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

Jian Liang, Frederick Park, and Hongkai Zhao
University of California, Irvine
March 21st, 2011

We present implicit surface reconstruction algorithms 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.

Key Words: principal component analysis (PCA), anisotropic Gaussian, distance function, edge indicator, normal, image segmentation, total variation, primal-dual algorithm.

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.

In this paper, our primary focus will be on surface reconstruction for PC. PC is 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. 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 [Amenta et al. 1998a; Amenta et al. 1998b; Boissonnat 1984; Edelsbrunner 1998; Edelsbrunner and Mucke 1994] often utilize Voronoi diagrams and Delaunay triangulations to determine connections among data points and subsequently construct triangulated surfaces by connecting adjacent vertices. For other explicit representations such as parametric surfaces, we refer the reader to [Piegl and Tiller 1996; Rogers 2003]. 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 [Rogers 2001; Hoppe et al. 1992; Zhao et al. 1998; Zhao et al. 2001; Lempitsky and Boykov 2007; Goldstein et al. 2009; Bae et al. 2009], 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 [Carr et al. 1997; Carr et al. 2001; Morse et al. 2001; Ohtake et al. 2003; Wendland 2002]. We also remark that, there are some recent fast and convex optimization techniques for implicit representations in the context of PC, see [Ye et al. 2010; Goldstein and Osher 2009; Goldstein et al. 2009; Zhu and Chan 2008; Chambolle 2004; Bae et al. 2009].

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 (1) all data points are used as vertices of the triangulated surfaces, (2) connection relation among data points are 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 flexible to deal 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 to some extent 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, and (2) a good edge indicator is needed. In this

The authors acknowledge ARO/MURI grant W911NF-07-1-0185 as well as NSF grant DMS0811254 for support with this work. Authors' addresses: Jian Liang, Frederick Park, and Hongkai Zhao, Department of Mathematics, University of California, Irvine, 340 Rowland Hall, CA, 92697-3875. email: jianl@uci.edu; fepark@uci.edu; see http://www.math.uci.edu/~zhao/

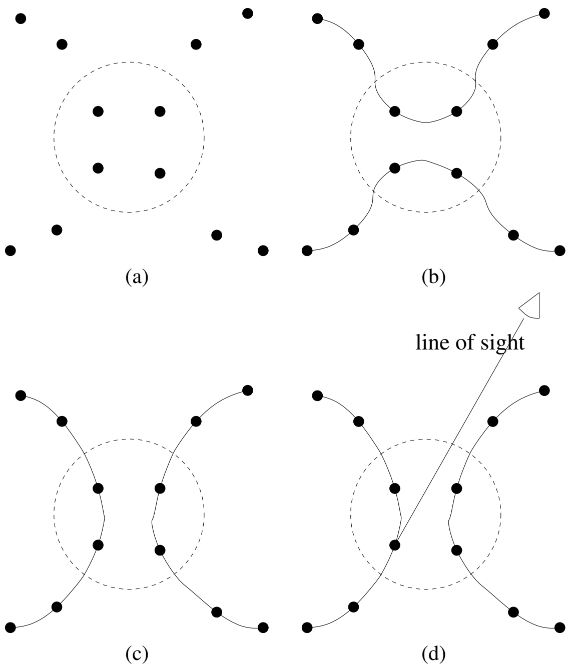


Fig. 1. Ambiguous PC reconstruction and alleviation by utilizing volumetric information.

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 to provide 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 a few variational models for implicit surface reconstruction and also talk about some binary image segmentation models, both of which motivate our work. We will then explain our methodology for the construction of the initial image and edge indicator 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. VARIATIONAL MODELS FOR IMPLICIT SURFACE RECONSTRUCTION AND IMAGE SEGMENTATION

One of the earliest variational models for implicit surface reconstruction was introduced by Zhao et al. in [Zhao et al. 1998; Zhao et al. 2001] where the authors propose the following weighted minimal surface model:

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

Here, $S = \{\vec{x}_i\}_{i=1,2,...}$ denotes the data set, and $d(\vec{x}) = \text{dist}(\vec{x}, S)$ 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:

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

where $\vec{n}$ 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 $\nabla d(\vec{x}) \cdot \vec{n}$ and the weighted surface tension $d(\vec{x})\kappa$ 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 [Zhao et al. 1998; Zhao et al. 2001] 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. Distance contour as an approximate offset of the true shape 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 minimum even if starting very close to the PC data. An explicit example of this can be seen in Figure 2 where in (a) and (b), a point cloud and it's distance function are shown. The distance contours for ϵ = 0.05 and 0.09 are seen in (c) and (d) respectively. If ϵ is chosen too small, the initial surface using the distance contour is not a closed surface. On the other hand, if ϵ is chosen too large, the local concave feature cannot be detected by the distance contour. These issues are also related to using 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. 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 Figure 2 (e) and (f) are the reconstruction results by the fast tagging algorithm in [Zhao et al. 2001] and the minimal surface model starting from the $\epsilon = 0.09$ contour of the distance function as the initial guess; the local concave feature cannot be felt by the minimal surface model and gets stuck at an unwanted local minimum.

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 [Caselles et al. 1997; Kass et al. 1987]. 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

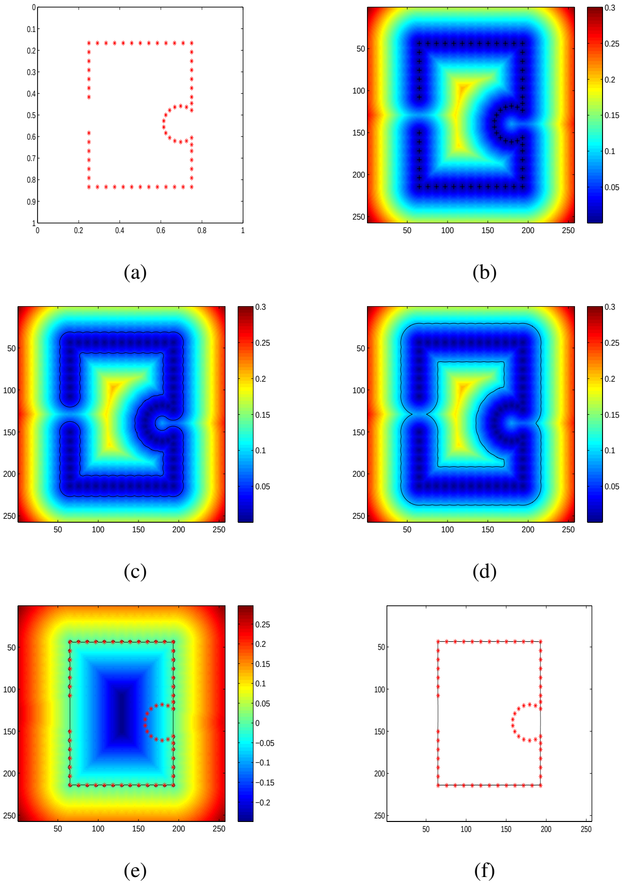


Fig. 2. Inability of the distance contour to capture fine feature in non-uniform PC data. (a) point cloud, (b) distance function, (c) ϵ = 0.05 contour, (d) ϵ = 0.09 contour, (e) fast tagging, (f) minimal surface reconstruction.

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.

One of the first convex binary image segmentation models is the following TVG-L1 model proposed in [Bresson et al. 2007],

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

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) [Giusti 1984; Federer 1969]. The second term is a volumetric image fitting term in which f is the given image. In particular, when $g(x) = 1$ and f is a binary image, it was shown in [Chan et al. 2004] that a global minimizer Σ of the non-convex variational problem (for binary image defined by Ω):

$$\min_\Sigma |\partial\Sigma| + \lambda|\Sigma \triangle \Omega|, \tag{4}$$

where $\triangle$ 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-L^1 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 a 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 is the following CVG model proposed in [Bresson et al. 2007],

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

Here, $g(x)$ and $f(x)$ is exactly as in the TV-L1 setting. c_1 and c_2 are two constants. The above model is a convex constrained version of the original model by Chan and Vese [Chan and Vese 2001]. If $g(x) = 1$, it is the same as the piecewise constant Mumford-Shah model. Again it was shown in [Chan et al. 2004] that to find the best approximation of an image $f(x)$ by two-valued piecewise constant functions,

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

which has the following variational formulation,

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

where $\Sigma \subset D$ is a region contained in the image domain and $|\partial\Sigma|$ denotes the measure of the boundary of Σ, it is equivalent to finding $(c_1, c_2, u(x))$ to minimize

$$\min_{c_1, c_2} \min_{0\le u\le 1} \int_D |\nabla u|dx + \lambda\left[\int_D (f - c_1)^2 - (f - c_2)^2\right] u(x)dx.$$

In other words, if $(c_1, c_2, u(x))$ is a solution to the above formulation, then for any $\mu \in (0, 1)$, the triplet, $(c_1, c_2, u(x) \ge \mu)$ is a global minimizer for (7). Again the first term in the CVG model (5) is the same as the first term in the TVG-L1 model (3) which is like 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 using the numerical algorithm described in Section 4. Both of these two models produce successful and accurate results. In comparison, TVG-L1 depends on the initial image f more and only locally, i.e., at each point only the sign of $f(x) - u(x)$ matters as can be seen from the Euler-Lagrange equation. If the data is clean and the initial image f gives a good binary approximation of interior and exterior regions, then TVG-L1 provides more accurate solutions. 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 in the data or to a rough initial image f. Numerical tests will be shown in Section 5 to demonstrate both models for various types of data.

Our work is also motivated by the recent work [Ye et al. 2010], in which the authors adapt the TV based segmentation models, i.e., TVG and CVG models proposed in [Bresson et al. 2007] 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 [Ye et al. 2010], 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:

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

for some $p > 0$. The foremost idea in [Ye et al. 2010] 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:

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

Then they propose to use the CVG segmentation model from [Bresson et al. 2007]:

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

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 [Lempitsky and Boykov 2007] along with some TV based iterative refinement by way of the Bregman iteration found in [Osher et al. 2005]. However, the above work suffers the intrinsic problems of using the distance function to the data set as the edge indicator. These problems will be discussed in details in Section 3. 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.

3. CONSTRUCTION OF THE INITIAL IMAGE AND THE EDGE INDICATOR FUNCTION

As we mentioned before, the major difference between point cloud problems and classical image processing problems is that there is no initial image sampled on a regular grid for the point cloud to begin with. 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. The basic tool in our approach is Principal Component Analysis (PCA) using neighboring points to extract more accurate local geometric information.

Given a PC consisting of points, $D = \{x_i\}_{i=1}^n$, the PCA at a given data point $p \in D$ is typically computed by first determining the *K-nearest neighbors* (KNN), $x_k \in D, k = 1, 2, \cdots K$, of p. Then form the covariance matrix by

$$P = \sum_{k=1}^{K} (x_k - \overline{p})^T (x_k - \overline{p}) \tag{11}$$

where $\overline{p}$ is the centroid of the KNN, $\overline{p} = \frac{1}{K}\sum_{k=1}^{K} x_k$. Let $0 \le \lambda_1 \le \lambda_2 \le \lambda_3$ be the three eigenvalues and v_1, v_2, v_3 be the three corresponding normalized eigenvectors. PCA reveals the local linear structure from the KNN of p. In particular v_1 gives the surface normal approximation and the relative size of λ_1, λ_2, and λ_3 indicates how curved the surface is in different directions. Figure 3 gives a 2D illustration. See [Pauly et al. 2002; Digne et al. 2011] for more characterization of local geometry in terms of the eigenvalues of the covariance matrix.

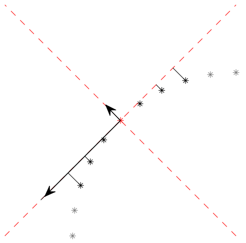


Fig. 3. Local coordinates using PCA.

3.1 The 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 D, we denote $p(x)$ as the data point that is closest to x. Then we define the inner product field

$$f(x) = v(x) \cdot n(p(x)), \quad v(x) = \frac{x - p}{|x - p|} \tag{12}$$

as the initial image, where $n(p(x))$ is the outward normal at $p(x)$. Figure 4 illustrates how $v(x), p(x)$, and $n(p(x))$ 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 $x - p(x)$ for the green and red points respectively.

In particular, $-1 \le f(x) \le 1$ is positive when x is outside and is negative when x is inside. The outward normal is used at

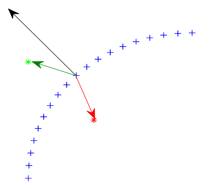


Fig. 4. Key idea of the inner product field.

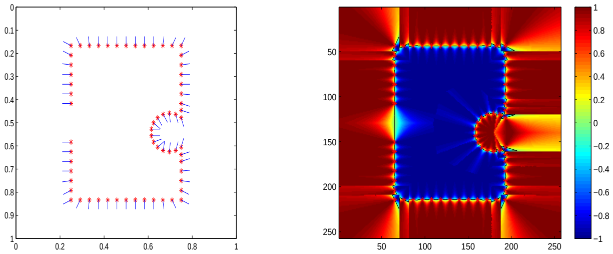


Fig. 5. Consistent normals and inner product field. The image on the left is an example of a 2D point cloud with consistent outward normal estimation, the right one is the inner product field $f(x)$ defined by (12). The black curve is the 0-level set of the inner product field.

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 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. Although PCA gives pretty robust normal estimation, it does not specify a consistent outward or inward normal estimation. For consistent normal estimation, the easiest way is to use line of sight information at each data point. For pure PC data, we first use PCA to estimate the normal at each data point and then use the simple method proposed in [Castillo and Zhao 2009] 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. To find closest point information $p(x)$, i.e., given a point $x \in D$ find the data point $p \in S$ which is closest to x, we use the fast sweeping method [Zhao et al. 1998; Zhao et al. 2001; Tsai 2002; Luo et al. 2010; Bak et al. 2010]. The fast sweeping method is an efficient iterative method that can compute the distance function as well as the closest point information on a rectangular grid in 4 iterations for 2D data and 8 iterations for 3D. Figure 5 shows an example of the inner product field. In practice, even when the data is noisy or the normal estimation is not that accurate, the inner product field 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 both the TVG-L1 model (3) and the CVG (5) model, the final implicit surfaces can be regarded as regularized (by the first term in both models) or modified segmentation based on $f(x)$.

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 6 illustrates the boundary extension and hole-filling scenario in 2D. 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. 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, i.e., the zero level set of the inner product field f may have corners and/or edges, the first regularization term in both models will smooth out these corners and/or edges. 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. We will show results for both hole filling and open surfaces in Section 5.

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.

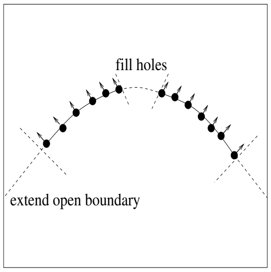


Fig. 6. Surface extension and hole-filling.

3.2 Edge indicator

A good edge indicator is crucial for the performance and quality of the TVG-L1 and CVG models. A straightforward edge indicator function is the distance function to the data set. The distance function is easy to compute and gives a good indicator far away from the data set. 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.

Here, 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. At a data point p_i, we first use PCA (11) on its KNN to obtain the eigenvalues $0 \le \lambda_1 \le \lambda_2 \le \lambda_3$ and the corresponding normalized eigenvectors v_1, v_2, v_3 respectively. We establish a local coordinate system (x, y, z) aligned with the three eigenvectors and centered at p_i. Then we generate the following anisotropic Gaussian kernel at p_i:

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

where (x_1, x_2, x_3) is the local coordinate for a grid point $x \in D$ under the local coordinate. In the above expression, the term $\exp\{\beta(p_i)\}$ is to characterize uncertainty information at p_i, which is useful in dealing with outliers. 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 [Fleishman et al. 2005]. 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 point cloud 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 or outliers $\beta(p_i)$ will be relatively small. Figure 7 (c) provides some demonstration in 2D, where $\beta(p_i) = -5\frac{\lambda_1}{\lambda_1+\lambda_2}$ and $s = 4$. 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 [Medioni et al. 2000] or the one proposed in [Pauly et al. 2004], which is more expensive to compute, can be used here too. For our variational models (3) and (5), we invert the likelihood function

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

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

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

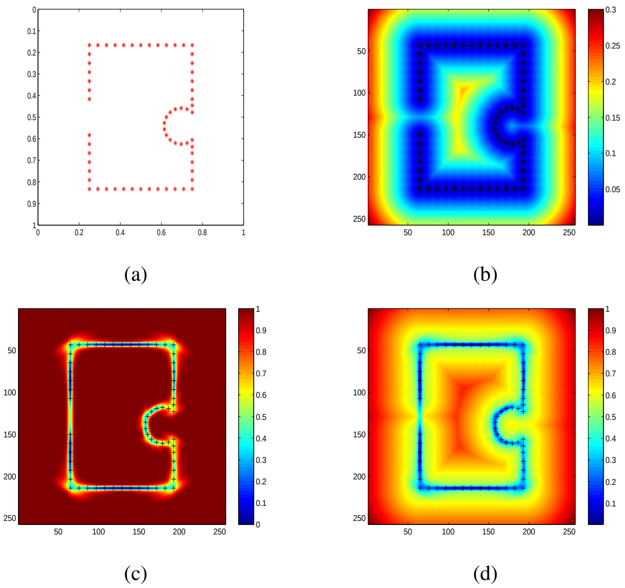
Some useful features of our edge indicator function include

—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 non-uniform and noisy data sets. An interesting observation is that the behavior of our constructed edge indicator 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 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. So in practice we combine these two. We normalize the distance function to $[0, 1]$ and average it with the inverse sum of the anisotropic Gaussians defined in (15). Another advantage of combining these two is that it is only necessary to evaluate each Gaussian on grids in a small local region (a narrow band) of each data point which in turn saves computation cost significantly. As a result, our edge indicator function behaves more like the inverse sum of anisotropic Gaussians (15) near the data set and behaves more like the distance function away from it. This behavior of our edge indicator function, which gives both strong global indication and sharp localization near the data set, also helps the optimization algorithms for both convexified segmentation models described below to converge fast.

Figure 7 gives a clear demonstration of the behaviors of different edge indicator functions. Figure 7 (b) shows the constantly sloped distance function which is not so sharp near the data set. Figure 7 (c) shows the inverse sum of anisotropic Gaussians (15) which is sharp near the data set but is almost a constant away from the data set. Figure 7 (d) shows a combination which inherits advantages of both.



(a)

(b)

(c)

(d)

Fig. 7. Proposed edge indicator function on concave data set. (a) point cloud, (b) distance function, (c) anisotropic Gaussian, (d) edge indicator (average of the scaled distance and anisotropic Gaussians)

4. NUMERICAL ALGORITHMS

4.1 Generating the initial image and edge indicator function

To generate the initial image we

(1) Compute the distance function and find the closest point map for the whole grid using fast sweeping method with locking [Zhao 2005; Bak et al. 2010; Luo et al. 2010].

(2) Do local PCA based on KNN to compute the normal at each point.

(3) Generate the normalized inner product field.

These are all efficient algorithms with linear complexity.

The edge indicator function can be computed directly from the local PCA and distance function obtained in the above step. We limit Gaussian evaluation only near each data point.

4.2 Image Segmentation Algorithms: TVG-L1 and CVG

Here, we outline two efficient algorithms based on the dual formulation of the TVG-L1 and CVG models. There are many recent methods for solving these models, most notably [Zhu and Chan 2008; Goldstein and Osher 2009; Bae et al. 2009; Chambolle 2004; Bresson et al. 2007]. We found that different methods perform comparably in our application mainly due to the fact that we have a pretty accurate initial image and a sharp edge indicator to start with. For ease of use, simple parameter tuning, and guaranteed convergence, we utilize the fixed point method found in [Bresson et al. 2007] based on the dual formulation of the TVG-L1 and CVG models. We further reduce the computational cost for closed surfaces by restricting our image segmentation in a neighborhood around the data points using the distance contour that is far enough to close the gap between neighboring points, see Figures 12–16. For open surfaces, i.e. surfaces with open boundary, we utilize a bounding box and do the segmentation in the whole box, see Figures 17–18. As discussed in Section 3.1 our method can fill holes and extend open surfaces smoothly along the tangential directions. We give a more detailed description of the algorithms below.

4.2.1 *TVG-L1.* For numerical implementation of the TVG-L1 model (3), we consider the convex regularized version suggested in [Bresson et al. 2007] which has the following formulation:

$$\min_{u,v} \int_\Omega \Big(g(x)|\nabla u| + \lambda|v| + \frac{1}{2\theta}|u+v-f|^2 \Big) dx \tag{16}$$

where f is the initial image from (12) and $g(x)$ is the PC edge detector:

$$g(x) = \frac{d(x) + g_0(x)}{2}, \tag{17}$$

with $d(x)$ being the scaled distance function, and $g_0(x)$ the scaled inverse sum of the Gaussians found in (15). It is assumed that the above model (16) converges to the original TVG-L1 model for θ chosen small enough, see [Aujol et al. 2006]. This regularized energy can be minimized by alternatingly minimizing over each single variable u and v in the coupled energies:

(1) for v fixed,

$$\min_{u} \int_\Omega \Big(g(x)|\nabla u| + \frac{1}{2\theta}|u+v-f|^2 \Big)\, dx \tag{18}$$

(2) for u fixed,

$$\min_{v} \int_\Omega \Big(\frac{1}{2\theta}|u+v-f|^2 + \lambda|v| \Big)\, dx. \tag{19}$$

It is shown in [Carter 2001; Chambolle 2004] that the minimizer u for (18) can be found in terms of the dual variable p by the update formula

$$u = f - v - \theta\, \nabla \cdot p \tag{20}$$

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

$$g(x)A(p) - |A(p)|p = 0 \tag{21}$$

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 [Chambolle 2004] which was also used for minimizing the TVG-L1 model in [Bresson et al. 2007]:
$p^0 = 0$,

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

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

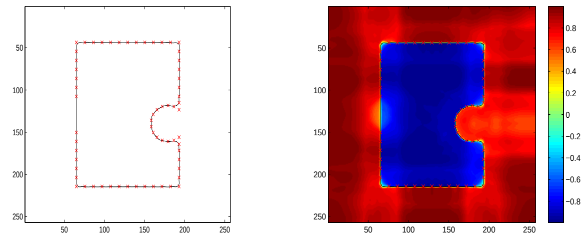
Now, v can easily be solved for from the shrinkage scheme below:

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

This scheme was initially proposed in [Chambolle et al. 1998; Donoho and Johnstone 1995] for wavelet shrinkage, but was later adapted to the TV-L1 and TVG-L1 minimization problems in [Aujol et al. 2006; Bresson et al. 2007] respectively.

Solving for the final solution amounts to alternating between the two equations (22) and (23) using the initial conditions $u_0 = 0 = v_0$ and $p_0 = \langle p_0^1, p_0^2, p_0^3 \rangle \equiv \vec{0}$ with time step taken as $\tau = 1/8$ for 2D examples and $\tau = 1/16$ for 3D ones. Discretization follows in exactly the same way as discussed in [Chambolle 2004; Bresson et al. 2007]. The stopping criteria used for the alternating minimization is $\max\left(|u_{n+1} - u_n|, |v_{n+1} - v_n|\right) \leq$ Tol, for some prescribed tolerance ‘Tol’. Lastly, the final segmented region Σ is obtained by simply taking the 1/2 upper level set of the minimizer $u(x)$; strictly speaking $\Sigma = \left\{x|\, u(x) \geq \frac{1}{2}\right\}$ (assuming the initial image is scaled between [0,1]). The boundary of the surface reconstruction to the PC is then realized by taking the $\mu = 1/2$ level set of $u(x)$. i.e. $\partial\Sigma = \left\{x|\, u(x) = \frac{1}{2}\right\}$.

The reconstructed surface of the above 2D example is shown in Figure 8 (a), using 100 iterations with $\lambda = 0.01$ and $\theta = 0.1$.



(a)

(b)

Fig. 8. Gap connection and concave feature reconstruction. (a) reconstructed surface using TVG-L1, (b) reconstructed image.

4.2.2 *CVG.* For the numerical implementation of the CVG model (5) for PC data, if the constants c_1, c_2 are fixed, we consider the following convex unconstrained formulation:

$$\min_u \int_\Omega \Big[g(x)|\nabla u(x)| + \lambda \underbrace{\left[(f-c_1)^2 - (f-c_2)^2\right]}_{\equiv R(x,c_1,c_2)} u(x) + \alpha\nu(u(x)) \Big] dx \tag{24}$$

where again $g(x)$ is the PC edge detector (17) and $f(x)$ the initial image (12). The function $\nu(\zeta) \equiv \max\{0, 2|\zeta - 1/2| - 1\}$ is an exact penalty function on u. The above unconstrained model (24) and the CVG model (5) have the same set of minimizers provided that α is chosen large enough in relation to λ. A criteria here would be $\alpha > \frac{\lambda}{2}\|R(x,c_1,c_2)\|_{L^\infty}$ as shown in [Chan et al. 2004; Bresson et al. 2007].

In order to take advantage of fast algorithms based on the dual formulation of the TVG norm, we utilize a regularized version of the CVG model (much as in the TVG-L1 case) found in [Bresson et al. 2007] as follows:

$$\min_{u,v} \int_\Omega \Big[g(x)|\nabla u| + \frac{1}{2\theta}|u-v|^2 + \lambda R(x,c_1,c_2)v + \alpha\nu(v) \Big] dx. \tag{25}$$

The energy (25) can be minimized by an alternating minimization method in the proceeding manner:

(1) for v fixed,

$$\min_u \int_\Omega \Big(g(x)|\nabla u| + \frac{1}{2\theta}|u-v|^2 \Big) dx \tag{26}$$

(2) for u fixed,

$$\min_v \int_\Omega \Big(\frac{1}{2\theta}|u-v|^2 + \lambda R(x,c_1,c_2)v + \alpha\nu(v) \Big) dx. \tag{27}$$

The sub-minimization problem (26) in u coincides with the sub-minimization problem (18) of the TVG-L1 model if we simply replace $f - v$ by the new variable v in (18). Thus, the same method for obtaining a minimizer u by way of solving the dual formulation of this sub-problem can be employed. Following in this manner, the update for u in terms of the dual variable p with regards to the dual formulation of (26) is given by $u = v - \theta\nabla \cdot p$. Now, if we set $A(p) = \nabla(\nabla \cdot p - v/\theta)$ then the dual variable p satisfies (much in the same way as in (21)) the optimality equation $g(x)A(p) - |A(p)|p = 0$. Thus, the dual variable p satisfying the optimality equation can be solved for by the exact same fixed point scheme as in the TVG-L1 model explicitly given above in (22).

Solving for v in (27) amounts to the following threshold scheme discussed in [Chan et al. 2004; Bresson et al. 2007]

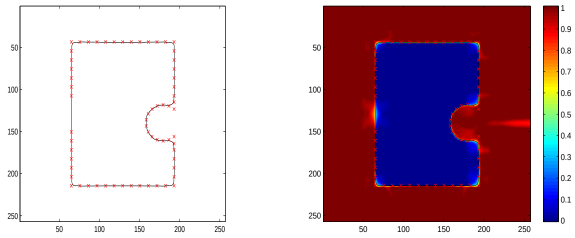
$$v = \min\{\max\{u(x) - \theta\lambda R(x,c_1,c_2), 0\}, 1\}. \tag{28}$$

Finally, the constants c_1 and c_2 are updated by the formulas:

$$c_1 = \frac{\int_\Sigma f(x)\,dx}{|\Sigma|}, \quad c_2 = \frac{\int_{\Omega\setminus\Sigma} f(x)\,dx}{|\Omega\setminus\Sigma|}. \tag{29}$$

For the CVG alternating minimization scheme between the fixed point method (22) and the threshold scheme (28), we use the same initial conditions, stopping criteria, discretization, and final thresholding on the minimizer u of (24) as in the TVG-L1 model outlined above to get the final segmented region Σ and reconstructed surface $\partial\Sigma$ of the PC. The only difference here is that during the alternating minimization between the fixed point and thresholding, we update the constants c_1 and c_2 every 10 iterations via the formula (29).

The reconstructed surface of the above 2D example is shown in Figure 9 (b), using 100 iterations with $\lambda = 0.05$ and $\theta = 0.1$.



(a) (b)

Fig. 9. Gap connection and concave feature reconstruction. (a) reconstructed surface using CVG, (b) reconstructed image.

4.3 Postprocessing

The above image segmentation models tend to produce a nearly binary image (characteristic function) with rapid transition near the sharp edge indicator. Although the segmentation contains the implicit surface information, it does not render a pleasing visual effect using standard visualization tools for implicit surfaces. We postprocess the segmentation by a reinitialization process [Peng et al. 1999] to transform the characteristic function to a signed distance function which results in a slightly smoothed surface without affecting the accuracy.

4.4 Complexity Analysis

We use a fast sweeping method to compute the distance function and closest point information. In order to additionally speed up the computation, we also use a simple locking technique for the fast sweeping method which will avoid unnecessary updates during each sweeping. We refer the readers to [Luo et al. 2010; Bak et al. 2010] for more details. The computation complexity for distance function and closest point information is $O(M)$ where M is the grid size. For computing the inner product field, we only need to sweep the whole grid once.

To find the KNN for each data point, we use a very efficient package called KDTREE2. The computation time is $O(KN\log N)$ where N is the total number of data points and K is the number of neighbors we use. We compute the PCA for each data point to form a local coordinate and we use the LAPACK to solve the eigenvalue problem, the total computation time is $O(N)$. The time for computing the anisotropic Gaussian for each data point is $O(N)$. A useful fact is that the anisotropic Gaussian decays so fast that we do not need to compute it on the whole grid, only a small portion of the grid near the data points is sufficient. In our implementation, we compute the anisotropic Gaussian values $G(p_i)$ 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.

The complexity for reinitialization is proportional to the grid size and number of iterations. Since our solution surface from TVG-L1 or CVG segmentation is close to the correct surface, only a few iterations are needed, thus, being very fast. More details about the reinitialization algorithm can be found in [Peng et al. 1999].

The overall complexity of our implicit surface reconstruction from point cloud is $O(N \log N + M)$. Since it is based on robust convexified segmentation algorithms it can handle noisy and nonuniform PC, PC with holes or open boundary as can be seen from the various numerical experiments found in the next section.

5. NUMERICAL EXPERIMENTS

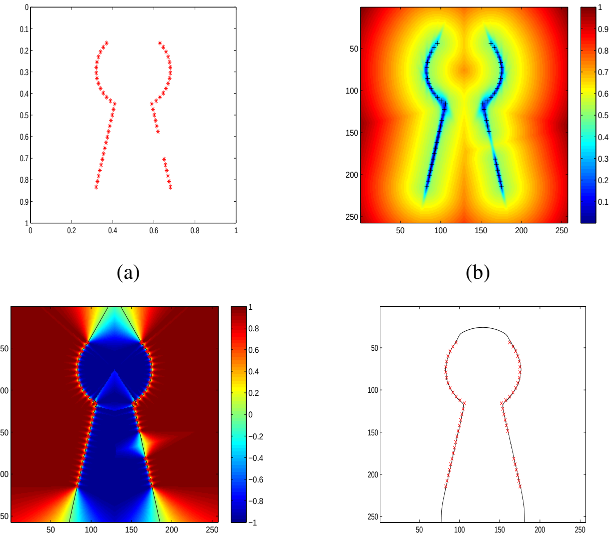
In this section, we present some 2D and 3D experiments to validate the effectiveness of the proposed model. For the segmentation step, which creates the boundary of the reconstructed surface, we utilize the two models discussed in Section 4.2. In general they both produce accurate and comparable results. However, there are some differences in their respective reconstructions due to their different data fidelity terms, namely, the fidelity term in the TVG-L1 model is more local in nature while the fidelity term in the CVG model is more global. Hence TVG-L1 tends to preserve singularities, e.g. corners and edges, sharper while the CVG model is more robust to noise. In particular, the TVG-L1 model tends to depend a bit more on the consistency of the normals while CVG one only requires that most of the normals are consistent. Choosing which model may depend on the particular point cloud and we provide some guidelines at the end of this section.

All experiments are performed on a Dell Inspiron 1525 notebook with an Intel(R) Core(TM) 2 Duo CPU T8100 @ 2.10 GHz processor and 4.00 GB RAM memory. Our algorithm is not particularly sensitive to parameters. All the parameters are easily chosen and some guidelines are provided later in this section.

5.1 2D Experiments

We use a 257×257 grid and the 5 nearest neighbors in PCA for all of the 2D experiments. The parameters used for all the 2D experiments are the following: $\beta(p_i) = -5\frac{\lambda_1}{\lambda_1+\lambda_2}$, $s = 2$, and $\alpha = 10$.

Figure 10 illustrates an example of a PC with both an open boundary and 2 portions having missing data points. Figure 10 (b) shows that our edge indicator not only provides strong connection in the linear missing part but also extends the open boundary in a correct and natural way. In 10 (c) is the normalized inner product field whose zero level set is taken as our initial surface. One can see that there are some incorrectly identified regions. Nonetheless, one important and beneficial property of either the TVG-L1 or CVG model is that they can resolve these misidentified regions while keeping the correct ones near the boundary of the PC. Our reconstructed surface 10 (d) can properly fill in the linear missing part of the PC boundary while also providing a reasonable gap connection in the non-linear region containing missing points somewhere between a tangent extension and minimal curvature rather than just connecting it using a straight line. We use TVG-L1 for this example, using 300 iterations with $\lambda = 0.001$ and $\theta = 0.1$.

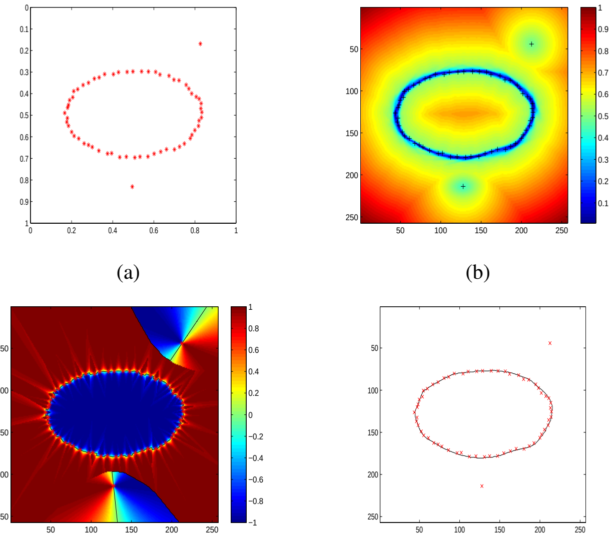


(a) (b)

(c) (d)

Fig. 10. Open Boundary Extension and Gap Connection. (a) point cloud, (b) edge indicator, (c) normalized inner product, (d) reconstructed surface by the proposed method using TVG-L1 segmentation.

In the next example we consider a PC in the shape of a noisy ellipse with two outlier points which can be seen in Figure 11 (a). Our edge indicator is shown in 11 (b) providing a strong indicator near the boundary of the ellipse while having much less indication near the outliers. The inner product field is observed in 11 (c) yielding a favorable initial surface with the possible exception being near the outlier points. The reconstructed surface viewed in 11 (d) provides a reasonably accurate surface for the noisy ellipse. Our edge indicator function automatically provides weaker indicator at the outlier points, thus, the outliers do not affect our reconstructed surface. We use the CVG model for the segmentation step in this example, using 500 iterations with $\lambda = 0.0001$ and $\theta = 0.1$.



(a) (b)

(c) (d)

Fig. 11. Reconstruction of a Noisy PC with Outliers. (a) point cloud, (b) edge indicator, (c) normalized inner product, (d) reconstructed surface by the proposed method using CVG segmentation.

5.2 3D Experiments

For the 3D examples, we utilize uniform grids whose grid size depends on both the size of a given point cloud and range of those data points. As mentioned before, when possible, we use a narrow band computational domain around the PC. 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.

5.2.1 *Data of significant size.* We consider the dragon and skeleton hand PC data sets which both have a significant number of points, large grid size, and complicated geometry detail. For these closed surfaces, we can restrict the computation to the grid points that are within a narrow band of the surface. In the reconstructed surfaces, all the details of the PC data can be observed, like the teeth of the dragon seen in Figure 12 and the fine feature thin bones near the wrist of the skeleton hand observed in Figure 13. Parameters for the dragon and hand are all the same: 50 iterations, $\lambda = 1$ and $\theta = 0.1$ for both the TVG-L1 and CVG models. Lastly, only 3 iterations of reinitialization are needed to achieve the final reconstructed surface. The Figures 12 and 13 show reconstructed surfaces using the TVG-L1 model of the dragon and skeleton hand respectively. Here, TVG-L1 is utilized in the segmentation step since the data set is well behaved in the sense of uniform sampling and low noise level.

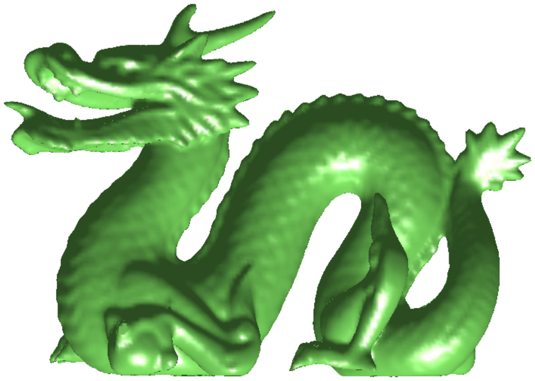


Fig. 12. Dragon PC surface reconstruction by the proposed method using TVG-L1 segmentation.

5.2.2 *Data with holes.* In both Figures 14 and 15, the left Figures 14 (a) and 15 (a) are the reconstructed surfaces and the right ones 14 (b) and 15 (b) are the reconstructed surfaces with the data points superimposed. For both of these examples, we use a narrow band computational domain. The holes, scratches, and missing parts in both PC's can be successfully filled as observed in Figure 14 (b) for the Stanford Bunny and in 15 (b) for the Knot. Parameters for the bunny with holes are: 50 iterations, $\lambda = 1$ and $\theta = 0.1$ for both the TVG-L1 and CVG models. Lastly, only 3 iterations of the reinitialization is needed to achieve the final reconstructed surface. Parameters for the knot PC with holes are: 50 iterations for both TVG-L1 and CVG models, $\lambda = 0.16$ and $\theta = 1$ in TVG-L1 model and $\lambda = 0.48$ and $\theta = 1$ in CVG model. For both models, once again, only 3 iterations of the reinitialization is needed to achieve the final reconstructed surface. In both Figure 14 and 15, the reconstructed surfaces are obtained using the TVG-L1 model in the segmentation step.

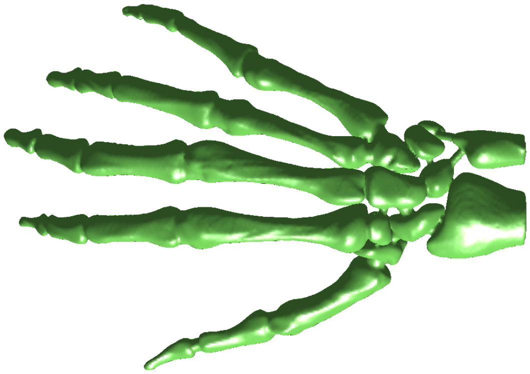
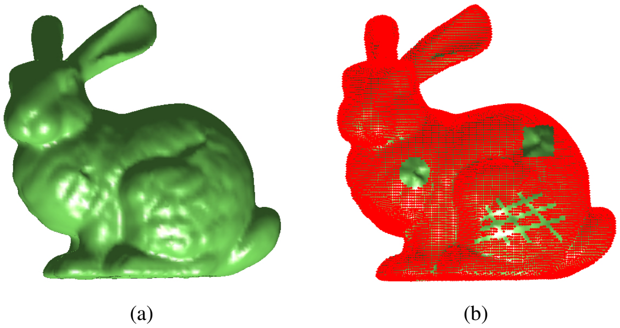
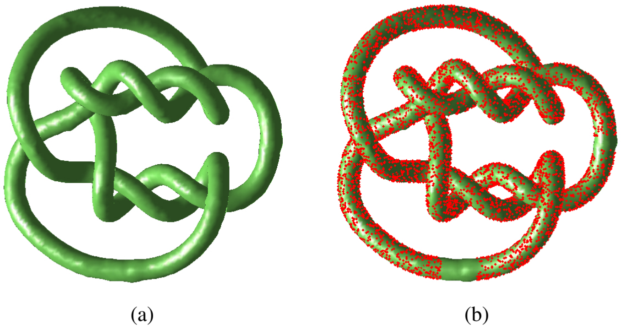


Fig. 13. Skeleton hand PC surface reconstruction by the proposed method using TVG-L1 segmentation.



(a) (b)

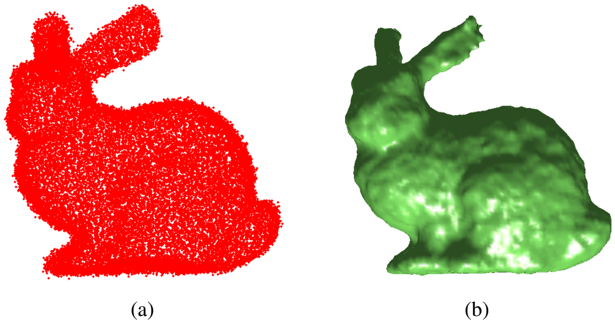
Fig. 14. Hole filling multiple regions. Bunny reconstruction by the proposed method using TVG-L1 segmentation.



(a) (b)

Fig. 15. Hole filling with difficult topology. Knot reconstruction by the proposed method using TVG-L1 segmentation.

5.2.3 *Noisy data.* In Figure 16 (a) a noisy bunny PC is observed. Here, we add Gaussian noise with 0 mean and variance of 0.5% of the diagonal of the bounding box to the clean bunny PC to obtain the observed one. The reconstructed surface is observed in 16 (b) where it is easily seen that most of the parts of the bunny are reconstructed quite well despite the noise with the possible exception being one of the ears. The reason here is that particular portion of the ear in question is relatively thin which can be construed as a fine feature, thus, the presence of noise affects it a great deal more. As mentioned in the beginning of this section, the TVG-L1 model is more local while the CVG model is more averaged (less local) and provides a smoother reconstructed surface. Hence, we utilize the CVG model for noisy data and the parameters for this particular experiment are 50 iterations with $\lambda = 0.4$ and $\theta = 1$. Lastly, 25 iterations of reinitialization are used. Again we only need to compute in a narrow band of the PC for this example.


(a)

(b)

Fig. 16. Noisy PC surface reconstruction by the proposed method using CVG segmentation.

5.2.4 *Data with open boundary.* Due to the open boundary, we cannot use the narrow band method in this case and will need to execute the computation in the bounding box. Nonetheless, an interesting benefit of our algorithm is that we can extend the boundary smoothly. We show that for the Venus and gargoyle data sets, we can reconstruct Venus's neck and extend the base of the gargoyle.

In the two examples of the Venus and the Gargoyle, their respective PC data sets are relatively dense, uniformly distributed, have low noise, and with few singularities, thus, the 0-level set of the normalized inner product field is already quite an accurate reconstructed surface. Hence, in this particular case, the computation of the edge indicator and the segmentation process can be optional. In both of the two examples shown in Figure 17 and 18, we omit the edge indicator computation and segmentation steps and only run 3 iterations of reinitialization to get smoothly reconstructed surfaces.

5.2.5 *LiDAR data.* In the last section we test our algorithm on challenging real LiDAR data. Figures 19 (a) and 20 (a) demonstrate two data sets that are segmented from a large LiDAR data set. One of the main challenges comes from non-uniformity and non-complete data due to occlusion or shadow. For example, our two data sets are all one-sided and also have some missing parts like the right broken (disconnected) leg. Our reconstructed surfaces show that our algorithm can deal with such difficult data quite reasonably. We can connect the broken leg, we can fit the one-sided data and we can capture features such as gaps between two legs. Since these data are one-sided, using anisotropic Gaussians as the edge indicator may extend the boundary undesirably. So we simply use the distance function as an edge indicator with the CVG model. Due to its local nature, the TVG-L1 model is not optimum for these LiDAR data sets and cannot provide optimum results. Parameters for LiDAR1 are: 100 iterations of CVG model with $\lambda = 0.06$ and $\theta = 1$ and then 10 iterations of reinitialization. Parameters for LiDAR2 (with broken leg) are: 100 iterations of CVG model with $\lambda = 0.11$ and $\theta = 1$ and then 10 iterations of reinitialization.

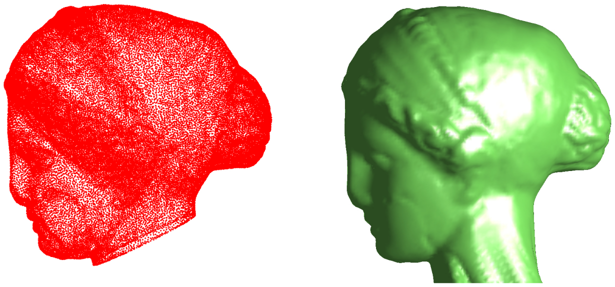


Fig. 17. Open boundary reconstruction and boundary extension of the Venus PC.

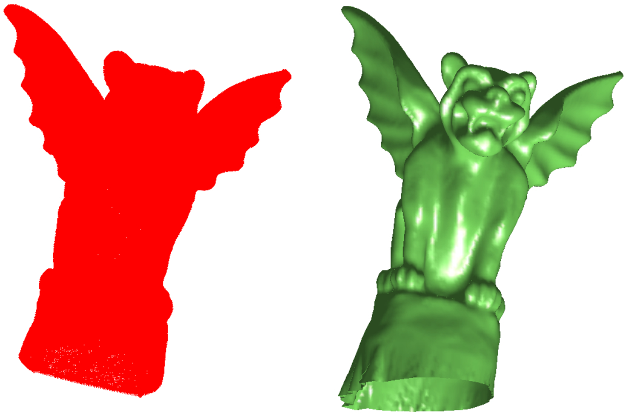


Fig. 18. Open boundary reconstruction and boundary extension of the Gargoyle PC.

5.3 Summary: Computation Times and Guidelines

The CPU running time (in seconds) for all 3D experiments are shown in the following table.

	pt size	grid size	ini proc	segmentation	reini	total
dragon	435,545	221×157×99	24.96	8.67(TVG-L1)	2.47	36.10
				16.03(CVG)		43.46
hand	327,323	301×211×103	38.64	14.60(TVG-L1)	4.62	57.86
				28.02(CVG)		71.28
bunny(hole)	35,132	129×127×101	7.05	7.80(TVG-L1)	1.17	16.02
				12.57(CVG)		20.79
knot	9,754	127×129×63	4.63	11.12(TVG-L1)	0.79	16.54
				16.19(CVG)		21.61
bunny(noisy)	35,947	127×129×63	6.94	10.78(CVG)	8.89	26.61
Venus	44,992	125×129×91	6.75		1.05	7.80
gargoyle	91,279	201×145×135	16.58		2.99	19.57
LiDAR1	1,006	39×39×129	0.42	12.34(CVG)	0.43	13.19
LiDAR2	1,009	37×27×129	0.31	8.49(CVG)	0.32	9.12

'pt size' is the number of data points, 'ini proc' is sum of the initial processing time (KNN, PCA, fast sweeping method with locking to compute distance function, closest point information, edge indicator, and normalized inner product field), 'segmentation' is the time

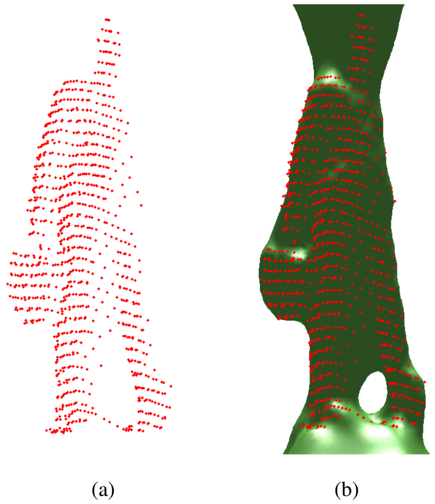


Fig. 19. LiDAR1 data set surface reconstruction by the proposed method using CVG segmentation.

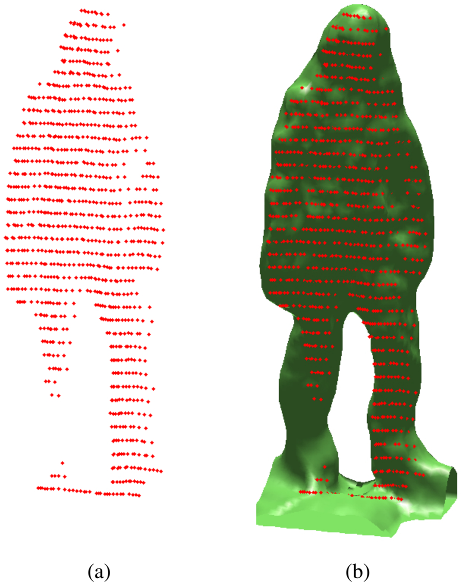


Fig. 20. LiDAR2 data set surface reconstruction by the proposed method using CVG segmentation.

for TVG-L1 or CVG minimization, ‘reini’ is the time for reinitialization, ‘total’ is the total time of computation.

Outlined below are some guidelines from our experience of which model to choose depending on the given PC data:

(1) If the data is dense enough, the 0 level set of the normalized inner product field is already an agreeable surface reconstruction; several iterations of reinitialization will provide a smooth final surface.

(2) For closed boundary data, we can use the narrow band computational domain. For data with an open boundary, we have to use the bounding box.

(3) If the data is dense and the noise level is small, the TVG-L1 model is suggested in the segmentation step. Moreover, it is a bit computationally faster than the CVG model.

(4) For a data set that is very noisy or with unreliable normal estimation, the CVG segmentation model is preferred.

6. CONCLUSIONS

In this paper, we proposed robust and efficient algorithms 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 noise, non-uniform sampling, holes, and open boundary.

ACKNOWLEDGMENTS

The authors would like to thank Edward Castillo for graciously providing us with his code for computing surface normals to PC data.

REFERENCES

ALMGREN, F., TAYLOR, J. E., AND WANG, L. 1993. Curvature-driven flows: a variational approach. *SIAM J. Control Optim. 31,* 2, 387–438.

AMENTA, N., BERN, M., AND EPPSTEIN, D. 1998b. The crust and the β-skeleton: combinatorial curve reconstruction. *Graphical Models and Image Processing 60,* 2, 125–135.

AMENTA, N., BERN, M., AND KAMVYSSELIS, M. 1998a. A new voronoi-based surface reconstruction algorithm. *Proc. SIGGRAPH 98*, 415–421.

AUJOL, J. F., GILBOA, G., CHAN, T. F., AND OSHER, S. 2006. Structure-texture image decomposition - modeling, algorithms, and parameter selection. *IJCV 67,* 1, 111–136.

BAE, E., YUAN, J., AND TAI, X. C. 2009. Global minimization for continuous multiphase partitioning problems using a dual approach. *UCLA CAM Report 09,* 75.

BAK, S., MCLAUGHLIN, J., AND RENZI, D. 2010. Some improvements for the fast sweeping method. *SIAM J. Sci. Comput. 32,* 5, 2853–2874.

BOISSONNAT, J. 1984. Geometric structures for three dimensional shape reconstruction. *ACM Trans. Graphics 3*, 266–286.

BOYKOV, Y. AND KOLMOGOROV, V. 2003. Computing geodesics and minimal surfaces via graph cuts. *ICCV03(I)*, 26.

BRESSON, X. AND CHAN, T. 2007. Active contours based on chambolle’s mean curvature motion. *IEEE International Conference on Image Processing*, 33–36.

BRESSON, X., ESEDOGLU, S., VANDERGHEYNST, P., THIRAN, J., AND OSHER, S. 2007. Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and Vision 28*, 151–167.

BURGER, M., OSHER, S., XU, J., AND GILBOA, G. 2005. Nonlinear inverse scale space methods for image restoration. In *Communications in Mathematical Sciences*. 2006.

CARR, J. C., BEATSON, R. K., CHERRIE, J. B., MITCHELL, T. J., FRIGHT, W. R., MCCALLUM, B. C., AND EVANS, T. R. 2001. Reconstruction and representation of 3d objects with radial basis functions. *SIGGRAPH 01 Conf. Proc., ACM SIGGRAPH*, 67–76.

CARR, J. C., FRIGHT, W. R., AND BEATSON, R. K. 1997. Surface interpolation with radial basis functions for medical imaging. *IEEE Transactions on Medical Imaging 16*, 96–107.

CARTER, J. 2001. Dual methods for total variation–based image restoration. *Ph.D. thesis, UCLA (Advisor: T. F. Chan).*

CASELLES, V., KIMMEL, R., AND SAPIRO, G. 1997. Geodesic active contours. *IJCV 22,* 1, 61–79.

CASTILLO, E. AND ZHAO, H. 2009. Point cloud segmentation via constrained nonlinear least squares surface normal estimates. *UCLA CAM Report 09,* 104.

CHAMBOLLE, A. 2004. An algorithm for total variation minimization and applications. *JMIV 20*, 89–97.

CHAMBOLLE, A., DE VORE, R. A., LEE, N., AND LUCIER, B. J. 1998. Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage. *IEEE Transcations on Image Processing 7,* 3, 319–335.

CHAN, T., ESEDOGLU, S., AND NIKOLOVA, M. 2004. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*.

CHAN, T., GOLUB, G., AND MULET, P. 1999. A nonlinear primal-dual method for total variation based image restoration. *SIAM J. Sci. Comp. 20*, 1964–1977.

CHAN, T. F. AND ESEDOGLU, S. 2004. Aspects of total variation regularized L^1 function approximation. *UCLA CAM Report 04,* 07.

CHAN, T. F. AND VESE, L. A. 2001. Active contours without edges. *IEEE Transactions on Image Processing 10*, 266–277.

COURANT, R., FRIEDRICHS, K., AND LEWY, H. 1967. On the partial dierence equations of mathematical physics. *IBM Journal*, 215–234.

CURLESS, B. AND LEVOY, M. 1996. A volumetric method for building complex models from range images. *SIGGRAPH96 Proceedings*, 303–312.

DIGNE, J., MOREL, J. M., SOUZANI, C. M., AND LARTIGUE, C. 2011. Scale space meshing of raw data point sets. *Computer Graphics Forum 00,* 0, 113.

DONOHO, D. AND JOHNSTONE, M. 1995. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association 90,* 432, 1200–1224.

DUGUET, F., DURAND, F., AND DRETTAKIS, G. 2004. Robust higher-order filtering of points. *Tech. Report, INRIA*.

EDELSBRUNNER, H. 1998. Shape reconstruction with delaunay complex. *Springer-Verlag, Proc. of LATIN98: Theoretical Informatics 1380*, 119–132.

EDELSBRUNNER, H. AND MUCKE, E. P. 1994. Three dimensional α shapes. *ACM Trans. Graphics 13*, 43–72.

FEDERER, H. 1969. *Geometric measure theory*. New York: Springer.

FLEISHMAN, S., COHEN-OR, D., AND SILVA, C. T. 2005. Robust moving least-squares fitting with sharp features. *ACM Trans. on Graphics (TOG), Proc. of ACM SIGGRAPH 24,* 3.

FLEISHMAN, S., DRORI, I., AND COHEN-OR, D. 2003. Bilateral mesh denoising. *ACM TOG (Proceedings of SIGGRAPH '03) 22,* 3, 950–953.

GIUSTI, E. 1984. *Minimal surfaces and functions of bounded variation*. Vol. 80.

GOLDSTEIN, T., BRESSON, X., AND OSHER, S. 2009. Geometric applications of the split bregman method: Segmentation and surface reconstruction. *UCLA CAM Report 09,* 06.

GOLDSTEIN, T. AND OSHER, S. 2009. The split bregman method for l1 regularized problems. *SIAM Journal on Imaging Sciences 2,* 2, 323–343.

HOPPE, H., DEROSE, T., DUCHAMP, T., MCDONALD, J., AND STUETZLE, W. 1992. Surface reconstruction from unorganized points. *Computer Graphics 26,* 2, 71–78.

KASS, M., WITKIN, A., AND TERZOPOULOS, D. 1987. Snakes: Active contour models. *Int. Journal of Comp. Vision 1,* 4, 321–331.

LEMPITSKY, V. AND BOYKOV, Y. 2007. Global optimization for shape fitting. *CVPR*.

LUO, S., GUIBAS, L., AND ZHAO, H. 2010. Euclidean skeletons using closest points. *Inverse Problems and Imaging 30,* 1, 95–113.

MEDEROS, B., VELHO, L., AND DE FIGUEIREDO, L. H. 2003. Point cloud denoising. *Proc. to Geom. Design and Computing '03*.

MEDIONI, G., LEE, M. S., AND TANG, C. K. 2000. *A Computational Framework for Segmentation and Grouping*. Elsevier.

MORSE, B. S., YOO, T. S., CHEN, D. T., RHEINGANS, P., AND SUBRAMANIAN, K. R. 2001. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. *IEEE Computer Society Press*, 89–98. in SMI '01:Proceedings of the International Conference on Shape Modeling & Applications.

MUMFORD, D. AND SHAH, J. 1989. Optimal approximation by piecewise smooth functions and associated variational problems. *Commun. Pure Appl. Math 42*, 577–685.

OHTAKE, Y., BELYAEV, A. G., AND SEIDEL, H. P. 2003. A multi-scale approach to 3d scattered data interpolation with compactly supported basis function. *Shape Modeling International 292*, 153–164.

OSHER, S., BURGER, M., GOLDFARB, D., XU, J., AND YIN, W. 2005. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling and Simulation 4*, 460–489.

PAULY, M., GROSS, M., AND KOBBELT, L. P. 2002. Efficient simplification of point-sampled surfaces.

PAULY, M., MITRA, N. J., AND GUIBAS, L. 2004. Uncertainty and variability in point cloud surface data. In *Symposium on Point-Based Graphics*.

PENG, D., MERRIMAN, B., OSHER, S., AND ZHAO, H. 1999. A pde-based fast local level set method. *Journal of Comp. Physics 155*, 410–438.

PIEGL, L. AND TILLER, W. 1996. *The NURBS book*. Springer-Verlag, Berlin, Germany.

ROGERS, D. F. 2001. *An introduction to NURBS: with historical perspective*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

ROGERS, D. F. 2003. *An Introduction to NURBS*. Morgan Kaufmann.

RUDIN, L. I., OSHER, S., AND FATEMI, E. 1992. Nonlinear total variation based noise removal algorithms. *Physica D 60*, 259–268.

SETHIAN, J. A. 1996a. A fast marching level set method for monotonically advancing fronts. *Proc. Nat. Acad. Sci. 93,* 4, 1591–1595.

SETHIAN, J. A. 1996b. *Level Set Methods: Evolving interfaces in geometry, fluid mechanics, computer vision, and materials science*. Cambridge University Press.

TSAI, Y. R. 2002. Rapid and accurate computation of the distance function using grids. *J.Comp.Phys. 178,* 1, 175–195.

WAN, M., WANG, Y., BAE, E., TAI, X. C., AND WANG, D. 2010. Reconstructing open surfaces via graph-cuts. *UCLA CAM Report 10,* 29.

WENDLAND, H. 2002. Fast evaluation of radial basis functions: Methods based on partition of unity. In *Approximation Theory X: Wavelets, Splines, and Applications*. Vanderbilt University Press, 473–483.

YE, J., BRESSON, X., GOLDSTEIN, T., AND OSHER, S. 2010. A fast variational method for surface reconstruction from sets of scattered points. *UCLA CAM Report 01,* 10.

ZHAO, H. K. 2005. A fast sweeping method for eikonal equations. *Mathematics of Computation 74*, 603–627.

ZHAO, H. K., OSHER, S., AND FEDKIW, R. 2001. Fast surface reconstruction using the level set method. *Proceedings. IEEE workshop*, 194–201.

ZHAO, H. K., OSHER, S., MERRIMAN, B., AND KANG, M. 1998. Implicit, nonparametric shape reconstruction from unorganized points using a variationallevel set method. *Computer Vision and Image Understanding 80*, 295–319.

ZHU, M. AND CHAN, T. F. 2008. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *UCLA CAM Report 08,* 34.