary with $l = 1$. (c) Periodic boundary condition with $l = 5$. (d) Periodic boundary condition with $l = 1$.

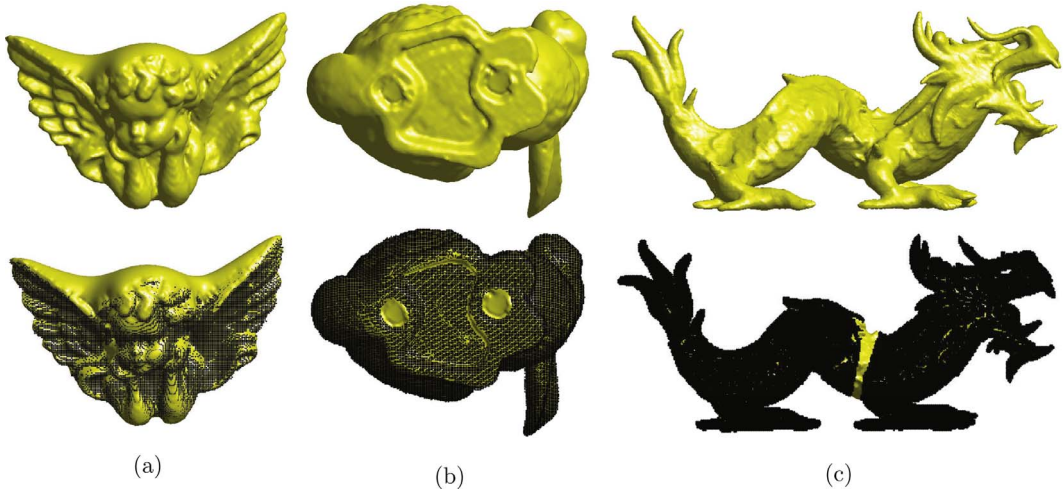

Fig. 8. Hole filling: (a) angel surface; (b) bunny surface; (c) dragon surface. From top to bottom, they are the reconstructed surfaces and the reconstructed surfaces with the data points superimposed, respectively.

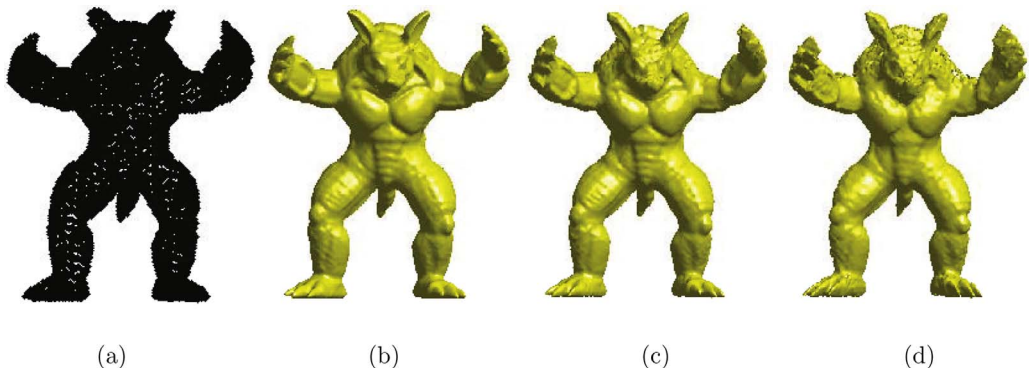

Fig. 9. Surface reconstruction with noisy data: (a) the given data points; (b) the reconstructed surface without noise; (c) the reconstructed surface with 10% noise; and (d)the reconstructed surface with 20% random noise.

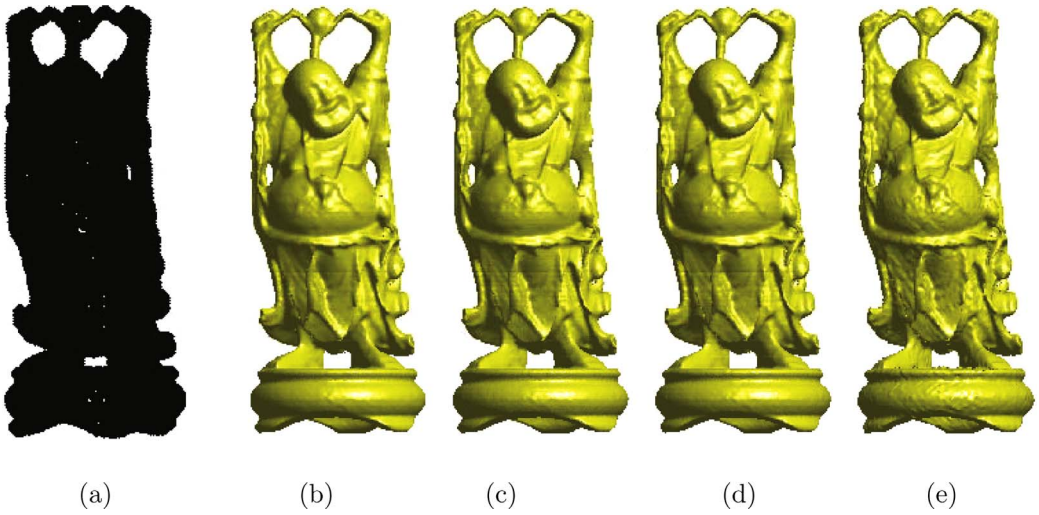

Fig. 10. Surface reconstruction with noisy normal vectors: (a) given data points; (b) reconstructed surface without noisy normal vectors; (c) reconstructed surface with 50% random noisy normal vectors; (d) reconstructed surface with 100% random noisy normal vectors; (e) reconstructed surface with 300% random noisy normal vectors.

Table 1
List of data information, iterations, and CPU times (second). 'Pt size' denotes the number of data points. 'CPU(ini)' is the time for the initial reconstruction. 'CPU(pro)' is the time for processing the surface reconstruction.

Case	Pt size	Grid size	CPU(ini)	CPU(pro)		
				l_0-norm	l_1-norm	l_2-norm
Fig. 2	70068	212 × 210 × 168	1.918	32.950	32.901	4.124
Fig. 3(a)	15302	212 × 212 × 212	3.619	–	–	5.686
Fig. 3(b)	15302	212 × 212 × 212	3.619	–	45.372	–
Fig. 3(c)	15302	212 × 212 × 212	3.619	45.381	–	–
Fig. 4(a)	15302	134 × 142 × 142	2.074	–	–	2.115
Fig. 4(b)	15302	134 × 142 × 142	2.074	–	15.969	–
Fig. 4(c)	15302	134 × 142 × 142	2.074	16.102	–	–
Fig. 5(b)	437645	226 × 164 × 110	1.170	12.277	12.222	1.531
Fig. 5(c)	437645	416 × 298 × 196	4.446	61.106	61.163	7.659
Fig. 5(b)	21883	226 × 164 × 110	1.170	12.277	12.389	1.533
Fig. 5(c)	21883	416 × 298 × 196	4.446	61.106	61.674	7.686
Fig. 6(c)	96240	124 × 124 × 124	0.514	6.583	6.538	0.828
Fig. 6(d)	84600	124 × 124 × 124	0.484	6.583	6.538	0.828
Fig. 8(a)	24566	241 × 177 × 107	0.811	31.122	31.415	3.964
Fig. 8(b)	70068	212 × 210 × 168	1.918	32.952	33.251	4.082
Fig. 8(c)	701891	342 × 166 × 238	3.697	33.228	33.218	4.102
Fig. 9(b)	172974	274 × 322 × 252	4.134	74.194	74.139	9.512
Fig. 9(c)	172974	274 × 322 × 252	4.134	74.194	74.139	9.512
Fig. 9(d)	172974	274 × 322 × 252	4.134	74.194	74.139	9.512
Fig. 10(b)	543652	186 × 422 × 186	3.449	35.989	35.831	4.498
Fig. 10(e)	543652	186 × 422 × 186	3.449	35.989	35.831	4.498
Fig. 10(f)	543652	186 × 422 × 186	3.449	35.989	35.831	4.498

for the initial reconstruction such as simplification of input data and computing the distance function. 'CPU(pro)' is the time for processing the surface reconstruction. Since the complexity of our proposed method is $O(N\log(N))$, our proposed method achieves fast convergence as shown in Table 1.

Compared with CPU times obtained by using the l_0-norm minimization and l_1-norm minimization, l_2-norm minimization results in fast convergence because we can directly optimize the l_2-norm minimization problem (3) in one iteration. However, l_2-norm suffers a tendency to oversmooth the data. Observing Table 1, we also can find the computational costs of l_0- and l_1-norms are similar since we can develop a similar resulting algorithm for l_1-norm minimization by using half-quadratic splitting method.

The l_0-norm in Eq. (3) can be replaced by the l_p-norm ($0 < p < 1$) (Krishnan and Fergus, 2009; Zhu et al., 2015, 2014). Here, we will perform a test to compare the results by using $l_{1/2}$ and $l_{2/3}$ as shown in Fig. 11. Note that we have used a fast alternating method as introduced in Krishnan and Fergus (2009), in which the authors described a way to solve an image deconvolution problem for two cases of $p = 1/2$ and $p = 2/3$. For more details, please refer to Krishnan and Fergus (2009). Observing the results in Fig. 11, we can see that there is no significant difference between the three results. To obtain the exact analytical solutions to the ψ sub-problem, a cubic and quartic polynomials for $p = 1/2$ and $p = 2/3$ are derived. To find the roots of these polynomials, we only can numerically solve these polynomials and should take more care to choose a suitable root. Compared with the l_p-norm, our method can directly find the solution of Eq. (7).

Although our proposed method does not depend on the accuracy of normal vectors and can perform well in surface reconstruction, the accuracy of the reconstructed surface should be improved, specifically oriented point sets point clouds have high-resolution details. In our presented method, a regular grid formulation is used and hence results in high storage requirements. On the other hand, to significantly reduce the computational cost, the authors in Kazhdan et al. (2006); Kazhdan and Hoppe (2013) and Calakli and Taubin (2011) discretized their linear systems by adapting an octree to the location of the data input. Fig. 12 shows an example of this situation for the reconstructed

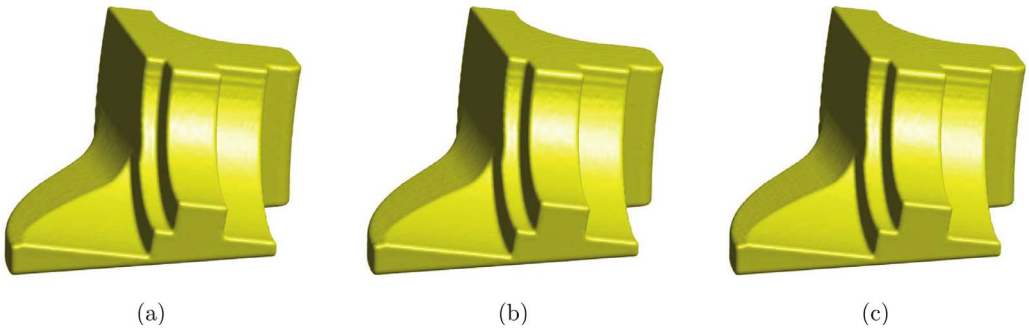

Fig. 11. Comparisons between the results by using different l_p-norms: (a) result using l_0-norm; (b) result using $l_{1/2}$-norm; (c) result using $l_{2/3}$-norm.

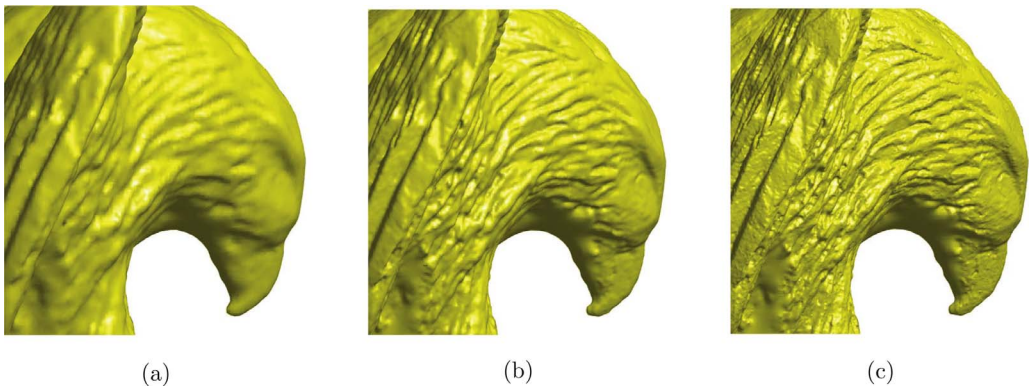

Fig. 12. Surface reconstruction of an eagle model: (a) our proposed method with a 314 × 572 × 438 mesh grid; (b) screened Poisson algorithm (Kazhdan and Hoppe, 2013) with an octree of depth 9; (c) screened Poisson algorithm (Kazhdan and Hoppe, 2013) with an octree of depth 11.

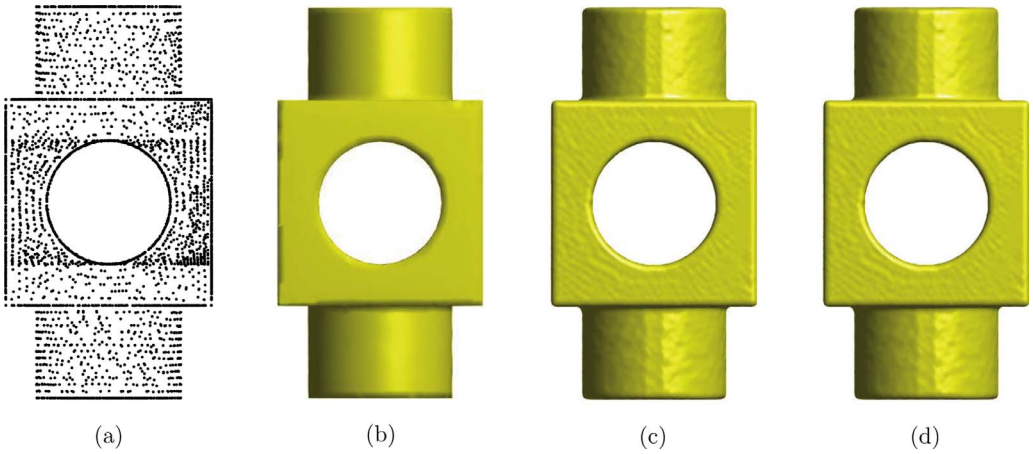

Fig. 13. Surface reconstruction with small input data points: (a) data points; (b) ground truth; (c) reconstructed surface with a 180 × 180 × 322 mesh grid; (d) reconstructed surface with a 234 × 234 × 422 mesh grid.

eagle surface, with the results obtained by our proposed method as shown in Fig. 12(a) and by the screened Poisson algorithm (Kazhdan and Hoppe, 2013) as shown in Fig. 12(b) and (c). A 314 × 572 × 438 mesh grid is used in Fig. 12(a). Spatial octrees of depth 9 and 11 are used for Fig. 12(b) and (c), respectively. Note that an octree of depth 9 and 11 approximately corresponds to 512^3 and 2048^3 regular grids, respectively. Owing to the restriction of stage memory, we reconstruct a surface with more regular grids in our proposed method. As Fig. 12 shows, the screened Poisson algorithm preserves more details as the depth of the octree increases. In future work, we will investigate the current algorithm over an adapted octree and realize it with a GPU. Our algorithm also cannot produce a clean surface when the number of data points is small and the mesh grid of our algorithm is large. In this case, the distance function is so sharp that the l_0-norm cannot distinguish features and our computational results tend to be over-sharpened. Fig. 13(a)–(d) show the input data points, ground truth, reconstructed surface with a 180 × 180 × 322 mesh grid, and reconstructed surface with a 234 × 234 × 422 mesh grid, respectively. As can be seen, even if we refine the mesh grid, the accuracy of the reconstructed surface cannot be improved in this situation.

5. Conclusion

In this article, we have proposed a novel accurate and fast method by using l_0 gradient minimization, which can directly measure the sparsity of a solution and produce sharper surfaces. Therefore, the proposed method is particularly effective for sharpening major edges and removing noise. Unlike the Poisson surface reconstruction approach and its extensions, our method does not depend on accurate directions of normal vectors of the unorganized points. The resulting algorithm has been developed by using decoupled alternating iterations over a smoothing step realized by a Poisson approach and an edge-preserving step through an optimization formulation. This iterative algorithm is fast and simple to implement. Various tests have been presented to demonstrate that our method is efficient at removing point noise, and normal noise, and filling holes, and thus produces good surface reconstruction results. In future work, we will investigate the current algorithm over an adapted octree and realize it with a GPU.

Acknowledgments

Huibin Li is supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 11401464. Yibao Li is supported by National Natural Science Foundation of China under Grant No. 11601416 and No. 11631012. The corresponding author (Junseok Kim) was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2016R1D1A1B03933243). The authors are grateful to the anonymous referees whose valuable suggestions and comments significantly improved the quality of this paper.

Appendix A

First, let us recall the definition of l_0-norm: if $\psi = \mathbf{0}$, then $\|\psi\|_0 = 0$. Otherwise, $\|\psi\|_0 = 1$. For every individual point $\mathbf{x}$, we need to minimize $F(\psi) = g(\mathbf{x})\|\psi\|_0 + \lambda \|\nabla\phi - \psi\|_2^2$, whose minima is F^*. It is equivalent to find a suitable ψ that can minimize $F = g(\mathbf{x})\|\psi\|_0 + \lambda \|\nabla\phi - \psi\|_2^2$. If $g(\mathbf{x}) = 0$, it is obvious that we can obtain the minimum F^* by setting $\psi = \nabla\phi$, otherwise we need to analyze the relationship between $\|\nabla\phi\|_2^2$ and $g(\mathbf{x})/\lambda$. The relations are summarized as follows.

1) When $\|\nabla\phi\|_2^2 \geq g(\mathbf{x})/\lambda$, we start splitting in the following two situations.
In the first situation, we consider $\psi \neq \mathbf{0}$. Then we can obtain

$$\min_{\psi} F(\psi \neq \mathbf{0}) = \min_{\psi}[g(\mathbf{x})\|\psi\|_0 + \lambda\|\nabla\phi - \psi\|_2^2] = g(\mathbf{x}). \tag{12}$$

Here we can obtain the minimal value F^* by setting $\nabla\phi = \psi$. Considering the second situation $\psi = \mathbf{0}$, we can obtain

$$\min_{\psi} F(\psi = \mathbf{0}) = \min_{\psi} \lambda\|\nabla\phi - \psi\|_2^2 = \lambda\,\|\nabla\phi\|_2^2 \geq g(\mathbf{x}) = \min_{\psi} F(\psi \neq \mathbf{0}). \tag{13}$$

Therefore, to minimize F, we should only let $\psi = \nabla\phi$. Combining Eqs. (12) and (13), the minimum energy (7) is produced when $\psi = \nabla\phi$.
2) When $\|\nabla\phi\|_2^2 < g(\mathbf{x})/\lambda$, in a similar way we obtain

$$\min_{\psi} F(\psi \neq \mathbf{0}) = g(\mathbf{x}) \tag{14}$$

and

$$\min_{\psi} F(\psi = \mathbf{0}) = \lambda\,\|\nabla\phi\|_2^2. \tag{15}$$

Combining Eqs. (14) and (15), we can find $\min_{\psi} F(\psi = \mathbf{0})$ is smaller than $\min_{\psi} F(\psi \neq \mathbf{0})$. The minimum $F^* = \lambda\,\|\nabla\phi\|_2^2$ reaches by choosing ψ is $\mathbf{0}$. In summary, to minimize $F(\psi) = g(\mathbf{x})\|\psi\|_0 + \lambda\,\|\nabla\phi - \psi\|_2^2$, we can obtain the following condition:

$$\psi = \begin{cases} \nabla\phi & \text{if } g(\mathbf{x}) = 0 \text{ or } \|\nabla\phi\|_2^2 \geq g(\mathbf{x})/\lambda \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{16}$$

The proof is complete.

Appendix B. Numerical solver

Let Ω be a 3D domain that embeds the given point cloud $X_p = (X_p, Y_p, Z_p)$ for $p = 1, ..., M$. The space step size h is defined as $h = \frac{X_{\max} - X_{\min}}{N_x - 2l} = \frac{Y_{\max} - Y_{\min}}{N_y - 2l} = \frac{Z_{\max} - Z_{\min}}{N_z - 2l}$. Here, $X_{\max}$ and $X_{\min}$ are the maximum and minimum locations of the point cloud along the x-direction. The other parameters are defined in a similar fashion. We use N_x, N_y, and N_z to denote the numbers of cells in the x-, y- and z-directions, respectively and l is a constant number to keep the point cloud away from the boundary of domain Ω. The center of each cell is located at $\mathbf{x}_{ijk} = (x_i, y_j, z_k)$, where $x_i = X_{\min} - lh + (i-1)h$, $y_i = Y_{\min} - lh + (j-1)h$, and $z_k = Z_{\min} - lh + (k-1)h$, for $i = 1, ..., N_x$, $j = 1, ..., N_y$, and $k = 1, ..., N_z$. Let d_{ijk}, g_{ijk}, ϕ_{ijk}, and ψ_{ijk} be the approximations of $d(x_i, y_j, z_k)$, $g(x_i, y_j, z_k)$, $\phi(x_i, y_j, z_k)$, and $\psi(x_i, y_j, z_k)$, respectively. Let ϕ^n and ψ^n be the solutions at the n^{th}-iteration.

At the beginning of each time step, given ϕ^n, ψ^n, and λ^n, we want to find ϕ^{n+1}, ψ^{n+1}, and λ^{n+1} by solving the discretized equations of two optimizations (Eqs. (10) and (11)) in time. The outline of the main procedure in one time step is as follows.

Step 1. Use a linear time method for computing the distance function $d(\mathbf{x})$ (see Calvin et al., 2003) and initialize $d(\mathbf{x})$, $s(\mathbf{x})$, $g(\mathbf{x})$, and $\phi^0(\mathbf{x})$, by using Eqs. (1), (2), (4), and (5), respectively. Here, λ starts from a value λ_0.

Step 2. Solve ψ^{n+1} from ϕ^n and λ^n by using Eq. (10) as:

$$\psi^{n+1} = \begin{cases} \nabla\phi^n & \text{if } g(\mathbf{x}) < \alpha \text{ or } (\|\nabla\phi^n\|_2^2 \geq g(\mathbf{x})/\lambda^n \text{ and } g(\mathbf{x}) \leq \beta), \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{17}$$

Here, α and β are positive constants. In general, α should be small. Combining the relations (10) and (11) together, we can find that if $g(\mathbf{x})$ equals zero, then ψ is given as $\nabla\phi$, which implies that Eq. (11) becomes an identical equation. Therefore, the reconstructed surface remains nearby the input point cloud. On the other hand, if $g(\mathbf{x})$ is not zero and $\|\nabla\phi^n\|_2^2$ is smaller than $g(\mathbf{x})/\lambda^n$, then ψ is set as zero, which implies that Eq. (11) becomes a Poisson equation at $\mathbf{x}$, i.e., $\Delta\phi(\mathbf{x}) = 0$. Solution of the Poisson equation will make the surface be smooth. If the point $\mathbf{x}$ is far from the input point cloud, $g(\mathbf{x})$ is large and its contribution is small in reconstructing the surface. Therefore, to speed up the computation, we assume that the regions far from the input point cloud are smooth. Combining the above reasons, we propose to use Eq. (17) instead of Eq. (10).

Step 3. Solve $\overline{\phi}^{n+1}$ from ψ^{n+1} by using Eq. (11):

$$\overline{\phi}^{n+1} = \mathfrak{F}^{-1}\left(\frac{\mathfrak{F}(\partial_x)^* \circ \mathfrak{F}(\psi_x^{n+1}) + \mathfrak{F}(\partial_y)^* \circ \mathfrak{F}\left(\psi_y^{n+1}\right) + \mathfrak{F}(\partial_z)^* \circ \mathfrak{F}(\psi_z^{n+1})}{\mathfrak{F}(\partial_x)^* \circ \mathfrak{F}(\partial_x) + \mathfrak{F}(\partial_y)^* \circ \mathfrak{F}(\partial_y) + \mathfrak{F}(\partial_z)^* \circ \mathfrak{F}(\partial_z)}\right), \tag{18}$$

where $\mathfrak{F}$ and $\mathfrak{F}^{-1}$ are the fast Fourier transform (FFT) and inverse FFT operators, respectively. The symbols * and $\circ$ denote the complex conjugate and component-wise multiplication, respectively. Note that we have diagonalized derivative operators after FFT for speedup. This yields the above solution.

Step 4. Reinitialize ϕ^{n+1} by the following process:

$$\phi^{n+1} = 2\left(\frac{\overline{\phi}^{n+1} - \overline{\phi}_{\min}^{n+1}}{\overline{\phi}_{\max}^{n+1} - \overline{\phi}_{\min}^{n+1}}\right) - 1. \tag{19}$$

Here, $\overline{\phi}_{\max}^{n+1}$ and $\overline{\phi}_{\min}^{n+1}$ are the maximum and minimum values of $\overline{\phi}^{n+1}$, respectively. From Eq. (11), we can find that if $\overline{\phi}^{n+1}$ is the solution of Eq. (11), then ϕ^{n+1} is also its solution, which implies the solution of Eq. (11) is not unique. Therefore, we should use the processing (19) to project the range of ϕ^{n+1} into $[-1, 1]$.

Step 5. Update $\lambda^{n+1} = \eta\lambda^n$, where η is larger than 1 to make λ increase with each iteration.

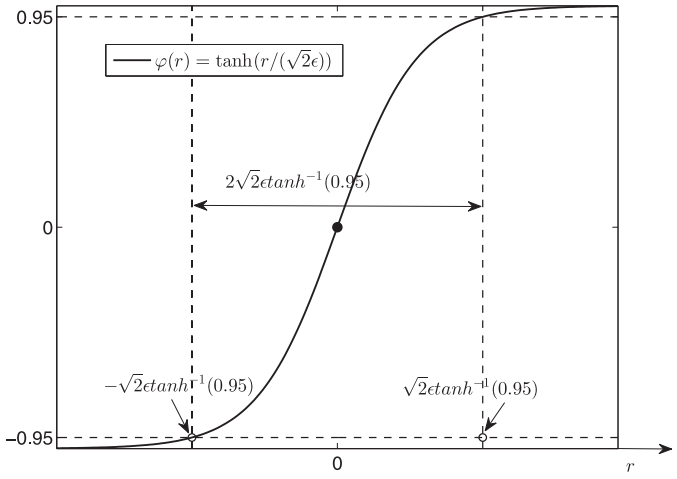

Fig. B.1. Across the interfacial regions, the concentration field varies from − 0.95 to 0.95 over a distance of approximately $2\sqrt{2}\epsilon\tanh^{-1}(0.95)$.

These complete the one time step. This alternating minimization algorithm is effective at speeding up convergence and stops if λ is larger than a given value $\lambda_{\max}$.

Parameter setting

For an equilibrium phase field profile, $\varphi(r) = \tanh(r/(\sqrt{2}\epsilon))$, of Eq. (5), the phase field varies from − 0.95 to 0.95 over $2\sqrt{2}\epsilon\tanh^{-1}(0.95)$. Therefore, if we want this to be approximately $2mh$, i.e., $2mh = 2\sqrt{2}\epsilon\tanh^{-1}(0.95)$, then the ϵ value needs to be taken as $\epsilon_m = mh/[\sqrt{2}\tanh^{-1}(0.95)]$. Here h is the uniform grid size. Furthermore, by denoting $\varphi_k = \varphi(r = kh)$, we can obtain $\varphi_k = \tanh[k\tanh^{-1}(0.95)/m]$. In this paper, we choose $\alpha = \varphi_{0.5}$ and $\beta = \varphi_{0.9m}$, which implies that if the distance between the original point cloud is less than $0.5h$, the surface will be much similar with the original point cloud. If its distance is larger than $0.9mh$, the contribution of that point is considered to be small. Unless otherwise specified, for other parameters, we take $m = 10$, $l = 5$, $\lambda_0 = 10h^2$, $\eta = 2$, and $\lambda_{\max} = 1000h^2$ (Fig. B.1).

Complexity analysis

We use a linear time method (Calvin et al., 2003) to compute the distance function and closest point information. In this algorithm, for each grid, the distance transform assigns a number that is the distance between that grid and the nearest point cloud. Its computational complexity for distance function and nearest point information is $O(M)$. To compute the inner product field, we only need to use the whole grid once. The time for computing the weighted function $g(\mathbf{x})$ for each grid point is $O(N)$, where N is the whole grid size, i.e., $N = N_x N_y N_z$. In our implementation, for each iteration, the complexities for *Steps 2* and *4* in the algorithm are linear with respect to the grid size, i.e., $O(N)$. For *Step 3*, the complexity is $O(N\log(N))$ because the FFT is an efficient $O(N\log(N))$ algorithm. In summary, the complexity of our whole processing is $O(N\log(N))$. Since our initialized surface is close to the correct surface, only a few iterations are required. The total iterations can be approximatively computed as the smallest integer not smaller than $\log_\eta(\lambda_{\max}/\lambda_0)$.

References

Allen, S.M., Cahn, J.W., 1979. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. Acta. Metall. 27, 1085–1095.

Avron, H., Sharf, A., Greif, C., Cohen-Or, D., 2010. l_1-sparse reconstruction of sharp point set surfaces. ACM Trans. Graph 29 (5), 135.

Bellocchio, F., Borghese, N.A., Ferrari, S., Piuri, V., 2013. 3D Surface Reconstruction. Springer, New York.

Berger, M., Tagliasacchi, A., Seversky, L., Alliez, P., Guennebaud, G., Levine, J., Sharf, A., Silva, C., 2017. A survey of surface reconstruction from point clouds. Comput. Graph. Forum 36 (1), 301–329.

Bödis-Szomorü, A., Riemenschneider, H., Goola, L.V., 2017. Efficient edge-aware surface mesh reconstruction for urban scenes. Comput. Vis. Image Understanding 157, 3–24.

Calakli, F., Taubin, G., 2011. SSD: smooth signed distance surface reconstruction. Comput. Graph. Forum 30 (7), 1993–2002.

Calvin, M., Qi, R., Raghavan, V., 2003. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. IEEE Trans. Pattern Anal. Mach. Intell. 2 (25), 265–270.

Carlini, E., Ferretti, R., 2017. A semi-lagrangian scheme with radial basis approximation for surface reconstruction. Comput. Visual. Sci. 18, 103–112.

He, L., Schaefer, S., 2013. Mesh denoising via l0 minimization. ACM Trans. Graph. 32 (4), 64.

He, R., Zheng, W.S., Tan, T., Sun, Z., 2014. Half-quadratic-based iterative minimization for robust sparse representation. IEEE Trans. Pattern Anal. Mach. Intell. 36 (2), 261–275.

Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W., 1992. Surface reconstruction from unorganized points. Comput. Graph. 26, 71–78.

Huang, H., Li, D., Zhang, H., Ascher, U., Cohen-Or, D., 2009. Consolidation of unorganized point clouds for surface reconstruction. ACM Trans. Graph. 28 (5), 176.

Kazhdan, M., Bolitho, M., Hoppe, H., 2006. Poisson surface reconstruction. Proceeding SGP '06 Proceedings of the fourth Eurographics symposium on Geometry processing. Eurographics Association, pp. 61–70.

Kazhdan, M., Hoppe, H., 2013. Screened poisson surface reconstruction. ACM Trans. Graph 32 (3), 29.

Khatamian, A., Arabnia, H.R., 2016. Survey on 3d surface reconstruction. J. Inf. Process. Syst. 12 (3), 338–357.

Kolluri, R.K., Shewchuk, J.R., O'Brien, J.F., James, F., 2004. Spectral surface reconstruction from noisy point clouds. Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing. pp. 11–21.

Krishnan, D., Fergus, R., 2009. Fast image deconvolution using hyper-laplacian priors. Adv. Neural Inf. Process. Syst. 1033–1041.

Li, Y., Guo, S., 2017. Triply periodic minimal surface using a modified Allen–Cahn equation. Appl. Math. Comput. 295, 84–94.

Li, Y., Kim, J., 2011. Multiphase image segmentation using a phase-field model. Comput. Math. Appl. 62, 737–745.

Li, Y., Kim, J., 2012. An unconditionally stable numerical method for bimodal image segmentation. Appl. Math. Comput. 219, 3083–3090.

Li, Y., Kim, J., 2015. Fast and efficient volume reconstruction from scattered data. Pattern Recognit. 48 (12), 4057–4069.

Li, Y., Lee, D., Lee, C., Lee, J., Lee, S., Kim, J., Ahn, S., Kim, J., 2014. Surface embedding narrow volume reconstruction from unorganized points. Comput. Vis. Image Understanding 121, 100–107.

Li, Y., Lee, H.-G., Jeong, D., Kim, J., 2010. An unconditionally stable hybrid numerical method for solving the Allen–Cahn equation. Comput. Math. Appl. 60 (6), 1591–1606.

Li, Y., Lee, H.G., Xia, B., Kim, J., 2016. A compact fourth-order finite difference scheme for the three-dimensional Cahn–Hilliard equation. Comput. Phys. Commun. 200, 108–116.

Lipman, Y., Cohen-Or, D., Levin, D., Tal-Ezer, H., 2007. Parameterization-free projection for geometry reconstruction. ACM Trans. Graph. 26 (3), 22.

Liu, S., Wang, C.C.L., 2012. Quasi-interpolation for surface reconstruction from scattered data with radial basis function. Comput. Aided Geom. Des. 29 (7), 435–447.

Liu, S., Wang, C.C.L., Brunnett, G., Wang, J., 2016. A closed-form formulation of HRBF-based surface reconstruction by approximate solution. Comput. Aided Geom. Des. 78, 147–157.

Mancosu, P., Jørgensen, K.F., Pedersen, S.A., 2005. Visualization, Explanation and Reasoning Styles in Mathematics. Springer Verlag.

Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., Seidel, H., 2005. Multi-level partition of unity implicits. Proceedings of ACM SIGGRAPH. pp. 173.
Reinhold, P., Oliver, M., Murat, A., Renato, P., Michael, W., 2014. Continuous projection for fast l_1 reconstruction. ACM Trans. Graph. 33 (4), 47:1–47:13.
Torquato, S., Donev, A., 2004. Minimal surfaces and multifunctionality. Proc. R. Soc. A Math. Phys. Eng. Sci. 460, 1849–1856.
Xiong, S., Zhang, J., Zheng, J., Cai, J., Liu, L., 2014. Robust surface reconstruction via dictionary learning. ACM Trans. Graph. 33 (6), 201.
Xu, L., Lu, C., Xu, Y., Jia, J., 2011. Image smoothing via l_0 gradient minimization. ACM Trans. Graph. 30 (6), 174.
Zagorchev, L.G., Goshtasby, A.A., 2012. A curvature-adaptive implicit surface reconstruction for irregularly spaced points. IEEE Trans. Vis. Comput. Graph. 18 (9), 1460–1473.
Zhao, H.K., Osher, S., Fedkiw, R., 2001. Fast surface reconstruction using the level set method. Proceedings of IEEE Workshop on Variational and Level Set Methods in Computer Vision (VLSM).
Zhao, H.K., Osher, S., Merriman, B., Kang, M., 1998. Implicit and nonparametric shape reconstruction from unorganized data using a variational level set method. Comput. Vis. Image Underst. 80, 295–319.
Zhu, F., Fan, B., Zhu, X., Wang, Y., Xiang, S., Pan, C., 2015. 10,000+ times accelerated robust subset selection. Proceedings of The Twenty-Ninth AAAI Conference on Artificial Intelligence. pp. 3217–3223.
Zhu, F., Wang, Y., Fan, B., Xiang, S., Meng, G., Pan, C., 2014. Spectral unmixing via data-guided sparsity. IEEE Trans. Image. Process 23 (12), 5412–5427.