

Robust and Efficient Implicit Surface Reconstruction for Point Clouds Based on Convexified Image Segmentation

Jian Liang · Frederick Park · Hongkai Zhao

Received: date / Accepted: date

Abstract We present an implicit surface reconstruction algorithm for point clouds. We view the implicit surface reconstruction as a three dimensional binary image segmentation problem that segments the entire space \mathbb{R}^3 or the computational domain into an interior region and an exterior region while the boundary between these two regions fits the data points properly. The key points with using an image segmentation formulation are: (1) an edge indicator function that gives a sharp indicator of the surface location, and (2) an initial image function that provides a good initial guess of the interior and exterior regions. In this work we propose novel ways to build both functions directly from the point cloud data. We then adopt recent convexified image segmentation models and fast computational algorithms to achieve efficient and robust implicit surface reconstruction for point clouds. We test our methods on various data sets that are noisy, non-uniform, and with holes or with open boundaries. Moreover, comparisons are also made to current state of the art point cloud surface reconstruction techniques.

Keywords principal component analysis (PCA) · distance function · anisotropic Gaussian · edge indicator · normal · image segmentation · total variation

1 Introduction

Point cloud (PC) data is defined simply as a set of unstructured points with no specific ordering and connection. In 2 or 3D, points are defined by their X, Y and X, Y, Z coordinates respectively. PC data is one of the most basic and ubiquitous ways for 3D modeling and rendering in practice, e.g., PC data obtained from 3D scanners or LiDAR (Light Detection And Ranging) measurements. There are several basic problems associated to point clouds including the likes of visualization, segmentation, feature extraction, and surface reconstruction.

This work is partially supported by ARO/MURI grant W911NF-07-1-0185, ONR grant N00014-11-1-0602 and NGA NURI HM1582-10-1-0012.

J. Liang, F. Park, and H. Zhao
Department of Mathematics, University of California Irvine
E-mail: jliang1@math.uci.edu, fepark@math.uci.edu, zhao@math.uci.edu

In this paper, our primary focus will be on surface reconstruction for PC. PC is often highly unstructured without ordering or connection information among data points. Also noise, non-uniform sampling, open boundaries, singularities (corners and edges), and complicated topology further complicate the reconstruction process. For PC with little noise and good sampling rate, the task of surface reconstruction is relatively easy and many efficient algorithms are available. However, in many applications PC data can be quite noisy. Moreover, the sampling rate can be highly non-uniform and there may even be missing pieces such as holes in the data. These issues further exacerbate the intrinsic ill-posedness of the problem. In general, most surface reconstruction methods for PC can be classified into two categories in terms of the form of the representation of the surface, namely explicit or implicit surface. Explicit surface representations prescribe the surface location and geometry in an explicit manner. For example, computational geometry approaches [3, 2, 7, 29, 30] often utilize Voronoi diagrams and Delaunay triangulations to determine connections among data points and subsequently construct triangulated surfaces by connecting adjacent vertices. However, not all Delaunay based methods are explicit. For other explicit representations such as parametric surfaces, we refer the reader to [52, 54]. Implicit surface representations typically embed surfaces as a codimension one level set of a scalar-valued function. For implicit surface representations, the scalar function is usually constructed on a grid [53, 37, 64, 63, 41, 35, 5, 40], which is typically related to the signed distance function, or is constructed grid free using a combination of an implicit surface defined in terms of some mesh-free approximation methods such as a radial basis interpolating function [13, 12, 45, 47, 60]. We also remark that, there are some recent fast and convex optimization techniques for implicit representations in the context of PC, see [61, 36, 35, 65, 17, 5]. Lastly, two works that utilize a graph cut approach for global optimization of PC reconstruction models in the discrete setting are [41, 59].

The advantage of using an explicit surface formulation include precise and concise representation. However, triangulated surface approaches usually have difficulties in dealing with noise and especially holes in data due to the fact that connection relation among data points is based on closest point information, which is local in nature. Parametric surfaces have even more compact and explicit representations, however, the lack of a global parametrization makes it difficult to deal with general data sets. Also explicit surface representations are less robust and less flexible in dealing with complicated topology due to the lack of volumetric information. In addition to the topological flexibility and robustness, we believe that global implicit surface reconstruction with the incorporation of volumetric or prior information can alleviate the ill-posedness and is more appropriate for noisy and highly non-uniform PC. As an example, for the PC in 2D shown in Figure 1 (a), we can see that there are two obvious reasonable reconstructions based on local points configuration, 1 (b) and (c). However, if we incorporate the line of sight (volumetric) information shown in Figure 1 (d), we can resolve the ambiguity easily. Essentially, construction of an implicit surface is equivalent to the segmentation of the domain into two regions, an interior and an exterior region with the boundary approximating the PC. In this work we formulate the implicit surface reconstruction as an image segmentation problem. Different from standard image segmentation where an initial image uniformly sampled on a rectangular grid is given, two key issues for PC are (1) there is no initial image given on a regular grid,

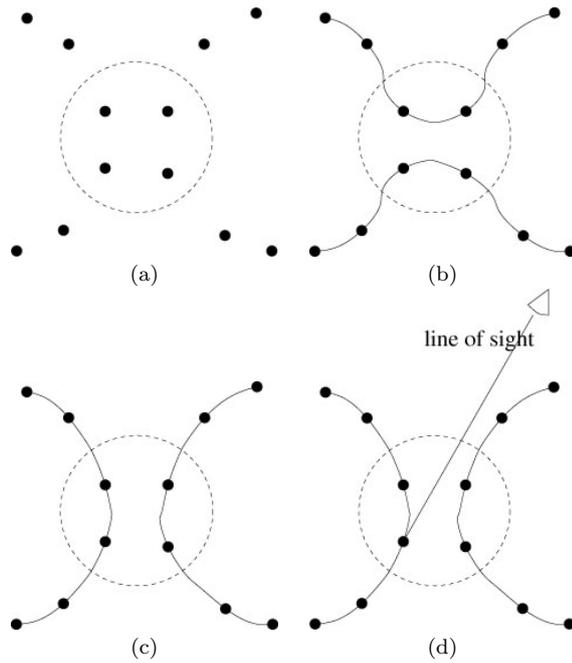


Fig. 1 Ambiguous PC reconstruction and alleviation by utilizing volumetric information.

and (2) a good edge indicator is needed. In this work we develop novel ways to construct this information directly from the scattered PC. In particular, the initial image provides a good indication of inside and outside regions and can incorporate volumetric information, such as line of sight. The edge indicator function is mainly for the purpose of providing a data fitting term that indicates how well the data points are approximated by the boundary between the inside and outside regions. We then adopt recent convexified image segmentation models and corresponding fast algorithms for implicit surface reconstruction. We show both efficiency and robustness of the proposed method to construct water tight surfaces for PC's with complicated topology, noise, and holes.

The paper is outlined as follows, in Section 2 we review some variational models for implicit surface reconstruction and also talk about some binary image segmentation models, both of which motivate our work. The Poisson surface reconstruction [39] will also be mentioned in this section due to its recent popularity as being one of the current state of the art PC reconstruction methods. We will also compare and contrast the Poisson method to the proposed method and also discuss its motivation to our approach. We will then explain our methods for the construction of the initial image and edge indicator, both of which are obtained directly from the PC, in Section 3. Fast algorithms and implementation details will be given in Section 4. Finally test results will be provided in Section 5 to demonstrate the ability of our method in dealing with various kinds of PC data.

2 Related Works

2.1 Surface reconstruction

One of the earliest variational models for implicit surface reconstruction was introduced by Zhao et al. in [64, 63] where the authors propose the following weighted minimal surface model:

$$E(\Gamma) := \left(\int_{\Gamma} d^p(\mathbf{x}) ds \right)^{1/p}. \quad (1)$$

Here, $S = \{\mathbf{x}_i\}_{i=1,2,\dots}$ denotes the data set, and $d(\mathbf{x}) = \text{dist}(\mathbf{x}, S)$ is the distance to the data S . Γ is an arbitrary surface and ds represents an element of surface area. When $p = 1$, the associated Euler-Lagrange equation has the following form:

$$\nabla d(\mathbf{x}) \cdot \mathbf{n} + \frac{1}{p} d(\mathbf{x}) \kappa = 0 \quad (2)$$

where \mathbf{n} denotes the unit normal, and κ is the mean curvature of Γ . The above PDE can be interpreted as a balance between the attraction of the data set $\nabla d(\mathbf{x}) \cdot \mathbf{n}$ and the weighted surface tension $d(\mathbf{x})\kappa$ due to the weighted surface energy. The balance also implies a requirement for local sampling density with respect to the local feature size. The approach proposed in [64, 63] is to start with an initial guess, here, a surface that encloses the data set, and follow the gradient descent of the energy functional, i.e., the initial surface will shrink and wrap around the data points. However, the energy functional (1) is non-convex in Γ . Nonetheless, for discrete data sets, one is looking for a local minimum since the global minimum is $\Gamma = \emptyset$. Since there are many local minima, the initial guess is very important especially if the shape is concave. The distance contour set to some value $\epsilon > 0$ is an approximation to the true shape and can be used as a good initial guess. A fast tagging algorithm is developed to further shrink the initial surface to get approximately the convex hull of the data set without solving a partial differential equation. The method works quite desirably except when the sampling of the data becomes very non-uniform and there are holes as well as concave features that are of comparable or smaller size. The evolving surface can often get stuck in unwanted local minima even if starting very close to the PC data. These issues are also related to using the distance function as an indicator of fitting error. The distance function to a discrete data set is determined totally by the closest point information which is mutually exclusive in the sense that data point information is not shared between neighboring data points. The only information stored at a grid point is simply the distance to the nearest data point. As a consequence, its behavior is isotropic in all directions near a point and does not utilize neighboring points to reveal more local geometric information. Later we will develop a better indicator in terms of both location and geometry for the underlying surface directly from the PC.

In the setting where $p = 1$ in the above minimal surface model (1), the model is equivalent to the geodesic active contour/snake model for image segmentation when the distance function is replaced by an edge indicator function derived from the given image, see [15, 38]. However, these image segmentation models suffer the same intrinsic difficulty of many local minima due to non-convexity of the

optimization problem. Recently there have been a few convexified binary image segmentation models proposed to overcome the above difficulty. There are two key ideas behind these models. First, instead of minimizing over all possible curves or surfaces, which is highly non-convex, the energy function is translated equivalently to the minimization over binary functions which becomes convex (although non strictly convex in general). Second, the fidelity term that incorporates the volumetric image is used in addition to the local edge indicator function. Moreover, efficient and robust optimization algorithms are available for these models. These models will be adopted for our implicit surface reconstruction method which is equivalent to segmenting the whole domain into interior and exterior regions. Here, we give a brief introduction to two of those convex models for binary image segmentation.

2.2 Two convexified image segmentation models

One of the first convexified binary image segmentation models involves the following TVG-L1 model proposed in [10],

$$\min_u \int_{\Omega} g(x)|\nabla u(x)| + \lambda|u - f|dx. \quad (3)$$

In the first term, $g(x) > 0$ is an edge indicator function which is close to zero at edge locations. This term alone is akin to the geodesic active contour model. For example, if $u(x)$ is a characteristic function, then $\int_{\Omega} g(x)|\nabla u(x)|$ is the weighted (by $g(x)$) length/area of the boundary by the co-area formula for total variation (TV) [34,31]. The second term is a volumetric image fitting term in which f is the given image. In particular, when $g(x) = 1$ and $f = 1_D$ is a binary image with D a given region, it was shown in [20] that a global minimizer Σ of the non-convex variational problem (for binary image defined by D):

$$\min_{\Sigma} |\partial\Sigma| + \lambda|\Sigma\Delta D|, \quad (4)$$

where Δ denoting the symmetric differences of the two sets, can be found by computing any global minimizer, $\bar{u}(x)$, which may not be unique, of the convex TV-L1 problem

$$\min_u \int_{\Omega} |\nabla u(x)| + \lambda|u - f|dx,$$

and defining $\Sigma = \{x|\bar{u}(x) > \mu\}$ for any $\mu \in (0, 1]$. Then Σ is a global minimizer of the original non-convex problem (4). In other words, any global minimizer is almost binary in this case, i.e., the measure of the set $\{x, u(x) \neq 0 \text{ or } 1\}$ is zero. So the most important advantage of the binary image segmentation model, TVG-L1, is that it becomes convex by reformulating the original geometric problem into the minimization over functions. Moreover, if f is close to a binary image, the minimizer is also close to a binary image with sharp transitions located at the places where $g(x)$ is small. Efficient algorithms are available for the convex minimization problem. The algorithm and implementation details will be given a little later in Section 4. However, we would like to point out that the most important issue in applying the TVG-L1 model to implicit surface reconstruction is how to construct the edge indicator $g(x)$, which should localize the surface well,

and the initial image $f(x)$, which should provide good volumetric information, i.e., indicator of inside and outside information, from unstructured and irregularly sampled PC. This issue will be discussed in detail in the next Section.

Another closely related binary image segmentation model we also consider as an alternative to the TVG-L1 model is the following CVG model proposed in [10],

$$\min_{0 \leq u \leq 1, c_1, c_2} \int_{\Omega} g(x) |\nabla u(x)| + \lambda \left[(f - c_1)^2 - (f - c_2)^2 \right] u(x) dx. \quad (5)$$

Here, $g(x)$ is an edge indicator, $f(x)$ a given image that is not necessarily binary, and c_1, c_2 are two constants. The above model is a convex constrained version of the original model by Chan and Vese [23] where it is understood that the functional is convex in u and (c_1, c_2) separately, but not biconvex. In practice, however, the minimization over u is the formal challenge. We note that if $g(x) = 1$ in (5), then the model coincides with the piecewise constant Mumford-Shah segmentation model. Again, it was shown in [20] that the problem of finding the best approximation of an image $f(x)$ by two-valued piecewise constant functions,

$$u(x) = c_1 1_{\Sigma} + c_2 1_{\Omega \setminus \Sigma} \quad (6)$$

has the following variational formulation,

$$\min_{c_1, c_2, \Sigma \subset \Omega} |\partial \Sigma| + \lambda \left[\int_{\Sigma} (f - c_1)^2 dx + \int_{\Omega \setminus \Sigma} (f - c_2)^2 dx \right], \quad (7)$$

where $\Sigma \subset \Omega$ is a region contained in the image domain and $|\partial \Sigma|$ denotes the measure of the boundary of Σ . Again the first term in the CVG model (5) is the same as the first term in the TVG-L1 model (3) which is akin to a geodesic active contour. However, the fitting term is different and involves two global constants. In particular, the CVG model is convex in c_1, c_2 when $u(x)$ is given and the optimal values are the average of $f(x)$ in each region respectively. Once c_1, c_2 are given, the optimization in u is also convex and can be efficiently computed. We implemented both models and they can produce accurate and visually appealing results that are comparable. In comparison, the TVG-L1 model is computationally more efficient and produces more accurate results if the data is not too ill behaved. On the other hand, due to the introduction of two global constants, c_1, c_2 , the CVG model is more robust with respect to noise or other difficulties that may include non-uniform sampling in the data. However, in this paper we will focus on the TVG-L1 model.

Our work is also motivated by the recent work [61], in which the authors adapt the TV based segmentation models, i.e., TVG and CVG models proposed in [10] for implicit surface reconstruction from PC. However, the most important issue in using an image segmentation model for implicit surface reconstruction is the choice of the edge indicator function and the initial image, especially when the PC is noisy, non-uniform and containing holes or open boundary. In [61], both of them are directly related to the distance function $d(x)$ to the data set. In particular, the edge indicator function is exactly $d(x)$, i.e., $g(x) = d(x)$. The initial image is constructed by the following argument. Suppose an image f is given, then a natural choice of an edge indicator function for f is simply:

$$g(x) = \frac{1}{\epsilon + |\nabla f(x)|^p} \quad (8)$$

for some $p > 0$. The foremost idea in [61] is to view the distance function $d(x)$ as the edge detector. By regularizing the distance function by setting $g = d + \epsilon > 0$ they compute the initial image f by solving the Eikonal equation:

$$|\nabla f| = \frac{1}{g^{1/p} + \epsilon}. \quad (9)$$

Note that such constructed f is always positive and does not provide any inside or outside information. Then they propose to use the CVG segmentation model from [10]:

$$\min_{0 \leq u \leq 1} \int_{\Omega} g(x) |\nabla u| + \lambda \left\{ (f - c_1)^2 - (f - c_2)^2 \right\} u(x) dx. \quad (10)$$

Thresholding the function u (usually taking the 0.5 level set) gives a coarse approximation to the implicit surface. The authors then utilize some post processing algorithms including a variant of the method found in [41] along with some TV based iterative refinement by way of the Bregman iteration found in [48]. However, the above method may suffer from the lack of inside and outside information in the initial image f which is crucial for implicit surface reconstruction. Our main contribution in this work is addressing the problem of how to choose a better edge indicator function and initial image directly from the PC, which is explained in the next section.

2.3 Poisson surface reconstruction

In order to compare our method and results with some of the state of the art methods, we briefly mention the Poisson surface reconstruction method proposed by Kazhdan et al. [39], which is a simple and efficient method for implicit surface reconstruction. The authors show that surface reconstruction from oriented points can be cast as a spatial Poisson problem. The key insight is that there is an integral relationship between oriented points sampled from a surface of a model and the indicator function of said model. In particular, the gradient of an indicator function is zero nearly everywhere with the exception at the points on a surface where the gradient coincides with the inward surface normal. In short, oriented point cloud points can be seen as samples of the gradient of the surface's indicator function. Thus, Poisson surface reconstruction has the following formulation of finding the scalar function χ such that $\nabla \chi \approx \mathbf{V}$ with \mathbf{V} being the oriented consistent normal that is computed from the point cloud. Thus, this problem has the following variational formulation:

$$\min_{\chi} \|\nabla \chi - \mathbf{V}\|^2. \quad (11)$$

The above problem (11) has the equivalent formulation with appropriate boundary condition:

$$\Delta \chi \equiv \nabla \cdot \nabla \chi = \nabla \cdot \mathbf{V}. \quad (12)$$

The authors go on to utilize an efficient octree that is adaptive to the PC resolution in both the representation of the implicit function and also in solving the Poisson system. They show watertight surface reconstructions with fine resolution even in the presence of non-uniform sampling and holes. We show results to compare our method and the Poisson reconstruction in Section 5.

3 Proposed Approach

We intend to use classical image processing methods to reconstruct surface from point cloud (PC). The major difference between point cloud problems and classical image processing problems is that there is no initial image. All that is known are the locations of the points. Hence, to view the implicit surface reconstruction for PC as an image segmentation problem we need to create an initial image and edge indicator directly from the given PC. Since the implicit surface can be thought about entirely as the segmentation of the domain properly into two regions, interior and exterior, the initial image should provide a good indication of this volumetrically. At the same time, the edge indicator function should provide more information about the precise location of the surface. In other words, the initial function gives the likelihood of a grid point to be inside or outside, and the edge indicator function gives the likelihood of a grid point on the surface. Below we will give our construction of these two functions directly from the PC. Then we outline the efficient image segmentation algorithm based on the dual formulation of the TVG-L1 model.

3.1 Edge indicator

A good edge indicator is crucial for the performance and quality of the TVG-L1 model. We design a sharp and robust edge indicator based on PCA which can extract local geometric information of the underlying surface at each data point. We call this new edge indicator anisotropic Gaussian.

3.1.1 Distance function and closest point information

Distance function to the data set $P = \{p_i\}_{i=1}^N$ is defined as $d(x) = \min_i \|x - p_i\|$ at a point x in the domain Ω . The distance function is easy to compute and gives a good indicator far away from the data set. Two major characteristics for distance function are that, first $d(x)$ is exactly 0 at data points, second $d(x)$ has a constant magnitude in gradient almost everywhere. The distance function for the 2D example is shown in left of Figure 2. A slice of the distance function for the Stanford Bunny is shown in right of Figure 2. However, due to the fact that the distance function is completely determined by the closest point, which is mutually exclusive, it suffers from the following issues,

- it behaves isotropically in all directions near a point and does not utilize neighboring points to reveal local orientation or geometry,
- it does not allow superposition or averaging and is difficult to deal with noise naturally.

We design the anisotropic Gaussian (in Section 3.1.2) to overcome these issues.

A useful byproduct of the distance function is a closest point map. At a point x in the domain Ω , denote $\text{cp}(x)$ as the data point that is closest to x , or $\text{cp}(x) = \arg \min_i \|x - p_i\|$. This closest point map is crucial for our inner product vector field, which is used as the initial guess in the TVG-L1 model. We talk about it in Section 3.2.

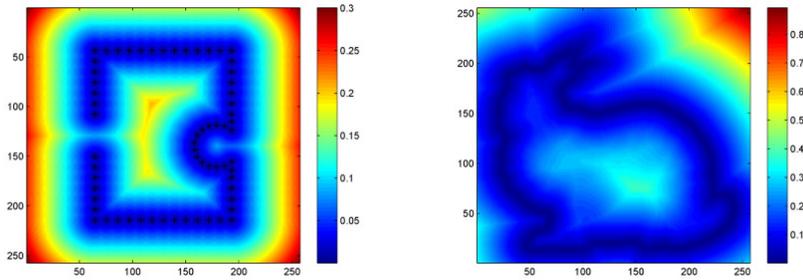


Fig. 2 Distance function. (left) is the distance function for the 2D example. (right) is a slice of the distance function for Stanford bunny.

3.1.2 Anisotropic Gaussian

We design a sharp and robust edge indicator based on PCA (principal component analysis) which can extract local geometric information of the underlying surface at each data point. At a data point p_i , we first use PCA on its KNN (K nearest neighbors) to obtain the eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq \lambda_3$ and the corresponding normalized eigenvectors v_1, v_2, v_3 respectively. We establish a local coordinate system aligned with the three eigenvectors and centered at p_i . Then we generate the following anisotropic Gaussian kernel at p_i :

$$G_i(x) = \exp\{\beta(p_i)\} \exp\left\{-s \left(\frac{x_1^2}{\lambda_1} + \frac{x_2^2}{\lambda_2} + \frac{x_3^2}{\lambda_3}\right)\right\} \quad (13)$$

where (x_1, x_2, x_3) is the local coordinate for a point $x \in \Omega$ under the local coordinate. In the above expression, the term $\exp\{\beta(p_i)\}$ is to characterize uncertainty information at p_i . For example, $\beta(p_i)$ can be related to measurement uncertainty, e.g., glancing angle, and/or fitting error in local moving least square approximation [32]. In our implementation we choose a simple formula, $\beta(p_i) = -r \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3}$ which characterizes how well the neighboring points of p_i can be approximated by a plane near p_i . If the local sampling rate is good enough to resolve the feature size, $\beta(p_i)$ will be relatively large. For corner points $\beta(p_i)$ will be relatively small. Another note is that since $\lambda_1, \lambda_2, \lambda_3$ equal to the sum of squared distance from the KNN points to the three local axes determined by the three eigenvectors from PCA respectively, the size of the Gaussian scales naturally according to the local sampling density. Moreover, the kernel is anisotropic in different directions according to the local PCA. Hence the constructed Gaussian gives a local sharp indicator of the surface at each data point based on its KNN. We compute such an anisotropic Gaussian for each data point and superpose them together to give a sharp likelihood function of the underlying surface. Similar likelihood function that characterizes uncertainty and variability of surface at each data point constructed based on weighted global information such as tensor voting [44] or the one proposed in [50], which is more expensive to compute, can be used here too. For our variational model (3), we invert the likelihood function

$$G(x) = \frac{1}{\sum_i G_i(x) + \alpha} \quad (14)$$

where $\alpha > 0$ is some constant to avoid division by zero, and scale it to $[0, 1]$ to get the anisotropic Gaussian function $g_0(x)$,

$$g_0(x) = \frac{G(x) - \min_{x \in \Omega} G(x)}{\max_{x \in \Omega} G(x) - \min_{x \in \Omega} G(x)}. \quad (15)$$

The anisotropic Gaussian function for the 2D example is shown in left of Figure 3. A slice of the anisotropic Gaussian function for the Stanford Bunny is shown in right of Figure 3. Some useful features of our edge indicator function include

- it utilizes neighboring data points to extract local geometry and is anisotropic in different directions, hence it is sharper than the distance function.
- it allows superposition from all data points as well as easy incorporation of other uncertainty information, hence it is more robust than using a distance function as the indicator.

These features are especially useful in dealing with more complicated data such as when sampling is non-uniform and noisy. An interesting observation is that the behavior of our constructed anisotropic Gaussian function is complementary to that of the distance function in the sense that it changes rapidly and provides a sharp indicator near the data set while it varies slowly (almost flat) and gives weak indication further away from the data set due to the fast decay of the exponential function. On the other hand, the distance function always has a constant magnitude in gradient and provides a good indicator away from the data set. These behaviors are demonstrated by Figures 2 and 3. Also using distance function pins down the reconstructed surface more closely to the data points. So in practice we may use both of these two in conjunction with the TVG-L1 model. First we use the distance function as the edge indicator to provide a good fit of the data points. If the data set is noisy, we use the result from the first step as an initial guess and smooth it by using the anisotropic Gaussian as the edge indicator instead.

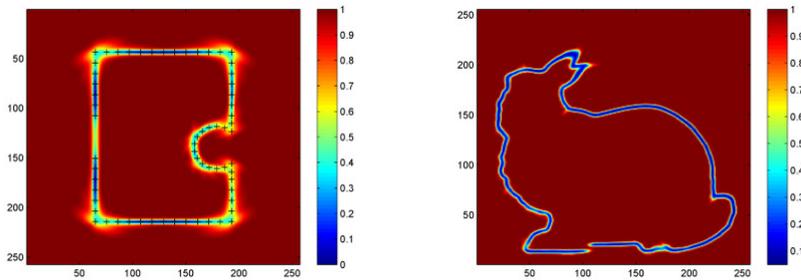


Fig. 3 Anisotropic Gaussian. (left) is the anisotropic Gaussian for the 2D example. (right) is a slice of the anisotropic Gaussian for Stanford bunny.

It is well known that the choice of K for KNN has the following guideline: larger K will increase the robustness while degrading the accuracy. We show our anisotropic Gaussian for the noisy bunny example (details in Section 5.2) with different choices of K (the number of neighboring points to be used) in Figure 4.

Expectedly, as the value of K increases, the sharpness of the anisotropic Gaussian decreases and fine features are removed while noise is suppressed.

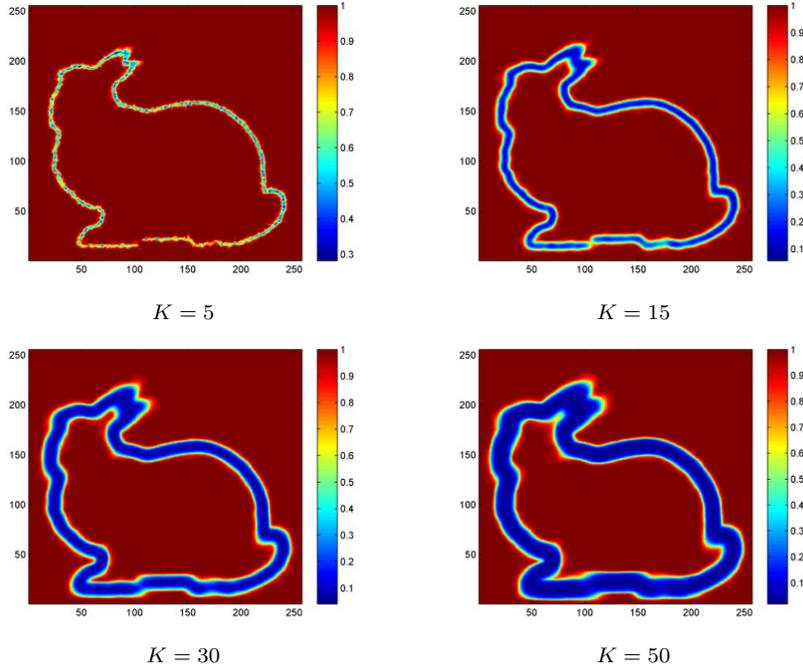


Fig. 4 Anisotropic Gaussian for the noisy bunny example for increasing values of K .

3.2 Initial image

To give a good indication of interior and exterior regions of the PC, we use the following inner product field as the likelihood function, i.e., the initial image. Suppose we have a consistent outward normal at each data point. At a point x in the domain Ω , we denote $\text{cp}(x)$ as the data point that is closest to x , or $\text{cp}(x) = \arg \min_i \|x - p_i\|$. Then we define the inner product field

$$f(x) = (x - \text{cp}(x)) \cdot n(\text{cp}(x)) \quad (16)$$

as the initial image, where $n(\text{cp}(x))$ is the outward normal at $\text{cp}(x)$. Figure 5 illustrates how $x - \text{cp}(x)$, $\text{cp}(x)$, and $n(\text{cp}(x))$ are defined. In the figure, data points are blue. The black vector is the outward normal at a given data point. The green and red vectors corresponds to $x - \text{cp}(x)$ for the green and red points respectively. In particular, $f(x)$ is positive when x is outside and is negative when x is inside. The outward normal is used at each data point. Far away from the data set, $f(x)$ is quite uniform. Near the data point $f(x)$ is more anisotropic and gives stronger indication in directions that are more aligned with the normal direction and gives

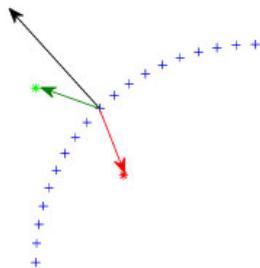


Fig. 5 Key idea of the product field. Data points are observed in blue while the black vector is the outward normal at a given data point. The green and red vectors are those vectors emanating from the closest data point to the green and red points respectively. The inner product with the normal yields an approximate indication of outside and inside of the PC.

weaker indication in directions that are closer to the tangential directions. The construction of f has two important ingredients. One is the consistent normal estimation for PC and the other one is the closest point information.

Figure 6 shows the inner product field for the 2D example and Stanford Bunny. A figure displaying the entire 0 level set of the inner product field for the Stanford Bunny can be found in upper left of Figure 8. In practice, even when the data is noisy or the normal estimation is not that accurate, the inner product field still gives a pretty good indication of the inside and outside regions of the PC data. Since this inner product field $f(x)$ is used as the initial image in the second term (fitting term) in the TVG-L1 model (3), the final implicit surface can be regarded as regularized (by the first term) or modified segmentation based on $f(x)$.

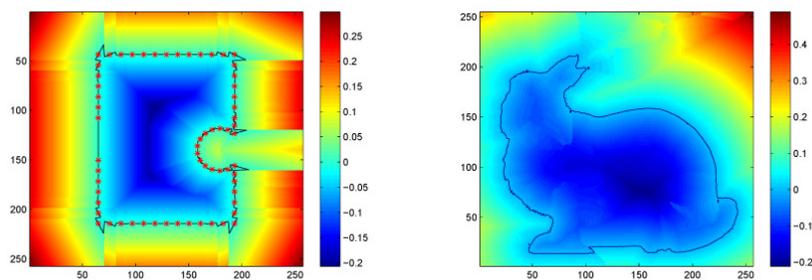


Fig. 6 Inner product field. (left) is the inner product field for the 2D example, the black curve is the 0 level set of the product field. (right) is a slice of the inner product field for Stanford Bunny, the black curve is the 0 level set of it.

Actually, using the above defined inner product field can extend an open boundary and fill holes in the data set naturally. The use of closest point and inner product extends the surface in the tangential direction at the boundary. Moreover, due to the regularization intrinsic to the variational model, extension at these boundaries or hole filling will not create edges, corners or kinks. Figure 7 illustrates the boundary extension and hole-filling scenario. For example, at the left-most boundary, points in the region left to the line that goes through the left end point and is parallel to the normal all have that left end point as the closest point. The inner product field in the region changes sign across the tangent line through the left end point. In another word the surface is extended all the way to the boundary of the bounding box for open surfaces. If we do not want to include the extended part for open surfaces, we can use distance to the data set to truncate those pieces in the reconstruction that are far away from the data points. The same thing happens in the middle where there is a hole. Extensions come from both ends and fills the hole. As demonstrated in the Figure, the filling will not be a straight line (or flat surface in 3D) as in most hole filling models based on diffusion, minimal surface or TV regularization. Although the extension from different parts of the boundary will meet and may create corners and/or edges, the first regularization term in TVG-L1 model will smooth out these corners and/or edges more and more further away from the point cloud when the edge indicator $g(x)$ becomes larger and larger. So the hole filling is neither flat nor containing corners and edges. This ability to extend surfaces at the boundary allows our method to handle not only holes in the data but also open surfaces, which are not so easy for traditional implicit surfaces. In practice, whether the hole should be filled or treated as open boundary should be problem dependent or given as a prior knowledge. Our method deals with both cases nicely. Results for both hole filling and open surfaces will be shown in Section 5.3.

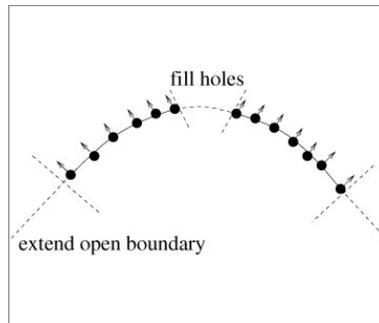


Fig. 7 Surface extension and hole-filling.

Our inner product field shares some similarities with the signed distance function introduced in [37]. Our inner product field can also be understood as the signed distance of a grid point to the tangent plane. In Hoppe’s work, their tangent plane is defined as the one passing through the mass center of the K nearest neighbor, while the inner product field used in the proposed method is defined through the data point and explicit closest point relation. Thus, our inner prod-

uct field places more trust on the data and does not average the position of the KNN. As a result, our inner product field is more precise in capturing the data. We then use the TVG-L1 model to incorporate both the inner product field and edge indicator for final surface reconstruction that can correct regions where the inner product field may be wrong and regularize the reconstructed surface as well. Also, the signed distance function in [37] is only defined close to the surface, while our inner product field is defined for the whole grid. This difference enables us to fill holes and to extend open boundaries.

As a final remark, other volumetric information can be easily incorporated in the construction of f . For example we can use available line of sight information for laser scanned PC easily to find consistent outward normals as well as fix the values of f in those cells along each line of sight.

3.3 TVG-L1 model

Here, we outline an efficient algorithm based on the dual formulation of the TVG-L1 model. There are many recent methods for solving this model directly, most notably [65, 36, 5, 17, 10, 19]. However, from an efficiency standpoint, due to the non-strict convexity of the TVG-L1 model, these methods all take an infeasible number of iterations for the solution to converge to a stationary state for our problem. Moreover, the 0.5 level set of the solution which represents the constructed surface, moves very slowly during the iterations. Also many iterations are used to make the solution u , which starts from the initial image f defined as the inner product field in (16), close to a binary function which is unnecessary for this application. Here we adopt a convex regularized version of the TVG-L1 model seen below in (17), where the problem becomes strictly convex and better conditioned and requires far fewer iterations for convergence as well as a good reconstruction. For this regularized model, we found that different methods perform comparably mainly due to the fact that we have a pretty good initial image and a sharp edge indicator to start with. For ease of use, simple parameter tuning, and guaranteed convergence, the fixed point method found in [10] based on the dual formulation of the TVG-L1 model is used. In practice, for this application, we found that this method required the fewest iterations for the 0.5 level set, i.e the surface reconstruction, to reach a steady state. As discussed in Section 3.2, our method can fill holes and extend open surfaces smoothly along the tangential directions. We give a more detailed description of the algorithm below.

For TVG-L1 model (3), we consider the convex regularized version suggested in [10] which has the following formulation:

$$\min_{u,v} \int_{\Omega} \left(g(x)|\nabla u| + \lambda|v| + \frac{1}{2\theta}|u+v-f|^2 \right) dx \quad (17)$$

where f is the initial image from (16) and $g(x)$ is an edge detector. We will use both $d(x)$, the distance function, and $g_0(x)$ the scaled inverse sum of the anisotropic Gaussian found in (15). It is assumed that the above model (17) converges to the original TVG-L1 model for θ chosen small enough, see [4]. This regularized energy can be minimized by alternately minimizing over each single variable u and v in the coupled energies:

1. for v fixed,

$$\min_u \int_{\Omega} \left(g(x)|\nabla u| + \frac{1}{2\theta}|u + v - f|^2 \right) dx \quad (18)$$

2. for u fixed,

$$\min_v \int_{\Omega} \left(\frac{1}{2\theta}|u + v - f|^2 + \lambda|v| \right) dx. \quad (19)$$

It is shown in [14,17] that the minimizer u for (18) can be found in terms of the dual variable p by the update formula

$$u = f - v - \theta \nabla \cdot p \quad (20)$$

where p solves the dual optimality equation associated to the dual formulation of the sub-minimization problem (18)

$$g(x)A(p) - |A(p)|p = 0 \quad (21)$$

with $A(p) = \nabla(\nabla \cdot p - (f - v)/\theta)$. The dual variable p can be subsequently solved for by the fixed point scheme that was introduced in [17] which was also used for minimizing the TVG-L1 model in [10]:

$$p^0 = 0,$$

$$p^{n+1} = \frac{p^n + \tau A(p^n)}{1 + \frac{\tau}{g(x)}|A(p^n)|} \quad (22)$$

which is guaranteed to converge for time step $\tau \leq 1/8$.

Now, v can easily be solved for from the shrinkage scheme below:

$$v = \begin{cases} f - u - \theta\lambda & \text{if } f - u \geq \theta\lambda \\ f - u + \theta\lambda & \text{if } f - u \leq -\theta\lambda \\ 0 & \text{if } |f - u| \leq \theta\lambda. \end{cases} \quad (23)$$

This scheme was initially proposed in [18,27] for wavelet shrinkage, but was later adapted to the TV-L1 and TVG-L1 minimization problems in [4,10] respectively.

To use the above minimization scheme, we first apply a linear transformation T to transform the inner product field $f(x)$ in (16) to $[0, 1]$ where $T(0) = 0.5$. Then using both the distance function and anisotropic Gaussian as edge indicators and the transformed inner product field as an initial guess, we run the above TVG-L1 model. Our reconstructed surface will simply be the 0.5 level set of the result u in TVG-L1 model, that is $\{x \in \Omega | u(x) = 0.5\}$.

4 Implementation

4.1 Consistent outward normal

To get the inner product field, we need consistent outward normals for the data points. Throughout our implementations, we find that an approximate consistent outward normal is sufficient enough. In practice, if some of the normals are not consistent, our approach can still reconstruct the surface correctly. For example,

even when some regions of the inner product field do not have the correct sign due to inconsistent normals, the accurate edge indicators in the TVG-L1 model can still subsequently correct those misidentified regions. All that is needed in most cases is possibly a smaller weight λ for the L1 fitting term and/or more iterations.

Due to the robustness of our method regarding normal direction, any method that can give an almost or nearly consistent normal to the data points can be utilized in our approach. For range data from The Stanford 3D Scanning Repository, one can easily obtain a consistent normal for each individual range scan set, thus amalgamating the normals from the range scans together will provide an almost consistent normal for the entire PC. For other forms of PC data, we first use PCA to estimate the normal at each data point and then use the simple method proposed in [16] to orient them consistently in which an adjacency information is formed among all data points. We refer the reader to that work for detailed description where Castillo et al. also discussed about how to use constrained nonlinear least squares to improve the normal estimation for PC near singularities.

4.2 Discretization

We use a straightforward uniform regular 3D grid to discretize the problem. We specify a grid number for the direction that has the largest range and the other two will be determined by their related range. Although this may not be memory efficient, we find such a uniform regular grid is appropriate for hole filling, open boundary, non-uniform and sparse data. For those water tight dense data, we can restrict our computation in a narrow computational band around the PC. And due to the fact that our edge indicators and initial guess image are already quite accurate, we do not need many iterations when running the TVG-L1 model. As a result, using such a uniform regular grid, the computation time for our approach is still small. In practice, we can also use an adaptive octree structure to generate a grid according to local data sampling rate to improve memory efficiency and resolution.

To speed up the computation of KNN (K nearest neighbor), PCA (Principal Component Analysis) and anisotropic Gaussian, we do a simple down-sampling preprocess based on the resolution of the uniform regular grid. Some data points may fall in the same small grid node. We simply use their mass center to replace points that are in the same small grid node and the normal for the new point is then simply the average of their normals. By such a simple simplification preprocess, the quality of the resulting surface does not change while we compute KNN, PCA and anisotropic Gaussian in a much more efficient way.

4.3 Edge indicators

We compute the distance function and find the closest point map for the whole grid using the fast sweeping method with locking [62, 6, 42]. The fast sweeping method is an efficient iterative method that can compute the distance function as well as the closest point information on a regular grid in 4 iterations for 2D data and 8 iterations for 3D. Using the locking trick for the fast sweeping method can avoid

unnecessary updates during the iteration and speed up the original fast sweeping method even further [6].

When computing the anisotropic Gaussian, we only need to compute it locally. Our anisotropic Gaussian for each data point decays so fast that only a small local region of each data point is important to be evaluated, which in turn saves computation cost significantly. In our implementation, we compute the anisotropic Gaussian values $G_i(x)$ for grids that are within the box center at p_i with length of side $3r$ where $r = \frac{1}{K} \sum_{k=1}^K \|p_k - p_i\|$ is defined as the average distance between the KNN. An interesting fact is that we can keep almost all the parameters the same for different data sets. We use 15 nearest neighbor points for all examples and set $\beta(p_i) = -5 \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3}$, $s = 1$, and $\alpha = 10$ in the computation of the anisotropic Gaussian.

4.4 TVG-L1 model

As mentioned before, we can use both the distance function and anisotropic Gaussian as edge indicators in the TVG-L1 model. Using the distance function fits the data points more closely and reveals fine features better. On the other hand, using the anisotropic Gaussian will create a smoother surface and is more robust in the presence of noise, holes, and non-uniform point cloud data. For example, the 0 level set of the inner product field for the Stanford bunny data set is shown in the upper left of Figure 8. We can see some incorrectly identified regions; for example the left ear and the base. Moreover, the field is not smooth. The TVG-L1 model using the distance function as an edge indicator and the inner product field as an initial image can successfully reconstruct the surface. However, the reconstruction is not very smooth (Figure 8 upper right). Thus, after that step, we use the previous TVG-L1 result as an initial guess and the anisotropic Gaussian as an edge indicator to run the TVG-L1 model again. The result is shown in Figure 8 lower left. In this way, we obtain both a smooth and finely reconstructed surface. Such a two step TVG-L1 model can deal with many challenging data sets and convincing numerical results will be shown in Section 5.

In our tests we found that the reconstruction results for our TVG-L1 model is not sensitive to the choice of those parameters. In general, for a good data set, i.e., clean and uniform, λ should be relatively large to enforce fidelity to the data. For noisy but pretty uniform data, then λ should be relatively small to allow more regularization. However, the most challenging case is when a data set is both noisy and non-uniform, i.e., has holes and openings, then the choice of λ becomes a little bit subtle. On one hand, λ should be relatively small to suppress the noise. On the other hand, smaller λ makes the surface stiffer, which may not adhere to the data and collapse through the holes or openings. For example, the LiDAR data set in our tests below is a challenging case. We have to pick the parameters more carefully to produce a good result. In our implementation, we scale the distance function, anisotropic Gaussian and inner product field to be in $[0, 1]$. And we use $\lambda = 0.01$ and $\theta = 0.05$ for all the examples except the LiDAR data example with $\lambda = 0.05$ and $\theta = 0.05$. Solving for the final solution amounts to alternating between the two equations (22) and (23) using the initial conditions $u_0 = v_0 = 0$ and $p_0 = \langle p_0^1, p_0^2, p_0^3 \rangle \equiv \mathbf{0}$ with time step taken as $\tau = 1/16$ for 3D examples. Discretization follows in exactly the same way as discussed in [17, 10].

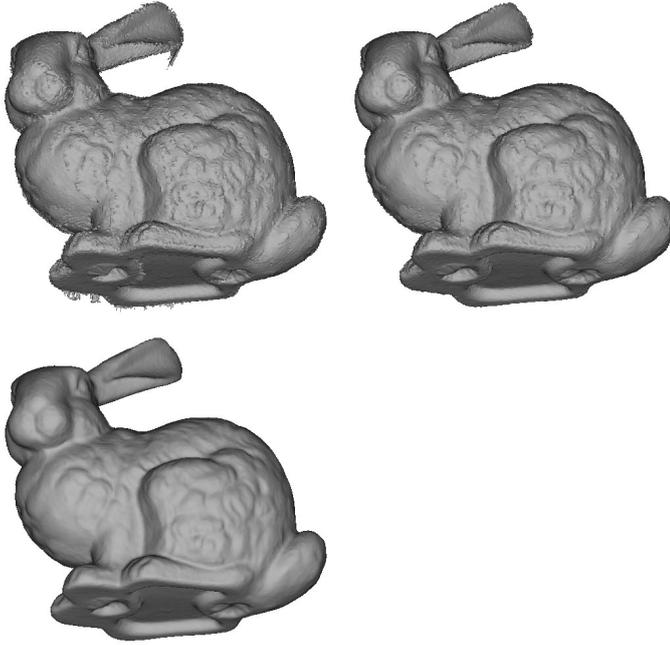


Fig. 8 Stanford Bunny. (upper left) is the 0 level set of inner product field. (upper right) is the result obtained from the TVG-L1 model using distance as an edge indicator and inner product as an initial guess. (lower left) is the result of the TVG-L1 model using anisotropic Gaussian as an edge indicator and the result in (upper right) as the initial guess.

The stopping criteria is $\frac{\|u_{n+1}-u_n\|_2}{\|u_n\|_2} \leq tol$, for some prescribed tolerance tol . Lastly, the final segmented region Σ is obtained by simply taking the 0.5 upper level set of the minimizer $u(x)$; strictly speaking $\Sigma = \{x|u(x) \geq 0.5\}$. The reconstructed surface to the PC is then realized by taking the $\mu = 0.5$ level set of $u(x)$. i.e. $\partial\Sigma = \{x|u(x) = 0.5\}$.

5 Results

To evaluate our method, we conduct a series of experiments. We test our method on 5 types of data: 3D scanning real-world data, noisy data, data with holes or open boundary, non-uniform and/or sparse data, and LiDAR data. We use both MeshLab and MatLab to visualize our results and we compare our results to those obtained from Poisson surface reconstruction. The Poisson Surface Reconstruction (Ver. 2) source code from Professor Michael Kazhdan's website (<http://www.cs.jhu.edu/~misha/Code/PoissonRecon/>) is used. The same grid resolution is used for both methods. Due to the following facts (1) Poisson reconstruction uses a heat kernel, i.e., isotropic diffusion, to regularize the implicit surface reconstruction while our method uses total variation, i.e., surface area, as the regularization, (2) Poisson uses a simple average of normals at data points to construction the vector field \mathbf{V} while our method uses the closest point relation

to create the inner product field, our method can resolve more fine details than Poisson reconstruction. Also for Poisson reconstruction, \mathbf{V} may not be adequate to provide accurate inside and outside information in some regions and additional line of sight information may be needed near features that have ambiguous or multiple possible reconstructions.

5.1 Range data

We obtain similar results for the Stanford Bunny data set (10 scans, 362,271 points, Figure 9) and Stanford Dragon (61 scans, 1,769,269 points, Figure 10) as Poisson surface reconstruction. In the reconstruction obtained from the proposed method, we observe more details and a sharper boundary for the base of Stanford Bunny.

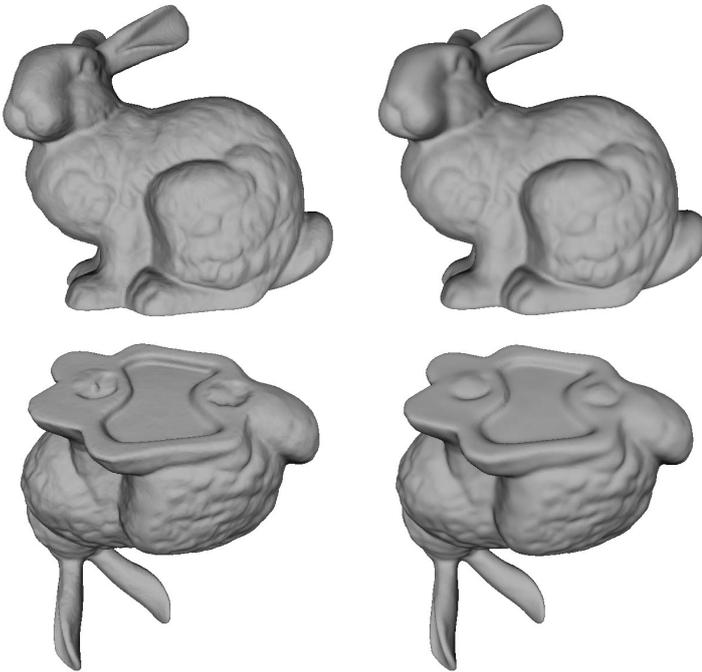


Fig. 9 Stanford Bunny. (upper left) is the reconstructed surface from the proposed model. (upper right) is result of Poisson surface reconstruction. (lower left) is the bottom view of (upper left). (lower right) is the bottom view of (upper right).

An interesting range data example is the Stanford Happy Buddha (58 scans, 3,344,135 points, Figure 11). Since there are no samples between the two feet, $\nabla \cdot \mathbf{V}$ does not provide adequate information in this region and Poisson surface reconstruction connects the two feet. Our method can separate the two regions correctly. Notice that in the regions in the red boxes, some small parts of tiny features are missing for both methods. One possible reason is that the anisotropic

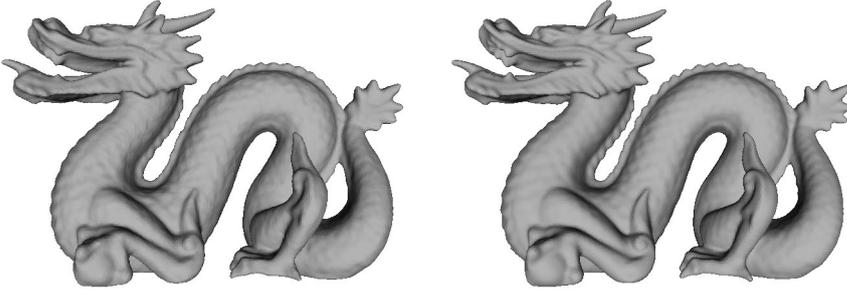


Fig. 10 Stanford Dragon. (left) is our reconstructed surface. (right) is result of Poisson surface reconstruction.

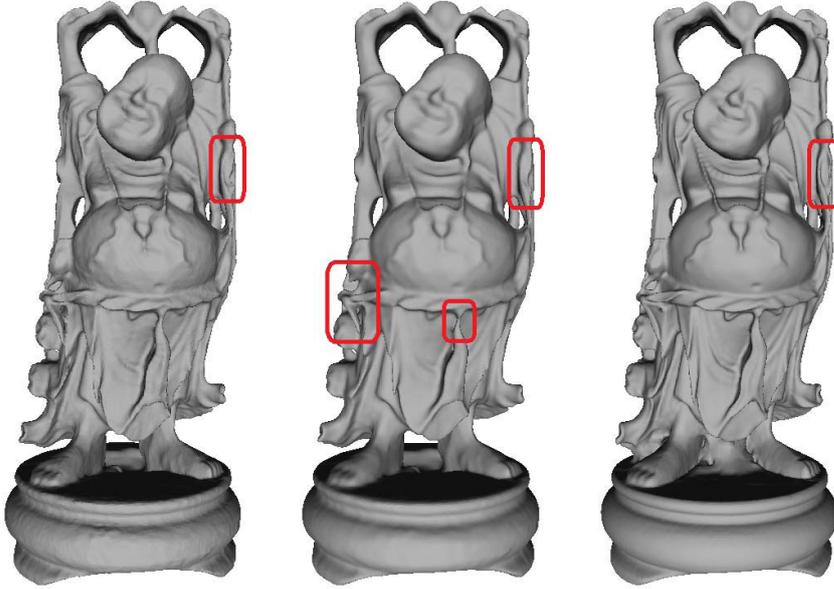


Fig. 11 Stanford Happy Buddha. (left) is our reconstructed surface using $\lambda = 0.01$ and $\theta = 0.05$ in both steps of TVG-L1. (middle) is our reconstructed surface using $\lambda = 0.01$ and $\theta = 0.05$ in the first step of TVG-L1 and $\lambda = 0.01$ and $\theta = 1$ in the second step of TVG-L1. (right) is result of Poisson surface reconstruction.

Gaussian not only removes noise but also smoothes out diminutive scale features. Using small θ (we use $\theta = 0.05$ in left of Figure 11) in TVG-L1 model (17) can keep most of the features but the surface may not be smooth enough. One can set θ larger (we use $\theta = 1$ in middle of Figure 11 in the second step of TVG-L1 model) to achieve a smoother surface, but at the same time, more ultra fine features will be smoothed out (red boxes in middle of Figure 11). Notice that in the convex regularized version of TVG-L1 model in (17), larger θ allows the result to be smoother. But at the same time, some exceedingly small scale features will

be smoothed out. Such an observation agrees with the study of the effect of θ in the convex relaxed version of the TVG-L1 model in [10].

5.2 Noisy data

We test our method on noisy data. Gaussian noise with 0 mean and a standard deviation of 0.2% of the diagonal of the bounding box is added to Stanford bunny data points. Similar results are obtained, see Figure 12, while our method can retrieve more details than what can be obtained by the Poisson method.

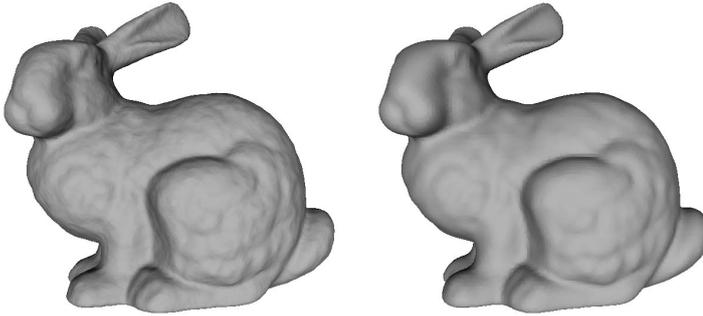


Fig. 12 Noisy Stanford Bunny. (left) is the proposed reconstructed surface of TVG-L1. (right) is the result from Poisson surface reconstruction.

5.3 Data with holes or open boundary

Here we show a few tests for data with missing pieces, such as holes and scratches. Figure 13 shows the result of our method and that of Poisson for Stanford Bunny data with defects. It can be seen that our method can inpaint features better than that of Poisson since we utilize the closest point information and extend tangentially from the boundary of the hole while Poisson surface reconstruction uses simple averaging of information and hence tries to extend in a more isotropic manner.

The knot example shown in Figure 14 is more challenging since the size of missing parts of the data is comparable to or even larger than that of local features. As can be seen from the reconstruction by Poisson method, the reconstructed missing piece shrinks where the size of missing part (right gap) is comparable to that of local features. Moreover, Poisson reconstruction disconnects the top gap where the size of missing part is larger than that of local features, i.e., the length of the missing gap is larger than the diameter of the knot. While our approach can smoothly and successfully reconstruct those missing parts of the data.

We also test the proposed method on data with an open boundary (Figure 15). Our approach can reconstruct the surface with fine details and extend the

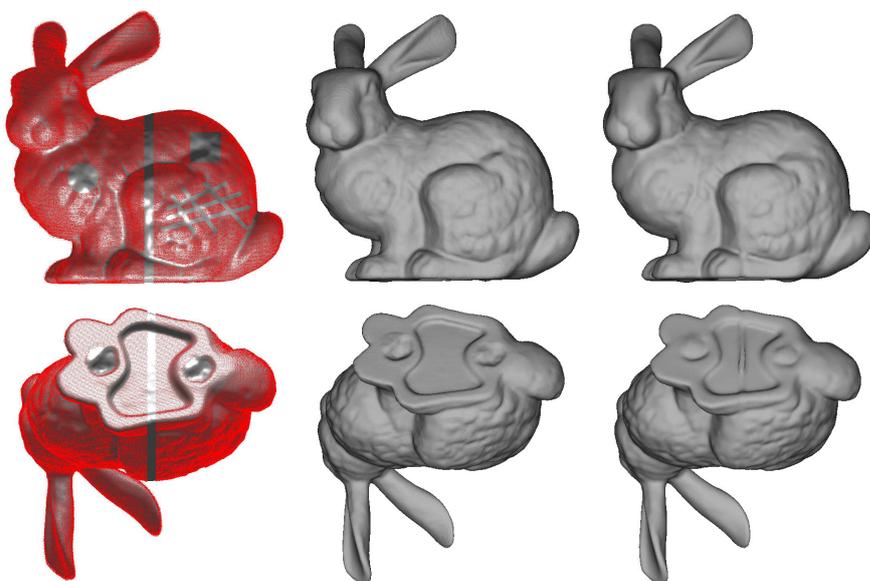


Fig. 13 Bunny with holes. (upper left) is the reconstructed surface from the proposed model with data points overlaid. (upper middle) is the proposed reconstructed surface. (upper right) is result of Poisson surface reconstruction. The lower row is the bottom view of the upper row.

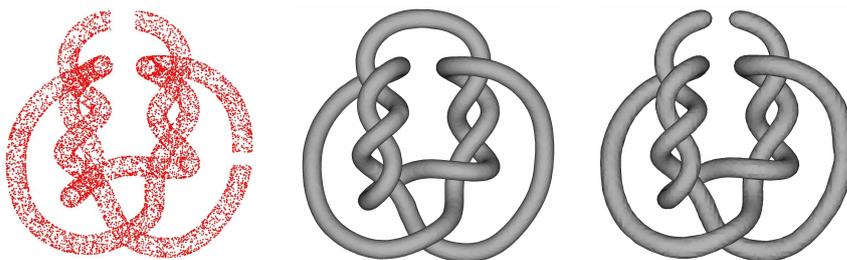


Fig. 14 Complicated Topology. Knots with holes. (left) is a set of data points. (middle) is the reconstructed surface from the proposed model. (right) is the result from Poisson surface reconstruction.

boundary in a natural way. The result of Poisson reconstruction appears to be over-smoothed and does not extend the boundary.

5.4 Non-uniform data

We test the ability of our approach to handle large variations in sampling density. To do that, we remove 99% of the original points from the right half of the Armadillo data set and then apply the proposed method. Our method is able to successfully handle a 100-to-1 sampling difference with accurate results (Figure 16). We also test the proposed method on sparse data where we remove 99% of



Fig. 15 Open Boundary Extension and Reconstruction. Venus Example. (left) is a set of data points. (middle) is the reconstructed surface from the proposed model. (right) is the result of Poisson surface reconstruction.

points from the entire Armadillo data set and then apply our method. The proposed method can still reconstruct the surface with much finer detail and features than the Poisson reconstruction (Figure 17) despite the sparsity of the data.



Fig. 16 Armadillo (non-uniform). (left) is a set of non-uniform data points, 99% of the original points are removed from the right half of the Armadillo. (middle) is the reconstructed surface from the proposed model. (right) is the result from Poisson surface reconstruction.

5.5 LiDAR data

Finally, we try our method on challenging real LiDAR data (Figure 18). The data is segmented from a large LiDAR data set. One of the main challenges comes from sparsity and non-complete data due to occlusion or shadow. The data is one-sided and also has some missing parts like the left broken (disconnected) leg. Our reconstructed surface shows that our algorithm can deal with such difficult data quite reasonably. We can connect the broken leg and capture features such as gaps between two legs.



Fig. 17 Armadillo (sparse). (left) is a sparse set of data points, 99% of the original points are removed. (middle) is the proposed reconstructed surface. (right) is the result obtained from Poisson surface reconstruction.

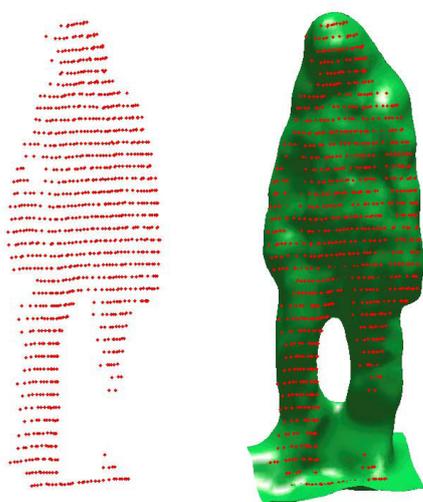


Fig. 18 LiDAR data. (left) is a LiDAR data point set. (right) is the reconstructed surface from the proposed model with data points overlaid.

5.6 Performance and Limitations

All experiments are performed on a laptop with an Intel(R) Core(TM) i5-2430M CPU @ 2.40 GHz processor and 8.00 GB RAM memory. As mentioned before, our algorithm is not particularly sensitive to parameters. we use $\lambda = 0.01$ and $\theta = 0.05$ for all the examples except the LiDAR data example with $\lambda = 0.05$ and $\theta = 0.05$. Only grid sizes and numbers of iterations of TVG-L1 are different. Parameters and CPU running time (in seconds) for all 3D experiments are shown in the following table.

	pt size	pt used	grid size	ini proc	TVG-L1	tolerance	iteration
bunny	362,271	157,860	257×255×201	12.82	21.63	[2.5E-4, 2.0E-2]	[47, 5]
dragon	1,769,269	170,188	257×181×117	8.71	3.74	[3.0E-3, 1.0E-2]	[6, 5]
Buddha (left)	3,344,135	632,783	215×513×213	38.13	82.18	[2.5E-4, 6.0E-3]	[63, 8]
Buddha (middle)	3,344,135	632,783	215×513×213	38.13	101.89	[2.5E-4, 8.0E-3]	[63, 27]
bunny (noisy)	362,271	227,758	257×255×201	17.37	24.94	[2.5E-4, 8.0E-3]	[46, 13]
bunny (holes)	341,387	149,954	257×255×201	12.47	16.74	[1.5E-3, 1.2E-3]	[15, 8]
knot	9,785	8,174	127×129×63	2.01	0.92	[2.5E-3, 2.0E-3]	[8, 5]
Venus	44,992	43,820	181×251×257	16.16	6.77	[1.5E-3, 1.2E-3]	[5, 5]
Armadillo(non-uniform)	1,142,120	111,377	217×257×199	11.68	13.21	[2.5E-4, 2.0E-2]	[43, 5]
Armadillo(sparse)	22,403	20,456	215×257×195	10.83	13.64	[2.5E-4, 2.0E-2]	[52, 5]
LiDAR	1,009	858	37×27×129	0.19	0.53	[2.5E-3, 2.0E-3]	[55, 5]

‘pt size’ is the number of data points. ‘pt used’ is the number of points after we replace points in the same grid cell by their mass center. ‘ini proc’ is sum of the initial processing time (simplification of points, fast sweeping method with locking to compute distance function and closest point information, computing KNN and PCA, anisotropic Gaussian and inner product field). ‘TVG-L1’ is the time for TVG-L1. We also report tolerance and number of iterations for TVG-L1 model. The first one is the tolerance and number of iteration using distance function as edge indicator and inner product field as the initial guess. The second one is the tolerance and number of iteration using anisotropic Gaussian as edge indicator and the result of the first TVG-L1 as the initial guess.

A limitation of our method is that it uses a uniform grid. A grid that is adaptive to sampling density with an octree implementation may help to improve overall efficiency and accuracy.

6 Conclusions

In this paper, we proposed a robust and efficient algorithm for implicit surface reconstruction that exploits the underlying geometric structure of the PC data combined with a global convexified image segmentation formulation. The main advantage of our method is its ability to deal with challenging PC data that have complicated geometry and topology as well as holes, open boundary, and non-uniformity.

Acknowledgements The authors would like to thank Edward Castillo for graciously providing us with his code for computing surface normals to PC data. The author would like to express their thanks to the Stanford 3D scanning Repository for their generosity in distributing their 3D data. The authors would also like to thank Professor Michael Kazhdan for graciously distributing his source code for Poisson surface reconstruction.

References

1. F. Almgren, J.E. Taylor and L. Wang, “Curvature-driven Flows: a Variational Approach,” *SIAM J. Control Optim.* vol. 32, pp. 387-438, 1993, doi:10.1137/0331020.
2. N. Amenta, M. Bern and D. Eppstein, “The Crust and the Beta-skeleton: Combinatorial Curve Reconstruction,” *Graphical Models and Image Process.*, vol. 60, pp. 125-135, Mar. 1998, doi:10.1006/gmip.1998.0465.
3. N. Amenta, M. Bern and M. Kamvysselis, “A new Voronoi-based Surface Reconstruction Algorithm,” *Proc. 25th Annu. Conf. Comput. Graph. and Interactive Tech. (SIGGRAPH ’98)*, pp. 415-421, 1998, doi:10.1145/280814.280947.
4. J.F. Aujol, G. Gilboa, T. Chan and S. Osher, “Structure-texture Image Decomposition - Modeling, Algorithms, and Parameter Selection,” *Int. J. Comput. Vision*, vol. 67, pp. 111-136, Apr. 2006, doi:10.1007/s11263-006-4331-z.

5. E. Bae, J. Yuan and X.C. Tai, "Global Minimization for Continuous Multiphase Partitioning Problems Using a Dual Approach," *Int. J. Comput. Vision*, vol. 92, pp. 112-129, Mar. 2011, doi:10.1007/s11263-010-0406-y.
6. S. Bak, J. McLaughlin and D. Renzi, "Some Improvements for the Fast Sweeping Method," *SIAM J. Scientific Comput.*, vol. 32, pp. 2853-2874, Agu. 2010, doi:10.1137/090749645.
7. J.D. Boissonnat, "Geometric Structures for Three Dimensional Shape Reconstruction," *ACM Trans. Graph.*, vol. 3, pp. 266-286, Oct. 1984, doi:10.1145/357346.357349.
8. Y. Boykov and V. Kolmogorov, "Computing Geodesics and Minimal Surfaces via Graph Cuts," *IEEE Int. Conf. Comput. Vision (ICCV '03)*, vol. 1, pp. 26-33, Oct. 2003, doi:10.1109/ICCV.2003.1238310.
9. X. Bresson and T. Chan, "Active Contours Based on Chambolle's Mean Curvature Motion," *IEEE Int. Conf. Image Process. (ICIP '07)*, vol. I, pp. 33-36, 2007, doi:10.1109/ICIP.2007.4378884.
10. X. Bresson, S. Esedoglu, P. Vanderheynt, J.P. Thiran and S. Osher, "Fast Global Minimization of the Active Contour/Snake Model," *J. Math. Imaging and Vision*, vol. 28, pp. 151-167, Jun. 2007, doi:10.1007/s10851-007-0002-0.
11. M. Burger, S. Osher, J. Xu and G. Gilboa, "Nonlinear Inverse Scale Space Methods for Image Restoration," *Commun. Math. Sci.*, vol. 3752, pp. 25-36, 2005, doi:10.1007/11567646_3.
12. J.C. Carr, R.K. Beatson, J.B. Cherrie, T.J. Mitchell, W.R. Fright, B.C. McCallum and T.R. Evans, "Reconstruction and Representation of 3D Objects with Radial Basis Functions," *Proc. 28th Ann. Conf. Comput. Graph. and Interactive Tech. (SIGGRAPH '01)*, pp. 67-76, 2001, doi:10.1145/383259.383266.
13. J.C. Carr, W.R. Fright and R.K. Beatson, "Surface Interpolation with Radial Basis Functions for Medical Imaging", *IEEE Trans. Med. Imag.*, vol. 16, pp. 96-107, Feb. 1997, doi:10.1109/42.552059.
14. J. Carter, "Dual methods for total variation-based image restoration," Ph.D. dissertation, Dept. Math., UCLA, LA, CA, 2001.
15. V. Caselles, R. Kimmel and G. Sapiro, "Geodesic Active Contours," *Proc. 5th Int. Conf. Comput. Vision (ICCV '97)*, pp. 694-699, 1997, doi:10.1109/ICCV.1995.466871.
16. E. Castillo and H.K. Zhao, "Point Cloud Segmentation via Constrained Nonlinear Least Squares Surface Normal Estimates", M. Breuss, A. Bruckstein, P. Maragos (Eds.): *Innovations for Shape Analysis: Models and Algorithms*. Springer, Berlin, 2012, in press.
17. A. Chambolle, "An Algorithm for Total Variation Minimization and Applications," *J. Math. Imag. and Vision*, vol. 20, pp. 89-97, 2004, doi:10.1023/B:JMIV.0000011321.19549.88.
18. A. Chambolle, R.A. De Vore, N.T. Lee and B.J. Lucier, "Nonlinear Wavelet Image Processing: Variational Problems, Compression, and Noise Removal through Wavelet Shrinkage," *IEEE Trans. Image Process.*, vol. 7, pp. 319-335, 1998, doi:10.1109/83.661182.
19. A. Chambolle and T. Pock, "A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging", *J. Math. Imag. and Vision*, vol. 20, no. 1, pp. 120-145, 2011, doi:10.1007/s10851-010-0251-1.
20. T.F. Chan, S. Esedoglu and M. Nikolova, "Algorithms for Finding Global Minimizers of Image Segmentation and Denoising Models," *SIAM J. Appl. Math.*, vol. 66, pp. 1632-1648, 2006, doi:10.1137/040615286.
21. T.F. Chan, G. Golub and P. Mulet, "A Nonlinear Primal-Dual Method for Total Variation-Based Image Restoration," *SIAM J. Sci. Comput.*, vol. 20, pp. 1964-1977, 1999, doi:10.1137/S1064827596299767.
22. T.F. Chan and S. Esedoglu, "Aspects of Total Variation Regularized L^1 Function Approximation," *SIAM J. Appl. Math.*, vol. 65, no. 5, pp. 1817-1837, 2005, doi:10.1137/040604297.
23. T.F. Chan and L.A. Vese, "Active Contours without Edges," *IEEE Trans. Image Process.*, vol. 10, pp. 266-277, Feb. 2001, doi:10.1109/83.902291.
24. R. Courant, K. Friedrichs and H. Lewy, "On the Partial Difference Equations of Mathematical Physics," *IBM J. Research and Develop.*, vol. 11, pp. 215-234, 1967, doi:10.1147/rd.112.0215.
25. B. Curless and M. Levoy, "A Volumetric Method for Building Complex Models from Range Images," *Proc. 23rd Ann. Conf. Comput. Graph. and Interactive Tech. (SIGGRAPH '96)*, pp. 303-312, 1996, doi:10.1145/237170.237269.
26. J. Digne, J.M. Morel, C.M. Souza and C. Lartigue, "Scale Space Meshing of Raw Data Point Sets," *Comput. Graph. Forum*, vol. 30, pp. 1630-1642, 2011, doi:10.1111/j.1467-8659.2011.01848.x.

27. D.L. Donoho and I.M. Johnstone, "Adapting to Unknown Smoothness via Wavelet Shrinkage," *J. Amer. Stat. Assoc.*, vol. 90, no. 432, pp. 1200-1224, Dec. 1995, doi:10.2307/2291512.
28. F. Duguet, F. Durand and G. Drettakis, "Robust Higher-Order Filtering of Points," *Tech. Rep. INRIA*, Apr. 2004, doi:10.1.1.59.152.
29. H. Edelsbrunner, "Shape Reconstruction with Delaunay Complex", *Proc. 3rd Latin Amer. Symp. Theoretical Informat. (LATIN '98)*, vol. 1380, pp. 119-132, 1998, doi:10.1007/BFb0054315.
30. H. Edelsbrunner and E.P. Mucke, "Three Dimensional Alpha Shapes," *ACM Trans. Graph.*, vol. 13, pp. 43-72, Jan. 1994, doi:10.1145/174462.156635.
31. H. Federer, *Geometric Measure Theory*, New York: Springer, 1969.
32. S. Fleishman, D. Cohen-Or and C.T. Silva, "Robust Moving Least-Squares Fitting with Sharp Features," *ACM Trans. Graph.*, *Proc. ACM SIGGRAPH 2005*, vol. 24, Jul. 2005, doi:10.1145/1186822.1073227.
33. S. Fleishman, I. Drori and D. Cohen-Or, "Bilateral Mesh Denoising," *ACM Trans. Graph.*, *Proc. ACM SIGGRAPH 2003*, vol. 22, pp. 950-953, Jul. 2003, doi:10.1145/882262.882368.
34. E. Giusti, *Minimal Surfaces and Functions of Bounded Variation*, Boston: Birkhauser, 1984.
35. T. Goldstein, X. Bresson and S. Osher, "Geometric Applications of the Split Bregman Method: Segmentation and Surface Reconstruction," *CAM*, LA, CA, Rep. 09-06, 2009.
36. T. Goldstein and S. Osher, "The Split Bregman Method for L1-Regularized Problems," *SIAM J. Imag. Sci.*, vol. 2, pp. 323-343, 2009, doi:10.1137/080725891.
37. H. Hoppe, T. Duchamp, J. McDonald and W. Stuetzle, "Surface Reconstruction from Unorganized Points," *Proc. 19th Ann. Conf. Comput. Graph. and Interactive Tech.*, *ACM SIGGRAPH Comput. Graph.*, vol. 26, pp. 71-78, Jul. 1992, doi:10.1145/142920.134011.
38. M. Kass, A. Witkin and D. Terzopoulos, "Snakes: Active Contour Models," *Int. J. Comput. Vision*, vol. 1, pp. 321-331, 1988, doi:10.1007/BF00133570.
39. M. Kazhdan, M. Bolitho and H. Hoppe, "Poisson Surface Reconstruction," *Proc. 4th Eurographics Symp. Geometry Process. (SGP '06)*, pp. 61-70, 2006.
40. R. Kolluri, J.R. Shewchuk and J.F. O'Brien, "Spectral Surface Reconstruction from Noisy Point Clouds," *Proc. 2004 Eurographics/ACM SIGGRAPH Symp. Geometry Process. (SGP '04)*, pp. 11-21, 2004, doi:10.1145/1057432.1057434.
41. V. Lempitsky and V. Boykov, "Global Optimization for Shape Fitting," *IEEE Conf. Comput. Vision and Pattern Recognition (CVPR '07)*, pp. 1-8, Jun. 2007, doi:10.1109/CVPR.2007.383293.
42. S. Luo, L.J. Guibas and H.K. Zhao, "Euclidean Skeletons Using Closest Points," *Inverse Problem and Imag.*, vol. 5, pp. 95-113, Feb. 2011, doi:10.3934/ipi.2011.5.95.
43. B. Mederos, L. Velho and L.H. de Figueiredo, "Point Cloud Denoising," *Proc. SIAM Conf. Geometric Design and Comput.*, 2003.
44. G. Medioni, M.S. Lee and C.K. Tang, *A Computational Framework for Segmentation and Grouping*, Elsevier, 2000.
45. B.S. Morse, T.S. Yoo, D.T. Chen, P. Rheingans and K.R. Subramanian, "Interpolating Implicit Surfaces from Scattered Surface Data Using Compactly Supported Radial Basis Functions," *Proc. Int. Conf. Shape Modeling & Applicat. (SMI '01)*, 2001.
46. D. Mumford and J. Shah, "Optimal Approximation by Piecewise Smooth Functions and Associated Variational Problems," *Commun. Pure Appl. Math.*, vol. 42, no. 5, pp. 577-685, 1989, doi:10.1002/cpa.3160420503.
47. Y. Ohtake, A. Belyaev and H.P. Seidel, "A Multi-Scale Approach to 3D Scattered Data Interpolation with Compactly Supported Basis Function," *Shape Modeling Int.*, pp. 153-161, May 2003, doi:10.1109/SMI.2003.1199611.
48. S. Osher, M. Burger, D. Goldfarb, J. Xu and W. Yin, "An Iterative Regularization Method for Total Variation-Based Image Restoration," *SIAM Multiscale Modeling & Simulation*, vol. 4, pp. 460-489, 2005, doi:10.1137/040605412.
49. M. Pauly, M. Gross and L.P. Kobbelt, "Efficient Simplification of Point-Sampled Surfaces," *Proc. Conf. Visualization (VIS '02)*, 2002.
50. M. Pauly, N.J. Mitra and L. Guibas, "Uncertainty and Variability in Point Cloud Surface Data," *Symp. Point-Based Graph.*, pp. 77-84, 2004.
51. D. Peng, B. Marriman, S. Osher, H.K. Zhao and M. Kang, "A PDE-Based Fast Local Level Set Method," *J. Comput. Physics*, vol. 155, pp. 410-438, Nov. 1999, doi:10.1006/jcph.1999.6345.

52. L. Piegl and W. Tiller, *The NURBS Book*, 2nd ed. New York, NY: Springer-Verlag, 1997.
53. D.F. Rogers, *An Introduction to NURBS: with Historical Perspective*, 1st ed. San Francisco, CA: Morgan Kaufmann, 2001.
54. D.F. Rogers, *An Introduction to NURBS*, San Francisco, CA: Morgan Kaufmann, 2003.
55. L.I. Rudin, S. Osher and E. Fatemi, "Nonlinear Total Variation Based Noise Removal Algorithms," *Physica D*, vol. 60, pp. 259-268, Nov. 1992, doi:10.1016/0167-2789(92)90242-F.
56. J.A. Sethian, "A Fast Marching Level Set Method for Monotonically Advancing Fronts," *Proc. Nat. Acad. Sci.*, vol. 93, pp. 1591-1595, Feb. 1996, doi:10.1073/pnas.93.4.1591.
57. J.A. Sethian, *Level Set Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, 2nd ed. Cambridge University Press, 1999.
58. Y.R. Tsai, "Rapid and Accurate Computation of the Distance Function Using Grids," *J. Comput. Phys.*, vol. 178, pp. 175-195, May 2002, doi:10.1006/jcph.2002.7028.
59. M. Wan, Y. Wang, E. Bae, X.C. Tai and D. Wang, "Reconstructing Open Surfaces via Graph-Cuts," CAM, LA, CA, Rep. 10-29, 2010.
60. H. Wendland, "Fast Evaluation of Radial Basis Functions: Methods Based on Partition of Unity," *Approximation Theory X: Wavelets, Splines and Applicat.*, vol. 473, pp. 473-483, 2002, doi:10.1.1.7.6130.
61. J. Ye, X. Bresson, T. Goldstein and S. Osher, "A Fast Variational Method for Surface Reconstruction from Sets of Scattered Points," CAM, LA, CA, Rep. 01-10, 2001.
62. H.K. Zhao, "A Fast Sweeping Method for Eikonal Equations," *Math. Comput.*, vol. 74, pp. 603-627, 2005, doi:10.1.1.153.6742.
63. H.K. Zhao, S. Osher and R. Fedkiw, "Fast Surface Reconstruction Using the Level Set Method," *Proc. IEEE Workshop Variational and Level Set Methods Comput. Vision*, pp. 194-201, Aug. 2002, doi:10.1109/VLSM.2001.938900.
64. H.K. Zhao, S. Osher, B. Merriman and M. Kang, "Implicit, Nonparametric Shape Reconstruction from Unorganized Points Using a Variational Level Set Method," *Comput. Vision and Image Understanding*, vol. 80, pp. 295-314, Dec. 2000, doi:10.1006/cviu.2000.0875.
65. M. Zhu and T.F. Chan, "An Efficient Primal-Dual Hybrid Gradient Algorithm for Total Variation Image Restoration," CAM, LA, CA, Rep. 08-34, 2008.
66. C. Zach, T. Pock and H. Bischof, "A Globally Optimal Algorithm for Robust TV-L1 Range Image Integration," *Proc. 11th IEEE Int. Conf. Comput. Vision (ICCV '07)*, vol. 1, pp. 1-8, Oct. 2007, doi:10.1109/ICCV.2007.4408983.
67. N. Amenta and J.K. Yong, "Defining Point-Set Surfaces," *ACM Trans. Graph.*, vol. 23, Aug. 2004, doi:10.1145/1015706.1015713.
68. M. Alexa and A. Adamson, "On Normals and Projection Operators for Surfaces Defined by Point Sets," *Eurographics Symp. Point-Based Graph.*, pp. 149-155, 2009, doi:10.1.1.116.873.
69. M. Alexa and A. Adamson, "Interpolatory Point Set Surfaces-Convexity and Hermite Data," *ACM Trans. Graph.*, vol. 28, Apr. 2009, doi:10.1145/1516522.1516531.
70. A. Adamson and M. Alexa, "Anisotropic Point Set Surfaces," *Comput. Graph. Forum*, vol. 25, pp. 717-724, Dec. 2006, doi:10.1111/j.1467-8659.2006.00994.x.
71. C. Shen, J.F. Brien and J.R. Shewchuk, "Interpolation and Approximating Implicit Surfaces from Polygonal Soup," *ACM Trans. Graph.*, vol. 23, Aug. 2004, doi:10.1145/1015706.1015816.
72. C. Oztireli, G. Guennebaud and M. Gross, "Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression," *Comput. Graph. Forum*, vol. 28, no. 2, pp. 493-501, Apr. 2009, doi:10.1111/j.1467-8659.2009.01388.x.
73. G. Guennebaud and M. Gross, "Algebraic Point Set Surfaces," *Proc. ACM SIGGRAPH 2007 (SIGGRAPH '07)*, *ACM Trans. Graph.*, vol. 26, Jul. 2007, doi:10.1145/1276377.1276406.
74. H. Takeda, S. Farsiu and P. Milanfar, "Kernel Regression for Image Processing and Reconstruction," *IEEE Trans. Image Process.*, vol. 16, pp. 349-366, Feb. 2007, doi:10.1109/TIP.2006.888330.
75. M. Pauly, R. Keiser, L.P. Kobbelt and M. Gross, "Shape Modeling with Point-Sampled Geometry," *ACM Trans. Graph.*, vol. 22, iss. 3, pp. 641-650, Jul. 2003, doi:10.1145/882262.882319.
76. A. Sharf, T. Lewiner, A. Shamir, L. Kobbelt and D. Cohen-or, "Competing Fronts for Coarse-to-Fine Surface Reconstruction," *Comput. Graph. Forum*, vol. 25, pp. 389-398, Oct. 2006, doi:10.1111/j.1467-8659.2006.00958.x.

77. A. Hornung and L. Kobbelt, "Robust Reconstruction of Watertight 3D Models from Non-uniformly Sampled Point Clouds Without Normal Information," Proc. 4th Eurographics Symp. Geometry Process. (SGP '06), Jun. 2006, doi:10.1145/1281957.1281963.
78. P. Alliez, D. Cohen-Steiner, Y. Tong and M. Desbrun, "Voronoi-Based Variational Reconstruction of Unoriented Point Sets," Proc. 5th Eurographics Symp. Geometry Process. (SGP '05), pp. 39-48, Jul. 2007, doi:10.1145/1281991.1281997.
79. P. Mullen, F. de Goes, M. Desbrun, D. Cohen-Steiner and P. Alliez, "Signing the Unsigned: Robust Surface Reconstruction from Raw Pointsets," Comput. Graph. Forum, vol. 29, no. 5, pp. 1733-1741, Jul. 2010, doi:10.1111/j.1467-8659.2010.01782.x.
80. C. Zach, "Fast and High Quality Fusion of Depth Maps," Int. Symp. 3D Data Process. Visualization Transmission (3DPVT), Jun. 2008.