Noname manuscript No.
(will be inserted by the editor)

Reconstructing Open Surfaces via Graph-Cuts

Min Wan, Yu Wang, Egil Bae, Xue-Cheng Tai, Desheng Wang

Received: date / Accepted: date

Abstract A novel graph-cuts-based method is proposed for reconstructing open surfaces from unordered point sets. Through a boolean operation on the crust around the data set, the open surface problem is translated to a watertight surface problem within a restricted region. Integrating the variational model, Delaunay-based tetrahedral mesh framework and multi-phase technique, the proposed method can reconstruct open surfaces robustly and effectively. Furthermore, a surface reconstruction method with domain decomposition is presented, which is based on the new open surface reconstruction method. This method can handle more general surfaces, such as non-orientable surfaces. The algorithm is designed in a parallel-friendly way and necessary measures are taken to eliminate cracks at the interface between the subdomains. Numerical examples are included to demonstrate the robustness and effectiveness of the proposed method on watertight, open orientable, open non-orientable surfaces and combinations of such.

Keywords Graph-cuts · Open Surface · Domain Decomposition · Delaunay triangulation

M. Wan, Y. Wang, X.-C. Tai, D. Wang
Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371
E-mail: {wanm0003;wang0312}@e.ntu.edu.sg, {xctai;desheng}@ntu.edu.sg

E. Bae, X.-C. Tai
Department of Mathematics, University of Bergen, Johannes Brunsgate 12, 5007 Bergen, Norway
E-mail: {egil.bae;tai}@math.uib.no

1 Introduction

Reconstructing a surface from an unordered point data set has been a significant yet challenging problem in computer graphics for the last decade. As a critical step of creating computer graphics, surface reconstruction fills the gap between machine perception and machine understanding, i.e. the process from discrete scanned data to a continuous model. Due to the development of three dimensional scanners and the increasing demand of computer graphics, extensive research has been conducted in the surface reconstruction field, much of which was dedicated to the watertight surface reconstruction for its topological simplicity and desirable properties. Open surface reconstruction problems, however, occur often in real applications, such as incomplete scanned data. As a topic which has been overlooked, the open surface reconstruction problem, to some extent, has more significance than the watertight surface problem for its topological generality. The definitions of watertight and open surface are as follows.

A surface is defined as 2-manifold embedded in R^3. In our study, we restrict a surface to be a compact 2-manifold, which we are referring to by saying a watertight surface. A surface with boundary is a 2-manifold with boundary embedded in R^3, which is the definition of an open surface. (Dey, 2007)

Most surface reconstruction methods can be categorized into two groups, explicit methods and implicit methods. Explicit methods are mainly local geometric approaches based on Delaunay triangulation and dual Voronoi diagram such as Alpha shape and CRUST algorithm (Adamy et al, 2000; Amenta et al, 1998, 2000; Boissonnat and Cazals, 2000; Dey and Goswami, 2003; Edelsbrunner and Mucke, 1992). One advantage of these methods is their theoretical guarantee that there exists

a sub-complex of Delaunay triangulation of the data set, which is homeomorphic to the ground truth surface given a sufficient sampling. Since these methods are local approaches, the global topological characteristics such as watertight or open, will not affect their performances. Their target is the potential homeomorphic sub-complex embedded in the Delaunay triangulation. The topology of the sub-complex surface does not make any difference. Hence, the explicit method can handle quite a number of open surface cases.

However, the explicit methods are subject to many reconstruction difficulties such as non-uniformity, undersampling and noises (Amenta et al, 1998; Franchini et al, 2010a; Zhao, 2000). Readers can compare the result from explicit methods in Fig. 1 to our experiment on the same data set in Section 5. Hence, during the last decade, variational models were brought into the reconstruction field. The reconstruction problem is formulated as a minimization problem of an energy functional defined over surfaces. To minimize an energy functional with respect to the surface, a consistent parametrization of the surface is not always available during the optimization procedure. As a result, researchers turned to the implicit methods (Alexa et al, 2001; Curless and Levoy, 1996; Franchini et al, 2010a,b; Hoppe et al, 1992; Ohtake et al, 2005; Solem and Heyden, 2004; Solem and Kahl, 2004, 2005; Solem and Overgaard, 2005; Ye et al, 2010; Zhao et al, 2001), such as the level set method, to gain flexibility of representation and mathematical facilities. One important such level set approach based on solving the underlying partial differential equations was proposed by Zhao in (Zhao et al, 2001; Zhao, 2000). As an alternative, graph cuts can also minimize the energy functionals over implicitly defined surfaces, and has been successfully applied to the surface reconstruction problem in (Hornung and Kobbelt, 2006b; Lempitsky and Boykov, 2007; Paris et al, 2006). The main advantages of graph cuts are the efficiency and ability to find global minima. However, the competence of both the level set method and graph cuts is lost on more general topologies (Osher and Fedkiw, 2002). Some reconstruction methods could also handle open surfaces (Hornung and Kobbelt, 2006b; Kuo and Yau, 2005; Yu, 1999). The success of (Kuo and Yau, 2005; Yu, 1999) relies on the uniformity of input data set. To determine the connectivity among data points depends on the statistical property of data closeness. Besides, they are derived from explicit methods and also subject to the noises and outliers. (Hornung and Kobbelt, 2006b) proposed a hierarchical approach which could reconstruct some open surfaces, in which openness can be treated as holes. In their work, the coarse grids automatically fill the holes and create a watertight environment in the first several hierarchical stages. However, this success relies on the size and shape of the surface boundary. When surface boundary does not resemble a hole at all, the hierarchical method would fail. Another recent work (Solem and Heyden, 2006) to reconstruct open surfaces is to use multiple level set functions based on curve motion (Bertalmío et al, 1999; Burchard et al, 2001; Cheng et al, 2002; Faugeras and Gomes, 2000; Smereka, 2000). Instead of the gradient descent in (Zhao et al, 2001), they propagate two implicit surfaces in a same vector field. The portion of the main surface enclosed in the auxiliary surface is the open surface. However, the efficiency of surface propagation is still a problem in that study. To sum up, all these methods for open surfaces have some disadvantages and " *it is not clear how to devise methods for curves and surfaces that have ends or edges (respectively) within the computational domain*" (Osher and Fedkiw, 2002).

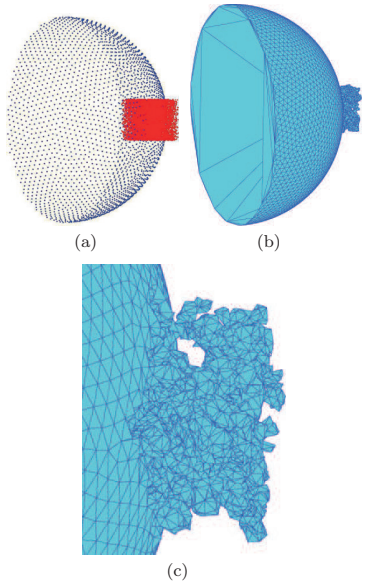


Fig. 1 Failure of CRUST algorithm on a noisy open case. (a) shows the input data points including blue ground truth data and red outliers. (b) shows the reconstructed result with unremoved outliers. (c) is zoomed view.

In this article, a novel variational reconstruction method for open surfaces is proposed. Unlike previous methods, our method separates the two types of ill-posedness in the open surface problem and handles them sequentially in different ways. The explicit methods are adapted to handle the uncertainty of the surface boundary. The medial axis frequently used in the

explicit methods are applied in a different manner. And the implicit method is used to tackle the uncertainty of the data connectivity issue. This new methodology not only grants the merits from both explicit and implicit methods, but also provides approaches to more general cases such as the combination of open and watertight surfaces. Following is a description of our algorithm.

In the proposed method, the data set points as well as the properly generated background points are inserted to an unstructured tetrahedral mesh framework in a Delaunay way. Due to its nearest connection property, the Delaunay triangulation combining a sufficient sampling density provides a theoretic guarantee that there exists a sub-complex of the Delaunay triangulation such that it is homeomorphic to the ground truth surface. In the tetrahedral mesh, a crust is established around the data set. The crust is the vicinity domain of the input data set. A more precise definition is given in Section 2. In (Wan et al, to appear), a graph dual to the whole mesh is built according to the energy functional and the minimization is achieved by applying max-flow/min-cut algorithms. Since these algorithms find a global minimum, it is essential to specify boundary conditions of the crust. This can only be accomplished under the assumption that the domain can be separated into two or more subdomains by the watertight crust, which does not hold any more for an open surface problem. Without specifying boundary conditions, the dual graph does not have valid n-links to both source and sink, resulting in trivial min-cuts.

To tackle this issue, a boolean operation is proposed to restrict the region of interest within a narrow band, which can be separated into two or more subdomains. In the proposed method, two crusts with different thickness are built around the data set. The medial axis of the thick crust is to be obtained. One more crust is then built around the boundary of the medial axis. Subsequently, the two crusts around the data set are trimmed by the crust around the boundary. The trimmed thick crust can be separated by the trimmed thin crust. Hence in the restricted region, i.e. the trimmed thick crust, the trimmed thin crust is watertight such that region growing algorithms and graph cut techniques can be applied. More details and illustrations of this series of operations are provided in Section 3. The method subsequently constructs a graph dual to the restricted mesh, applies max-flow/min-cut algorithms and extracts the surface from the tetrahedral mesh according to the obtained minimal cut. A flow chart of the whole algorithm is shown in Fig. 2.

Furthermore, a surface reconstruction method based on domain decomposition is presented. The domain decomposition idea has been applied to computer vision

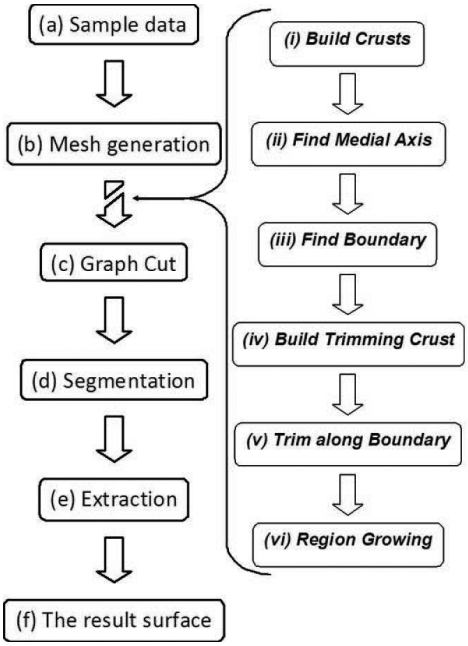


Fig. 2 The flowchart of open surface reconstruction

(Kohlberger et al, 2003, 2004, 2005). Recently it is found also useful as a robust alternating minimization scheme between overlapped subspaces (Tai and Duan, 2009; Tai and Xu, 2002). In recent study, the dual graph could be subdivided into subgraphs as well to gain extra efficiency (Strandmark and Kahl, 2010). In the decomposition method, the whole domain is decomposed into several subdomains. In each subdomain, a surface reconstruction problem, input of which is a subset of the whole data points, is solved. Merging all the surface patches from different subdomains is the critical task. Such a fix-the-boundary measure is taken before the graph technique is applied that potential conflicts and cracks can be eliminated effectively. The parallel efficiency may be undermined due to the interaction between subdomains. However, it can be compensated largely by a proper decomposition scheme. The method proposed in this article can handle not only open surfaces but also more general surfaces such as combinations of open and watertight surfaces.

The remainder of this paper is organized as follows. In Section 2, a brief review of watertight surface reconstruction based on Delaunay triangulation and graph-cuts will be given. Section 3 deals with the open surface problem. The new method to tackle this problem is proposed and the algorithm is given in details. Section 4

gives an important application of the open surface reconstruction method, the surface reconstruction based on domain decomposition, which can handle more general surfaces. To the best of our knowledge, this is the first attempt to approach the nonorientable surface reconstruction problem via graph-cuts. In Section 5 various numerical examples are presented to demonstrate effectiveness and robustness of the proposed method on all kinds of surfaces. Finally, Section 6 concludes the article.

2 Graph-cuts Reconstruction of watertight surface

In the previous work (Wan et al, to appear) a variational reconstruction method was proposed for watertight surfaces based on graph-cuts. The cost energy functional is a generalization from the weighted minimal surface model (Zhao et al, 2001), which is also related to the minimal surface (Caselles et al, 1997b) or geodesic active contours (Caselles et al, 1997a) approaches. This functional is minimized on an unstructured tetrahedral mesh framework, which provides more flexibility and effectiveness than structured grids used in other graph-based methods (Hornung and Kobbelt, 2006a,b; Paris et al, 2006). As a matter of fact, the Delaunay-based mesh guarantees the existence of a subcomplex homeomorphic to the ground truth surface given a sufficient sampling. The method can handle various reconstruction difficulties such as noise, undersampling and non-uniformity. By adopting the idea presented in (Bae and Tai, 2009), the method is able to address two phase and multi-phase problems in a unified approach. In addition, an automatic phase detecting method based on region growing algorithms is developed to minimizes user intervention. A brief review of the ideas and techniques on watertight surfaces will be given in this section.

2.1 Two phase surface reconstruction via graph-cuts

For convenience, this subsection only discusses two phase problems, in which the ground truth surface S simply separates the embedding domain $X \subset R^3$ into two connected regions, inside and outside. Let P be a point data set sampled from S in the domain X. Define the distance function as $d(x) = d(x, P) = \inf_{y \in P} d(x, y)$, where $d(x, y)$ is the Euclidean distance between points x and y in R^3. As in (Wan et al, to appear; Zhao et al, 2001), the following cost energy is proposed for surface reconstruction,

$$E(\Gamma) = \int_X |\phi_\Gamma(x) - I(x)|\, \beta(x) dx + \int_\Gamma d(x) ds + \alpha \int_\Gamma ds\,, \tag{1}$$

where Γ is an arbitrary surface and ds is the surface area.

The above $\phi_\Gamma(x)$ is the piecewise constant level set function same as (Lie et al, 2006) corresponding to the surface Γ

$$\phi_\Gamma(x) = \begin{cases} c_1 \text{ if } x \text{ inside } \Gamma \\ c_2 \text{ if } x \text{ outside } \Gamma \end{cases}. \tag{2}$$

c_1 and c_2 serve as the constant level set value and could be any distinct constants. As a consequence, the surface Γ is implicitly represented as the discontinuities of $\phi_\Gamma(x)$.

The crust around P is defined as $C_d^P = \{x \in X : d(x, P) \leq d\}$. Given a watertight surface and a reasonably dense sampling, we assume the crust around the sampling data set is able to partition the whole domain into two connected regions, i.e. interior and exterior. $I(x)$ is an indicator function which labels these two subdomains as well as the crust region. Compared with $\phi_\Gamma(x)$ which labels the final partitioning , this indicator function serves as an initial labelling.

$$I(x) = \begin{cases} 0 \text{ if } x \text{ in } C_d^P \\ c_1 \text{ if } x \text{ in the interior part of } X \backslash C_d^P \\ c_2 \text{ if } x \text{ in the exterior part of } X \backslash C_d^P \end{cases}. \tag{3}$$

In (1), $\beta(x)$ is a confidence function suggesting the extent to which the indicator function, the estimate for the level set function, is faithful. The reconstructed surface is rather unlikely to fall outside the crust region given a low noise level, which results in the following specification of $\beta(x)$.

$$\beta(x) = \begin{cases} 0 \text{ if } x \text{ in } C_d^P \\ \sigma \text{ others} \end{cases}, \tag{4}$$

where σ is a relatively large positive value.

The first term in (1), can be viewed as specifying boundary conditions on ϕ at the boundary of the crust. By specifying proper $I(x)$ and $\beta(x)$, the first term would constrain the resulting surface within a restricted region, i.e. C_d^P. Otherwise, if there are any disagreements between $\phi_\Gamma(x)$ and $I(x)$ out of the crust region where $\beta(x) = \sigma$, the energy would not be minimized. This term is important; without it the global minimum of (1) would be the trivial null surface, where ϕ_Γ is just a constant everywhere.

The second term is the essential part in the weighted minimal surface model (Zhao et al, 2001) and the third

term is the regularization term concerning the surface area. By tuning the regularization coefficient α, a compromise between faithfulness and smoothness can be achieved.

In this method, (1) is discretized on an unstructured tetrahedral mesh $\mathcal{T}_h$ instead of structured grids used in other graph-based methods. And a mesh and triangulation are referring to the same thing (George and Borouchaki, 1998). Generally, a mesh can be defined by a pair (V, C). V is the set of all vertices and C is a complex consisting of four types of simplexes, i.e. vertices, edges, triangles, and tetrahedra. For vertices u, v, w, z, we define $\{v, u\}$ as the edge between v and u, $\{v, u, w\}$ as the triangle with vertices v, u, w, and $\{v, u, w, z\}$ as the tetrahedron with vertices v, u, w, z. $\{K_i\}_{i=1}^N$ are used to denote all N tetrahedra in $\mathcal{T}_h$. In our case V is the set of mesh points including data points and background points $P \cup Q$.

In a mesh $\mathcal{T}_h$, we can define 1-ring neighborhood of a vertex v as $N_v^1 = \{u | \{v, u\} \in C\}$ and M-ring neighborhood in a recursive way $N_v^M = \{u | \exists w \in N_v^{M-1}, \{w, u\} \in C\}$. Based on this neighborhood system, the crust around the data set P can be defined as $K_M^P = \{K_i | \exists v \in K_i, v \in N_u^M, u \in P\}$.

Given P, the sizing function $h(v)$ for each vertex $v \in P$ can be defined as the $d(v, P \backslash \{v\})$: the closeness measure to the other vertices. Under uniformity assumption, the average sizing function $\bar{h} = \sum_{v \in P} v(h)$ could well approximate that of each individual vertex. The background points aim to construct the mesh of reasonable size and good quality. Either regular grid or Body-centered cubic (BCC) lattice is a good choice.

$$BCC(h) = h \cdot \left(\mathbb{Z}^3 \cup \left(\mathbb{Z}^3 + (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}) \right) \right) \tag{5}$$

where $\mathbb{Z}^3$ are points with integer coordinates and h is the size of BCC lattice (Gray and Neuhoff, 2002). The background point set is defined as $Q = \{v | v \in BCC(\bar{h}) \cap X, d(v, P) > \bar{h}\}$. The restricting inequity $d(v, P) > \bar{h}$ is necessary since too close background points would destroy the ground truth embedded in the mesh.

When the data sets are non-uniform, i.e. with widely varying sizing function values, the uniform background lattice points are not suitable any more. A well graded sizing mesh is required instead. The mesh element sizes shall conform with the local sizing function. The parameter h of $BCC(h)$ shall vary correspondingly. A good meshing technique is (Labelle and Shewchuk, 2007), which utilized octree to construct graded BCC meshes. More advanced non-uniform mesh generation technique is also available, see (Du and Wang, 2002). The benefit of such a reasonable sized mesh will be seen later.

In this mesh framework, the surface Γ can be approximated by Γ_h, a sub-complex of $\mathcal{T}_h$. (Amenta et al, 1998) shows that there exists a sub-complex of the Delaunay triangulation of P, which is homeomorphic to the ground truth surface S. As a consequence of this fact and the local property of Delaunay triangulations, there also exists a homeomorphic-to-S sub-complex of the Delaunay triangulation of $Q \cup P$ given a reasonable distribution of background points Q.

The first term in (1) , the integral over the whole domain X can be simply discretized as

$$\begin{aligned} &\int_X |\phi_\Gamma(x) - I(x)| \, \beta(x) dx \\ &= \sum_{i=1}^N \int_{K_i} |\phi_\Gamma(x) - I(x)| \, \beta(x) dx \\ &\approx \sum_{i=1}^N |\phi_{\Gamma_h}(K_i) - I(K_i)| \, \beta(K_i) \, . \end{aligned} \tag{6}$$

$$\phi_{\Gamma_h}(K_i) = \begin{cases} c_1 \text{ if } K_i \text{ inside } \Gamma_h \\ c_2 \text{ if } K_i \text{ outside } \Gamma_h \end{cases} , \tag{7}$$

$$I(K_i) = \begin{cases} 0 \text{ if } K_i \in K_M^P \\ c_1 \text{ if } K_i \text{ in the interior} \\ c_2 \text{ if } K_i \text{ in the exterior} \end{cases} , \tag{8}$$

$$\beta(K_i) = \begin{cases} 0 \text{ if } K_i \in K_M^P \\ \sigma \text{ others} \end{cases} , \tag{9}$$

where σ is a relatively large positive value.

The second and third terms in (1) are integrals over the surface area. The surface triangulation Γ_h can be thought of as the union of the triangular faces shared by tetrahedra with different level set values.

$$\Gamma_h = \bigcup_{\phi_\Gamma(K_i) \neq \phi_\Gamma(K_j)} \Gamma_{ij} \, ,$$

where $\Gamma_{ij} = K_i \cap K_j$. Hence combining (6), (1) can be discretized as follows

$$\begin{aligned} E(\Gamma) &\approx \sum_{i=1}^N |\phi_{\Gamma_h}(K_i) - I(K_i)| \, \beta(K_i) \\ &+ \sum_{i,j} (d_{ij} + \alpha) S_{ij} \mathbf{1}_{\{\phi_{\Gamma_h}(K_i) \neq \phi_{\Gamma_h}(K_j)\}} \, , \end{aligned} \tag{10}$$

where

$$d_{ij} = \frac{\int_{\Gamma_{ij}} d(x) ds}{\int_{\Gamma_{ij}} ds} \quad , \quad S_{ij} = \int_{\Gamma_{ij}} ds \, . \tag{11}$$

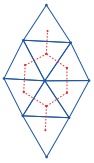


Fig. 3 The primal-dual relationship of triangular mesh (Wan et al, to appear)

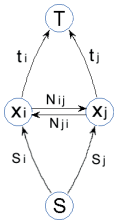


Fig. 4 Graph edge weight assignment (Wan et al, to appear)

Table 1 Relationship between cut and surface

Cut in dual graph	Surface in primal mesh
$C = \bigcup_{x_i,x_j \in V,\ \phi_i \neq \phi_j} (x_i, x_j)$	$\Gamma = \bigcup_{K_i,K_j \in \mathcal{T}_h,\ \phi_i \neq \phi_j} (K_i \cap K_j)$

The energy of $E(\Gamma)$ can be minimized very efficiently by graph-cuts, since this energy functional is graph representable, which can be verified by the conclusion of (Kolmogorov and Zabin, 2004). First, a graph dual to the primal tetrahedral mesh is constructed, in which each node corresponds to a tetrahedron in the mesh and each edge corresponds to a triangular face in the mesh. This primal-dual relationship is illustrated for two dimensions in Fig 3.

The edge weights are determined by different terms in $E(\Gamma)$ as shown in Fig. 4 and below

$$\begin{aligned} s_i &= |I(K_i) - c_2|\,\beta(K_i)\,, \quad t_i = |I(K_i) - c_1|\,\beta(K_i)\,, \\ N_{ij} &= (d_{ij} + \alpha) S_{ij}\,, \qquad N_{ji} = (d_{ij} + \alpha) S_{ij}\,, \end{aligned} \tag{12}$$

where c_1 and c_2 are the piecewise constant level set function values, standing for the regions inside and outside the surface.

After graph construction, max-flow/min-cut algorithms can be applied on the obtained graph. The algorithm in (Boykov and Kolmogorov, 2004) is a good choice for its empirically good performance. Due to the primal-dual relationship in Table 1, the reconstructed surface can be directly extracted from the background

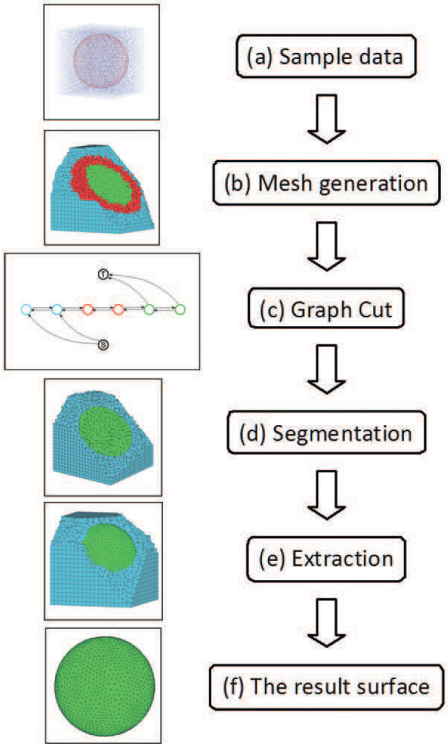


Fig. 5 Watertight surface reconstruction. Given a data set (a) sampled from an object surface, proper background points such as grid points are generated according to the data points distribution. An unstructured tetrahedral mesh (b) is generated in a Delaunay way and the crust around the data set is established. A graph dual to the mesh is constructed (c). Graph-cuts are applied and segmentation on the primal mesh is obtained (d). Extract the surface from tetrahedral mesh (e), the reconstructed surface is obtained (f)

Table 2 Watertight surface reconstruction method

Inputs	A data point set P
Algorithm	
1.	Generate background points Q according to the density of P
2.	Insert P and Q to a tetrahedral mesh $\mathcal{T}_h$ in a Delaunay way
3.	Establish the crust, K_P^M
4.	Region growing on the regions outside K_P^M
5.	Specify the Indicator function according to (3)
6.	Construct a graph dual to the mesh
7.	Assign edge weights according to (12)
8.	Apply the graph-cuts
9.	Extract the surface according to the minimal cut
Outputs	The surface triangulation S

mesh according to the minimal cut. The whole algorithm is shown in Table 2 and the flow chart is shown in Fig 5.

2.2 Multi-phase surface reconstruction

We assume now that the interior and exterior of the surface are not connected sets. Such cases can be handled by introducing more labels. We assume the surface separates X into M connected regions $\{X_i\}_{i=1}^{M}$. Surfaces of this kind can be represented in the level set framework of (Lie et al, 2006) by defining ϕ_Γ as $\phi_\Gamma(x) = c_i$ for $x \in X_i$, $i = 1, ..., M$. As before, Γ is represented as the discontinuities of ϕ_Γ. The complete energy functional (1) is therefore given in the discrete setting as

$$E(\Gamma) \approx \sum_{i=1}^{N} |\phi_{\Gamma_h}(K_i) - I(K_i)|\, \beta(K_i) + \sum_{i,j} (d_{ij} + \alpha) S_{ij} \mathbf{1}_{\{\phi_{\Gamma_h}(K_i) \neq \phi_{\Gamma_h}(K_j)\}} \tag{13}$$

where

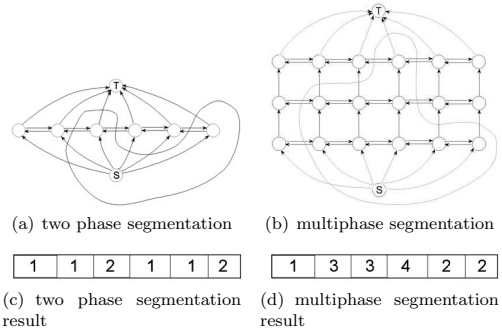
$$d_{ij} = \frac{\int_{\Gamma_{ij}} d(x) ds}{\int_{\Gamma_{ij}} ds} \quad , \quad S_{ij} = \int_{\Gamma_{ij}} ds \,. \tag{14}$$

Minimization problems with multiple phases, or labels, have been studied previously in image processing. The work of (Ishikawa, 2003) and a later modification (Bae and Tai, 2009) presented techniques to efficiently minimize certain such multilabel problems by graph cuts. By making a simplification of the length term in (13), we can convert the problem (13) to such graph representable form. It was observed that several surfaces could be represented by a hyper-surface in a higher dimensional domain. Hence the multi-way cut problem is equivalent to a binary cut problem in a multi-layer graph. Therefore, an extra dimension is introduced to the original graph dual to the primal mesh. This multilayer graph idea is illustrated in Fig. 6.

The multi-layer idea does not change much from image processing to surface reconstruction. As earlier, we let C_d^P denote the crust around the data points P. The domain $X \backslash C_d^P$ now contains several disconnected subdomains (instead of just two as in the last subsection). The indicator function I should be specified such that it takes different values in different subdomains

$$I(x) = \begin{cases} 0 \text{ if } x \text{ in } C_d^P \\ c_i \text{ if } x \text{ inside the } i\text{th subdomain.} \end{cases} . \tag{15}$$

Once the original graph dual to the primal mesh is constructed, it is duplicated $M-1$ times if the number of subdomains is M. More specifically, a graph is created



(a) two phase segmentation (b) multiphase segmentation

(c) two phase segmentation result (d) multiphase segmentation result

Fig. 6 One dimensional example to illustrate multilayer graph (Bae and Tai, 2009)

such that $M-1$ vertices in the vertex set are associated to each tetrahedra K_i. The notation v_i^k is used for the vertex corresponding to K_i at level $k \in \{1, ..., M-1\}$. We let $c(a, b)$ denote the cost on the edge between vertex a and b. The edges connecting vertices in the same level are called horizontal edges, while the others are called vertical edges. The weights for the vertical edges represent the data term, and are defined by

$$\begin{aligned} c(s, v_i^1) &= |c_1 - I(K_i)|\, \beta(K_i) \text{ for } i = 1, ..., N, \\ c\left(v_i^k, v_i^{k+1}\right) &= |c_{k+1} - I(K_i)|\, \beta(K_i) \text{ for } i = 1, ..., N, \\ &\quad \forall k \in \{1, ..., M-2\}, \\ c\left(v_i^{M-1}, t\right) &= |c_M - I(K_i)|\, \beta(K_i) \text{ for } i = 1, ..., N. \end{aligned} \tag{16}$$

The weights for the horizontal edges represent the regularization term in functional (1), and are defined as follows.

$$c(v_i^k, v_j^k) = (d_{ij} + \alpha) S_{ij} \,, c(v_j^k, v_i^k) = (d_{ij} + \alpha) S_{ij}, \tag{17}$$

$$\forall i, j \in \{1, ..., N\}, \ \forall k \in \{1, ..., M-1\}.$$

After finding the minimum cut $\mathcal{C}$ on this graph, the labeling function can be recovered by

$$\phi_i = \begin{cases} c_1 & \text{if } (s, v_i^1) \in \mathcal{C} \\ c_{k+1} & \text{if } (v_i^k, v_i^{k+1}) \in \mathcal{C}, \ k = 1, ..., M-2 \\ c_M & \text{if } (v_i^{M-1}, t) \in \mathcal{C} \end{cases} . \tag{18}$$

As shown in Fig. 7, the multi-layer graph idea is illustrated by two intersecting spheres. Fig. 7(a) presents the cut view of the mesh, where red crust separates the domain into four regions marked with different colors. The corresponding three layer graph is shown in Fig. 7(b), in which $I(K_i) = 1, 2, 3, 4$ when K_i is blue, green, purple, or brown. The nodes in the graph correspond

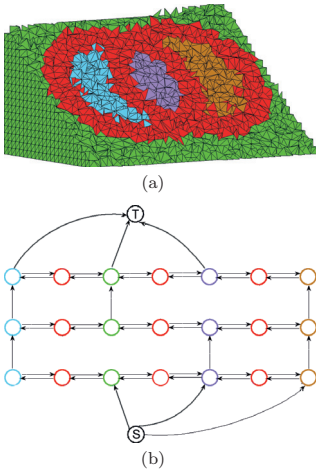


Fig. 7 A multi-phase surface problem and the corresponding multi-layer graph (Wan et al, to appear)

to the tetrahedra with the same color. The weights distribution among vertical edges depend on $I(K_i)$. It is worth noticing that some vertical edges vanish as shown in (b) and the red nodes do not have vertical edges at all.

In order to determine the number of subdomains M, an intelligent method for detecting the number of subdomains based on region growing algorithms is applied after the mesh generation and crust establishment. In this procedure, the indicator function $I(K_i)$ is specified automatically. User intervention is optional, but in most cases unnecessary. As first developed in the image segmentation field, the region growing algorithm (Adams and Bischof, 1994) mainly consists of the following steps. Firstly, several initial seeds are selected. Secondly, for each seed, its neighborhood is examined to decide whether that belongs to the same partition or not. Based on this idea, a phase detection method is developed on the tetrahedral mesh, in which the neighborhood of a tetrahedron K_i are four tetrahedra sharing one face with K_i respectively. In this method, the seeds are not required to be appointed. Instead they are picked automatically during the algorithm, which is presented in Table 3 and Table 4.

Table 3 Phase detecting method based on region growing

Inputs	
1.	A mesh $\mathcal{T}_h = (P \cup Q, C), \{K_i\}_{i=1}^N \in C$
2.	Labelling values $c_i, i = 1, \ldots, M$.
Algorithm	
1.	Construct the crust K_P^M
	%% Initiate all tetrahedra
2.	For $i = 1 : N$
3.	If $K_i \in K_P^M$
4.	$I(K_i) = 0$
5.	Else
6.	$I(K_i) = -1$
7.	End If
8.	End For
	%% Region growing all tetrahedra out of crust
9.	$L = 1$
10.	For $i = 1 : N$
11.	If $I(K_i) == -1$
12.	$region\_growing(K_i, c_L)$
13.	$L = L + 1$
14.	End If
15.	End For
	%% At the end, all tetrahedra in the same partition are labeled the same value.

Table 4 Region growing function

function	$region\_growing(K, l)$
Function	
	%% $\{N_i\}_{i=1}^4$ are four neighbors of K, and l is the label value
1.	$I(K) = l$
2.	For i=1:4
3.	If $I(N_i) == -1$
4.	$region\_growing(N_i, l)$
5.	End If
6.	End For

3 Open Surface Reconstruction via Graph-cuts

The method discussed in Section 2 can reconstruct watertight surfaces, which has an interior and exterior region in R^3. In this section we discuss open surfaces, which obviously does not have a clear interior and exterior. One critical step of the previous method was the specification of the indicator function $I(x)$ as the establishment of the boundary conditions, which was completed by the phase detection method based on region growing algorithms described in Section 2. If the crust around the data set fails to separate the domain into two or more partitions as in Fig. 5(b), the phase detector would label all regions out of the crust with the same indicator value. A solid and reasonable boundary condition is not available and hence the global minimum would be the trivial null surface. Therefore graph-cuts can not be conducted properly. Fig 8 illustrates this situation and the failure of our previous method by an example in two dimensions.

Certain interactive specification can be used in this situation as in Fig. 9(a). One spot (small region) on each side of the potential surface has been assigned with different indicator values as two 'seeds'. The graph cut result is shown in Fig. 9(b). It can be noticed that the result has been artificially extended from two ends

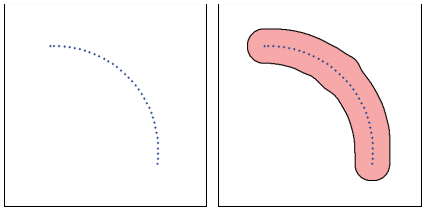


Fig. 8 The failure of previous graph based methods

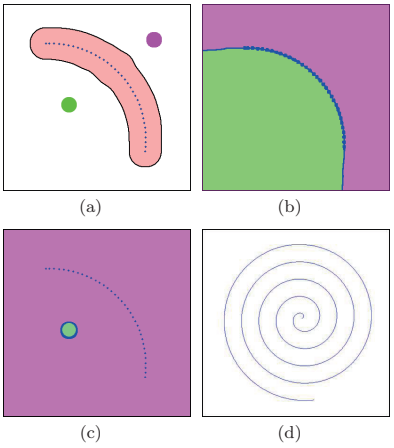


Fig. 9 An interactive method to handle open surface and its limitation

of the ground truth curve. This is inevitable since the minimal cut is required to separate the whole graph, which corresponds to the whole domain. In addition, the selection of the "seeds" spot should be rather cautious. Otherwise, improperly small "seeds" as well as a great regularization coefficient is likely to lead to a trivial result as shown in Fig. 9(c), in which the cut and the corresponding surface shrink to the boundary of a "seed" spot. All these disadvantages aside, this interactive method apparently lacks generality to be applied on more complicated cases such as Fig. 9(d). It is troublesome for users to select two 'seeds' in the complex spiral curve, not to mention that the graph-cuts result would be ruined by the artificially extended surface. All above considered, in this article, a more intelligent and robust reconstruction method for general surfaces, including open surfaces, watertight surfaces, and combinations of such is proposed.

As presented and illustrated above, the gap between our previous method and the new problem of open surfaces is a reasonable partitioning of the region out of the crust. The proposed method consists of an automatic partitioning procedure followed by all steps contained in Section 2. By defining a boolean operation on the vicinity of the data set, the region of interest has been trimmed in such a way that it can be separated into two or more partitions by a watertight crust. Subsequent phase detection and graph techniques can be applied on the trimmed region. Detailed description is as follows.

3.1 A description of the method

Given a point set P in the domain $X \subset R^3$, which is sampled from the surface S. The distance $d(x, P)$ and the crust C_d^P is defined in the same way as in Section 2. Firstly, two crusts with different thickness parameters $d_1 < d_2$ are constructed around P: $C_{d_1}^P$ and $C_{d_2}^P$. $C_{d_2}^P$ rather than the whole domain X is the region of interest. The resulting surface is supposed to lay in $C_{d_1}^P$. These two crusts are illustrated in Fig. 10(a), in which the inner crust $C_{d_1}^P$ fails to separate the region $C_{d_2}^P$ and to create a watertight environment.

Secondly the medial axis M_d of the boundary of $C_{d_2}^P$ is to be found. As defined in (Amenta et al, 1998), the medial axis of a manifold $\Sigma \subset R^k$ is the closure of the set of points in R^k that have at least two closest points in Σ. Under a noise-free assumption, this medial axis M_d itself is a good approximation to the ground truth surface S. Well approximating as it is, the medial axis is only an intermediate product of the algorithm. More steps are required to handle difficulties such as noises and non-uniformity.

Thirdly Bd, the boundary of M_d is found, which well approximates the boundary of the ground truth surface S. A crust around Bd is constructed: $C_{d_3}^{Bd} = \{x \in X, : \ d(x, Bd) \leq d_3\}$, $d_3 \geq d_2$. Subsequently those two crusts around P, i.e. $C_{d_1}^P$ and $C_{d_2}^P$, are trimmed by the crust around Bd, i.e. $C_{d_3}^{Bd}$, which can be expressed as the boolean operation: $\widetilde{C}_{d_1}^P = C_{d_1}^P - (C_{d_3}^{Bd} \cap C_{d_1}^P)$, $\widetilde{C}_{d_2}^P = C_{d_2}^P - (C_{d_3}^{Bd} \cap C_{d_2}^P)$. We can safely assert that $\widetilde{C}_{d_2}^P$ can be separated into two or more partitions by $\widetilde{C}_{d_1}^P$ given sufficient sampling and proper d_1, d_2 and d_3. This procedure is illustrated in Fig 10(b), in which Bd in two dimensions is the two ends of the curve. Two red crusts, $C_{d_1}^P$ and $C_{d_2}^P$, have been trimmed by the gray circles, i.e. $C_{d_3}^{Bd}$, and the remaining light red crust $\widetilde{C}_{d_2}^P$ is separated by the remaining dark red one $\widetilde{C}_{d_1}^P$. Hence the phase detector can label these disconnected subdomains with different indicator values and graph-cuts can be applied to the $\widetilde{C}_{d_2}^P$ as shown in Fig. 10(c),(d). These two steps are same to those described in Section 2 except that

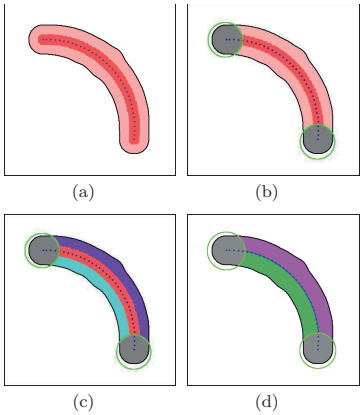


Fig. 10 Crust establishments and boolean operation

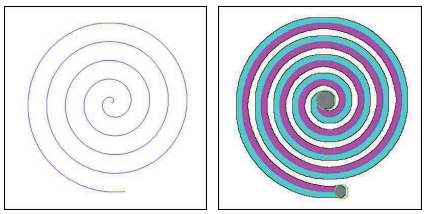


Fig. 11 Approaching the case in Fig. 9(d)

the region of interest is no longer the whole domain X. As a proof of the effectiveness of the proposed method, the case in Fig 9(d) can be approached perfectly with the result shown in Fig 11.

d_1, d_2, and d_3 serve as the thicknesses for three crusts used in the proposed method. The discussion of these parameters setting and their impact is given as follows. d_1 is the thickness of the inner crust $C^P_{d_1}$. Under noise-free assumption, $d_1 > \min_{v\in P} h(v)$, where $h(v)$ is the sizing function defined in Section 2. This restriction means the $C^P_{d_1}$ shall connect any data point to at least another data point. On uniform and noise-free cases, we usually set $d_1 = 2\bar{h}$. d_2, as the thickness of the outer crust $C^P_{d_2}$, is only required be slightly larger than d_1 in continuous circumstance, usually set to $4\bar{h}$. But in discrete implementation, the difference between d_1 and d_2 is more important, which we will discuss in Section 3.2. d_3 is the thickness of the trimming crust $C^{Bd}_{d_3}$, which is required to be slightly larger than d_2, both in continuous and discrete circumstances. It is set to $5\bar{h}$ in the experiments. When facing a non-uniform data set, a single tuple of d_i parameters obviously is not enough. $\{d_i, i = 1, 2, 3\}$ shall vary according to the data density, which means the $\bar{h}$ shall be replaced by the local $h(v)$. This could be a difficult issue, which, however, will be tackled well in the discrete implementation.

It is worth noticing that Bd would be an empty set if the ground truth surface S is watertight. Therefore an empty crust $C^{Bd}_{d_3}$ is constructed and no boolean operation is done upon $C^P_{d_1}$ and $C^P_{d_2}$. In other words, the method in Section 2 is a special case of the proposed method. Various types of cases, including open, watertight, and hybrid surfaces, can be approached by a single algorithm without any a priori knowledge of surface topology or beforehand hole detections.

It is also worth mentioning that the surface falling in $C^{Bd}_{d_3}$ is designed to be abandoned. The loss in the reconstructed result is estimated to be comparable to d_3, which is negligible compared to the huge data set. We choose to sacrifice the portion of surface in $C^{Bd}_{d_3}$ to gain more robustness. The slightly loss could hardly be observed in the numerical examples.

3.2 The implementation of the method

In this subsection, we provide the discrete versions of the concepts involved in the above algorithm. This algorithm is implemented upon a tetrahedral mesh based on these discrete concepts, which is presented in detail in the Table 5. Before presenting these concepts, the establishment of the mesh framework is briefly introduced. Given a data point set P, background points Q are generated according to the local density of P. Usually, uniform or adaptive grid points are a good choice. Both P and Q are inserted into a tetrahedral mesh $\mathcal{T}_h$ in a Delaunay way. In the mesh $\mathcal{T}_h = (P \cup Q, C)$, $\{K_i\}_{i=1}^N \subset C$ are the tetrahedra and $\{F_i\}_{i=1}^L \subset C$ triangular faces.

Let the mesh and the crust be defined in the same way as in Section 2. The discrete distance between v_i and v_j is defined as $d_h(v_i, v_j) = \min_M\{M | v_j \in N^M_{v_i}\}$. Then the discrete distance between a vertex v and a vertex set V can be defined as $d_h(v, V) = \min_{x\in V} d_h(v, x)$. Further, given a surface triangulation Σ_h, the discrete medial axis can also be defined in two ways. The discrete medial axis in vertices $M_V = \{v | \exists u_1, u_2 \in \Sigma_h,$ $d_h(v, u_1) = d_h(v, u_2) = d_h(v, \Sigma_h)\}$. The discrete medial axis in triangular faces $M_F = \{F_i = \{u, v, w\} | u, v, w \in M_V\}$.

Notice that the d_1, d_2 and d_3 parameters are replaced by the discrete distance N_1, N_2 and N_3. As mentioned in previous subsection, for non-uniform data sets, the tuple of $\{d_i, i = 1, 2, 3\}$ shall vary according to the sizing function h. Recall the reasonable sized mesh we construct for the non-uniform data set. Since the

Table 5 Open surface reconstruction on a tetrahedral mesh

Inputs	A point set P
Algorithm	
1.	Generate background points Q according to the density of P
2.	Insert P and Q to a tetrahedral mesh $\mathcal{T}_h$ in a Delaunay way
3.	Build two crusts $K^P_{N_1}$ and $K^P_{N_2}$
4.	Find Σ, the boundary triangulation of $K^P_{N_2}$
5.	The medial axis of Σ in vertices M_V is found
6.	The medial axis of Σ in faces M_F is found
7.	The boundary of M_F is found B
8.	The vertices on the boundary B is to be found: $B_d = B \cap (P \cup Q)$
9.	Build a crust $K^{B_d}_{N_3}$ around B_d with $N_3 > N_2$
10.	Trim the two crusts around P: $\widetilde{K}^P_{N_1} = K^P_{N_1} - (K^{B_d}_{N_3} \cap K^P_{N_1})$, $\widetilde{K}^P_{N_2} = K^P_{N_2} - (K^{B_d}_{N_3} \cap K^P_{N_2})$
11.	Partition the region $\widetilde{K}^P_{N_2} - \widetilde{K}^P_{N_1}$ by region growing algorithms
12.	Construct a graph G dual to $\widetilde{K}^P_{N_2}$
13.	Apply graph-cuts on G and extract the surface S from the minimal cut
Outputs	The surface triangulation S

mesh element sizes conform with the local sizing function, the varying sizing function h has been included in the varying mesh size. Hence the discrete parameter $\{N_i, i = 1, 2, 3\}$ can be fixed globally. For example, instead of $\{d_1, d_2, d_3\} = \{2h, 4h, 5h\}$ with varying h, $\{N_1, N_2, N_3\} = \{2, 4, 5\}$. This largely facilitates the implementation.

Based on these definitions in a discrete language, the proposed algorithm can be effectively implemented on a tetrahedral mesh as described in Table 5. The underlying Delaunay-based mesh makes the resulting surface more likely to be homeomorphic to the ground truth. More examples are shown in Section 5 to demonstrate the effectiveness and robustness of the proposed method.

What is also worth mentioning is the difference between N_1 and N_2 should be paid special attention to. The discrete distance are measured by vertices and the subsequent region growing algorithm are conducted on tetrahedra. The region growing procedure could be easily obstructed if the cavity between $K^P_{N_1}$ and $K^P_{N_2}$ are too slim. The connected region in continuous circumstance would be detected as several disconnected regions in discrete mesh. The Armadillo example in Section 5 shows this situation.

4 Reconstruction of open surfaces based on domain decomposition

In Section 3, the open surface reconstruction method has been proposed, whose effectiveness and robustness will be shown in Section 5. The good performance on various kinds of surfaces leads to further consideration of its applications. One of the most significant applications is to reconstruct a surface based on domain decomposition. Domain decomposition has been successfully applied on computer vision field for a long time. One option is to use domain decomposition idea as preconditioners to get fast solvers for some related linear problems (Kohlberger et al, 2003, 2004, 2005). Some recent analysis reveals that domain decomposition can be used as a robust alternating minimization scheme between overlapped subspaces, see (Tai and Duan, 2009; Tai and Xu, 2002). In recent study, the dual graph could be subdivided into subgraphs as well to gain extra efficiency (Strandmark and Kahl, 2010). In surface reconstruction, the robustness and effectiveness of such kind of divide-and-conquer algorithms will strongly depend on a good reconstruction method for general surfaces, since the surface in a subdomain may be open or have disconnected interior. Hence, based on the method proposed in Section 3, we present a reconstruction method based on domain decomposition. Since the idea of parallel surface reconstruction is also very attractive, the method is designed in such a way that it can easily be adapted to parallel machines.

Another motivation is the incompetence of the method proposed in Section 3 on some special cases. As is known, all 2-manifolds without boundary in R^3, i.e. watertight surfaces, are orientable (Dey, 2007). The methods dedicated to watertight surfaces do not have to face the difficulty about non-orientability. However, the 2-manifolds with boundaries, i.e. open surfaces, may be nonorientable. This nonorientable surface problem would be a great challenge for those methods based on implicit representations. For instance, the method proposed in Section 3 cannot handle nonorientable surfaces such as Mobius strip. After the trimming operation, $\widetilde{C}^P_{d_1}$ may still fail to separate $\widetilde{C}^P_{d_2}$ into two or more subdomains. A surface reconstruction method based on domain decomposition would be helpful when facing this difficulty. Once the domain X has been decomposed properly, the surface piece in each subdomain is orientable, and can be approached by the method in Section 3. To the best of our knowledge, this study is the first to reconstruct non-orientable surfaces via graph-cuts.

4.1 Overlapping domain decomposition scheme

Given a domain $X \subset R^3$, a partitioning $\{X_i\}_{i=1}^N$ of X can be obtained according to a decomposition scheme. In practice, the decomposition scheme can be spatial oriented, feature oriented or data oriented. In this study,

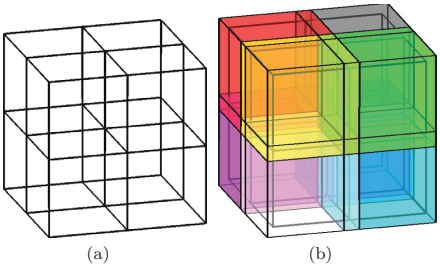


Fig. 12 Nonoverlapping and overlapping decomposition schemes

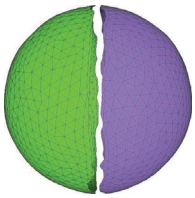


Fig. 13 A concrete example of cracks in non-overlapping decomposition scheme

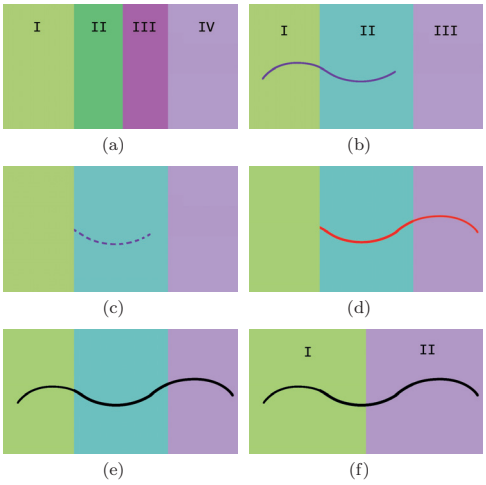


Fig. 14 A sequential fix-the-boundary method is presented to tackle cracks and conflicts. In (a), $X_i : I + II$; $X_j : III + IV$. $X_i^{'} : I + II + III$; $X_j^{'} : II + III + IV$. In (b)-(e), $X_i^{'} : I + II$; $X_j^{'} : II + III$; $X_{ij} : II$. In (f), $X_i : I$; $X_j : II$

a common spatial decomposition scheme is used. Obviously, any rectangular cuboid B can be decomposed into small tessellating rectangular cuboids $\{B_i\}_{i=1}^N$ as illustrated in Fig. 12(a). In our problem, by choosing B to be a rectangular cuboid properly bounding X, i.e. $X \subset B$, $\{X_i\}_{i=1}^N$ can be obtained through $X_i = X \cap B_i$. Notice that $\cup_{i=1}^N X_i = X$, $X_i \cap X_j = \emptyset$.

However, to avoid the cracks between subdomains as in Fig. 13, overlapping parts are necessary. In our study, an overlapping decomposition scheme could be obtained by expanding cuboid cells $\{B_i\}_{i=1}^N$ to $\{B_i'\}_{i=1}^N$ as shown in Fig. 12(b). A new partitioning with overlapping $\{X_i^{'}\}$ is then obtained. The surface reconstruction problem on P is decomposed into the sub-problems of $P_i^{'} = P \cap X_i^{'}$. To tackle the issue of possible conflicts and cracks in overlapping part, in this study, a sequential fix-the-boundary method is proposed. As a result, some parallel potential is lost due to the increasing interaction between neighboring subdomains, which can be compensated in some degree as explained later in this section.

4.2 Fix-the-boundary reconstruction method

Without loss of generality, it is assumed that the whole domain is decomposed into only two subdomains, i.e. X_i and X_j. The partitioning with overlapping is $X_i^{'}$ and $X_j^{'}$. The overlapping region is $X_{ij} = X_i^{'} \cap X_j^{'}$. Both these two partitioning systems are depicted in Fig. 14(a). The sequential algorithm begins from $X_i^{'}$. Once the partial data set $P_i^{'} = P \cap X_i^{'}$ is ready, the background points $Q_i^{'}$ for this subdomain are generated. Both $P_i^{'}$ and $Q_i^{'}$ are inserted into the tetrahedral mesh $\mathcal{T}_i$. Meanwhile the background points falling into the overlapping region, i.e. $Q_i^{'} \cap X_{ij}$ are stored. The graph-based method is applied and the reconstruction result $S_i^{'}$ is obtained as in Fig. 14(b). The reconstructed surface falling into the overlapping region, i.e. $S_i^{'} \cap X_{ij}$ is also stored for further use as in Fig. 14(c).

When the second subdomain $X_j^{'}$ is processed, one measure is taken upon the background points. After the background points $Q_j^{'}$ for $X_j^{'}$ are generated, the background points falling into the overlapping region are replaced by those background points stored in the $X_i^{'}$ stage: $Q_j^{'} = (Q_j^{'} - X_{ij}) \cup (Q_i^{'} \cap X_{ij})$. This operation ensures that the two subdomains contain the same background points in the overlapping region, i.e. $Q_i^{'} \cap X_{ij} = Q_j^{'} \cap X_{ij}$. The same data points and background points add up to an identical mesh point set in the overlapping region. Under the assumption of general positions, the Delaunay triangulation of a point set is unique. Combined with the local property of Delaunay triangulations, it is safe to assert that the meshes in the overlapping region from two subdomains are identical, i.e. $\mathcal{T}_i \cap X_{ij} = \mathcal{T}_j \cap X_{ij}$, which guarantees that $S_i^{'} \cap X_{ij} \subset \mathcal{T}_j \cap X_{ij}$.

Following the routine of the graph-based method, a graph is constructed dual to the mesh $\mathcal{T}_j$. Then we increase the weights of those edges corresponding to the stored faces $S_i^{'} \cap X_{ij}$, to a relatively large value. Through this adjustment, the surface $S_j^{'}$ reconstructed in $X_j^{'}$ is forced to coincide with $S_i^{'}$ in the overlapping region, i.e. $S_i^{'} \cap X_{ij} = S_j^{'} \cap X_{ij}$ as in Fig. 14(d). The surface in the overlapping region X_{ij} serves as the boundary of both $S_i^{'}$ and $S_j^{'}$. In the $X_i^{'}$ stage, the choice of the boundary of $S_i^{'}$ is relaxed. In the $X_j^{'}$ stage, the boundary of $S_j^{'}$, i.e. $S_j^{'} \cap X_{ij}$, will be fixed through the adjustment on the edge weight assignment. Hence conflicts and cracks can be avoided as in Fig. 14(e). We refer to this adjustment of the edge weight as "fix the surface in X_{ij}" for short.

Furthermore, some measures are taken to eliminate the redundant output of surface. Notice the curve in the overlapping region in Fig 14(e) has been outputted twice in two stages. This redundant output is harmless and can be eliminated by a trimming operation. After the surface piece $S_i^{'}$ in each subdomain is obtained, the non-overlapping decomposition X_i is used to trim the surface piece, i.e. $S_i = S_i^{'} \cap X_i$. The union of all trimmed surface pieces $S = \cup_{i=1}^{N} S_i$ is the final result, which is free of redundant output, cracks or conflicts as in Fig. 14(f). The whole divide-and-conquer algorithm is given in Table 6.

Table 6 Algorithm of surface reconstruction based on domain decomposition

Inputs	
1.	A point set P
2.	Partition of X, $\{X_i\}_{i=1}^{N}$
3.	Partition of X with overlapping, $\{X_i^{'}\}_{i=1}^{N}$
4.	$Neigh_i[N_i]$, $i = 1, \cdots, N$, N_i is the number of neighbors of X_i, and the array $Neigh_i$ stores N_i neighbors.
Algorithm	
1	Initialize a flag matrix $\{F_{ij}\} = 0$
2	Allocate storage for Q_{ij} background points in X_{ij}
3	Allocate storage for S_{ij} the surface in X_{ij}
4	For $i = 1 : N$
5	$P_i^{'} = P \cap X_i^{'}$
6	Generate $Q_i^{'}$ according to $P_i^{'}$
7	For $k = 1 : N_i$
8	$j = Neigh_i[k]$
9	if $F_{ij} == 1$
10	$Q_i^{'} = (Q_i^{'} - X_{ij}) \cup Q_{ij}$
11	else
12	$Q_{ij} = Q_{ji} = Q_i^{'} \cap X_{ij}$
13	End if
14	End For
15	Insert $P_i^{'}$ and $Q_i^{'}$ to generate the mesh $\mathcal{T}_i$
16	For $k = 1 : N_i$
17	$j = Neigh_i[k]$
18	if $F_{ij} == 1$
19	Fix all S_{ij} in X_{ij}
20	$F_{ij} = F_{ji} = 1$
21	End if
22	End For
23	Apply graph-based method and obtain $S_i^{'}$
24	Trim the surface piece $S_i = S_i^{'} \cap X_i$
25	End For
Outputs	The surface triangulation $S = \bigcup_{i=1}^{N} S_i$

4.3 Parallel efficiency regained

As mentioned, some parallel potential is lost due to the interaction between subdomains in this method. Two neighboring subdomains cannot be processed simultaneously. As in the example of Fig. 14, the subdomain $X_j^{'}$ cannot be processed until $S_i^{'} \cap X_{ij}$ is obtained. To adapt this method to parallel machines, it would be helpful to color all subdomains at the beginning so that no neighboring subdomains have the same color. Then the group of subdomains sharing the same color can be processed simultaneously because of the independence between any two of them. This coloring preprocessing turns the sequential algorithm in Table 6 to a parallel algorithm. However the coloring strategy and the number of colors required determine the parallel efficiency.

In two dimensional problems such as image segmentation (Hodneland et al, 2009), the well known four-color theorem can limit the number of the colors required within four. Unfortunately, there is no such theoretic bound of the number of colors required in three dimensions. However, for some special cases, we still can figure out the number of colors required. For the rectangular cuboids decomposition scheme described above and the underlying 26-neighborhood system, it can easily be shown that only eight colors are required for a neighbor-different coloring. An example of $5 \times 5 \times 5$ decomposed cube's 8 coloring scheme is shown in Fig. 15. The parallel efficiency of these decomposition cases is still high even with the dependence between subdomains.

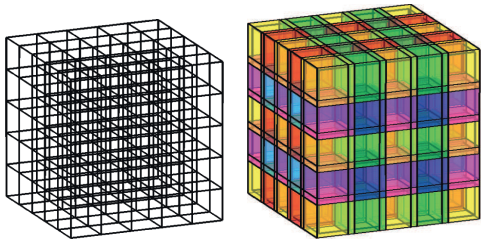


Fig. 15 Eight Coloring Scheme

In this section, a new reconstruction method based on domain decomposition was proposed. Interaction between subdomains was introduced to eliminate possible cracks and conflicts. Though this interaction between subdomains requires a sequential algorithm, a proper decomposition manner as well as a coloring preprocessing allows for parallel algorithms. Some examples approached by this decomposition based method are included in Section 5 to show its robustness and effectiveness.

5 Examples

In this section, various examples are presented to demonstrate the efficiency and robustness of our method as well as the quality and faithfulness of reconstructed surfaces. All experiments had been conducted on a desktop PC with Intel Pentium 4 CPU of 3.2GHz. Most models were obtained from Stanford 3D Scanning Repository, Large Geometric Models Archive of Georgia Institute of Technology and Digital Shape Workbench Project while the others were synthesized by ourselves. We applied Computational Geometry Algorithms Library (cga, 1997) in our program. All surfaces are rendered by MeshLab. Only points locations were utilized in the algorithm. Based on the properties and purposes of theirs, these examples can be categorized into four groups: simple open surfaces, complicated (general) surfaces, watertight surface approached by domain decomposition, and non-orientable surfaces approached by domain decomposition.

5.1 Simple open surfaces

Simple open surfaces generally refer to manifolds with boundaries. As the initial motivation of this study, several examples of the simple open surfaces are demonstrated in Fig. 16, 17, and 18 including the data point sets and the reconstructed surfaces. Two human faces, one representative category of open surfaces, are faithfully reconstructed. The front views show the well preserved features and the back or bottom view shows the boundaries of reconstructed surfaces. The other example, a hand, is presented as well. All these three examples can be seen as the application on incomplete data. After all, it is hardly possible to obtain watertight models of human body parts by a 3D laser scanner.

5.2 Complicated (General) surfaces

Since simple open surface cases can be approached perfectly, the proposed method is challenged by some more complicated examples: multi-phase open surfaces, hybrids of open and watertight surfaces, and open surfaces with noises or outliers. To sum up, this subsection presents surface examples which are more general and occur ubiquitously in daily life.

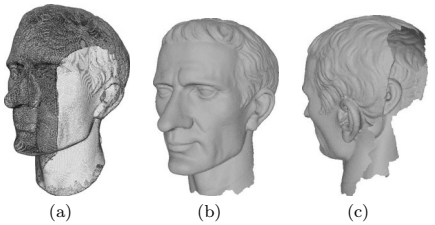


Fig. 16 Julius Caesar

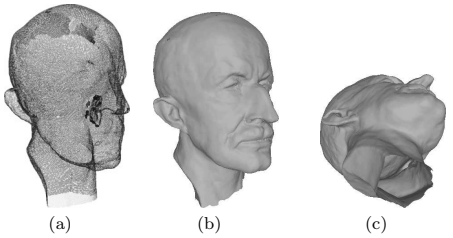


Fig. 17 Max Planck

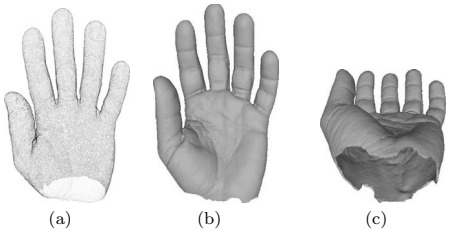


Fig. 18 A hand

Multi-phase open surfaces do not have to separate the domain into more than two regions. Multi-phase means the trimmed crust $\widetilde{C}^P_{d_2}$ is partitioned by $\widetilde{C}^P_{d_1}$ into more than two regions. These cases may involve intersections or not. Multi-phase cases without intersections, i.e. disconnected surface patches, are still 2-manifolds with boundaries and apparently no challenge to the proposed method. Furthermore, an example of two intersecting semi-spheres, which is no longer 2-manifold, is shown in Fig. 19, from which we can see that all features of the intersecting parts are reconstructed faithfully.

The above example can be seen as a union of two 2-manifolds, both of which have boundaries. Next pre-

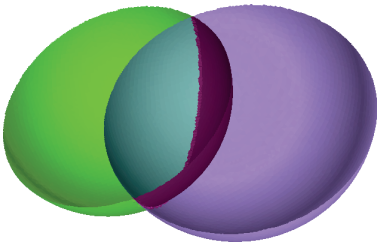


Fig. 19 A multi-phase open surface example: two intersecting semi-spheres

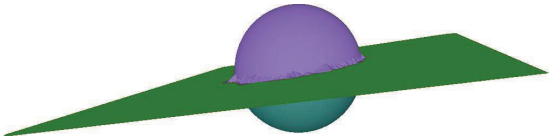


Fig. 20 A hybrid of a watertight surface and an open one: a rectangle intersecting a sphere

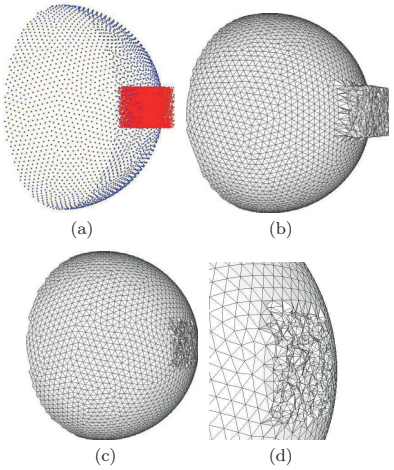


Fig. 21 The noisy case of a semi-sphere

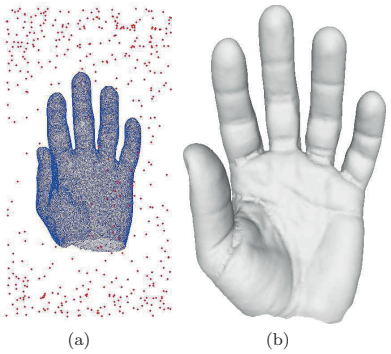


Fig. 22 A hand example with outliers

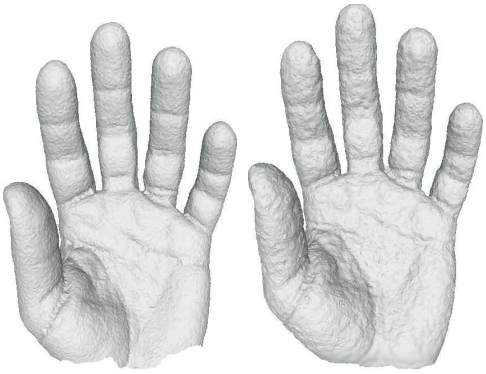
sented is a union of two 2-manifolds, one of which has boundaries while the other has not, i.e. a union of a watertight surface and an open one. The reconstruction result of a rectangle intersecting a sphere is shown in Fig. 20, from which we can see that both the sphere and the rectangle have been reconstructed faithfully. From a technical point of view, this example has nothing special compared to the one in Fig 19. It becomes, however, more meaningful after post-processing. The watertight sphere surface can be thought of as the boundary of a 3-manifold ball. Once the domain bounded by the sphere is volumetrically meshed, the union of the 3-manifold and the 2-manifold can be represented discretely by a triangular and tetrahedral mixed mesh. This issue ubiquitously occurs in animations, medical applications, and CAD industries.

The next example is an open surface with noise. The noises in real world may be introduced during the data acquisition procedure. In this study, the noise is added artificially. The data set in blue as well as the noise in red is shown in Fig. 21(a). This distinguishing coloring scheme is only for clear demonstration and the algorithm treats data and noise as a whole input. Fig 21(b) and (c) show results with regularization coefficient $\alpha = 0$ and 0.001 respectively. Readers can compare our results to the result of the explicit methods in Fig. 1. The noise removal result with $\alpha = 0.001$ is zoomed and shown in Fig 21(d).

Then the example of an open surface with outliers is shown. The input of 53,054 blue data points sampled from a hand and 447 red artificial outliers are shown in Fig. 22(a). The clean reconstructed result in Fig. 22(b) shows that our method is also robust to the difficulty of outliers.

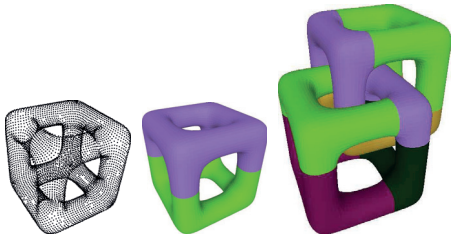
Fig. 23, the last example in this subsection, is radnom Gaussian noises added on each points in the hand data set. Results under two levels of noises are shown. The noise in Fig. 23(a) is pertubed by a Gaussian noise of $0.3\bar{h}$ standard deviation and in (b) $0.6\bar{h}$.

By Fig. 21-23, we demonstrate the robustness of our method to noises and outliers. The reason lies in the fact that only the precision of estimated boundary matters to the final result. Thus the intermediate steps could tolerate the distortion. As to the noises and outliers near the boundary, we leave more discussions to Section 6.



(a) Noise of $0.3\bar{h}$ (b) Noise of $0.6\bar{h}$

Fig. 23 The hand model with Gaussian noises



(a) (b) (c)

Fig. 24 Perforated cubes approached by domain decomposition

5.3 Watertight surfaces approached by domain decomposition

Surface reconstruction based on domain decomposition is an important application of the open surface reconstruction method. In this subsection, some watertight cases, which had been approached by previous graph-cuts methods, are used to test the effectiveness of the decomposition based reconstruction method, especially the overlapping and interface part.

Fig. 24 shows that two cube-based objects are reconstructed in a domain decomposition way. The perforated cube is reconstructed in two subdomains as indicated by different colors in Fig 24(b). Similarly, two tangling perforated cubes in Fig 24(c) has been approached in eight subdomains, each of which contains multiple disconnected surface patches. These are same to the results obtained by previous methods.

Next three classic examples, armadillo, horse, and dragon, are shown in Fig. 25. The colorfulness of armadillo is used to illustrate the relationship between the choice of thickness parameters, d_1 and d_2, and the multiphase issue. Once we increase the difference between d_1 and d_2, the colorfulness disappears gradually as the phase number decreases, which is shown in the horse and dragon examples.

At last of this subsection, three statuettes are shown in Fig 26. From left to right, the statuettes are reconstructed in four, two and three subdomains respectively. Through this subsection, the absence of undesirable conflicts and cracks proves the effectiveness of our method.

Table 7 Statistics of open surface examples

Example	Data Set	Mesh Generation Time	Graph Built Time	Graph Cut Time
Caesar	387900	248.4952	7.48386	22.3256
Planck	199169	96.7383	5.2332	9.8995
Hand	53054	41.37467	1.16082	8.49171

5.4 Nonorientable Surfaces

As mentioned, when the research area is extended to the open surfaces, i.e. 2-manifolds with boundaries, the nonorientable issue becomes a problem for all implicit methods. In this subsection, Mobius strip, one motivation of this decomposition based method, is approached perfectly with the result shown in Fig. 27. Another famous nonorientable surface, Klein bottle, is also presented in Fig. 28.

Table 7 gives the sizes of the data sets of several open surface examples and corresponding CPU time counted in seconds. The first column gives the examples' names. The second column contains the numbers of data points P. The third column is the mesh generation time, the fourth the graph construction time, and the fifth the graph cut time. In Table 8 included are sizes and time of the domain decomposition examples. Each block contains the statistics of every subdomain as well as those in total.

6 Conclusion

In this article, a variational reconstruction method for open surface is proposed based on Delaunay triangulation and graph-cuts. In the proposed method, the graph is constructed dual to the mesh in a restricted region obtained after crust establishments and boolean operations, by which the open surface problem in the

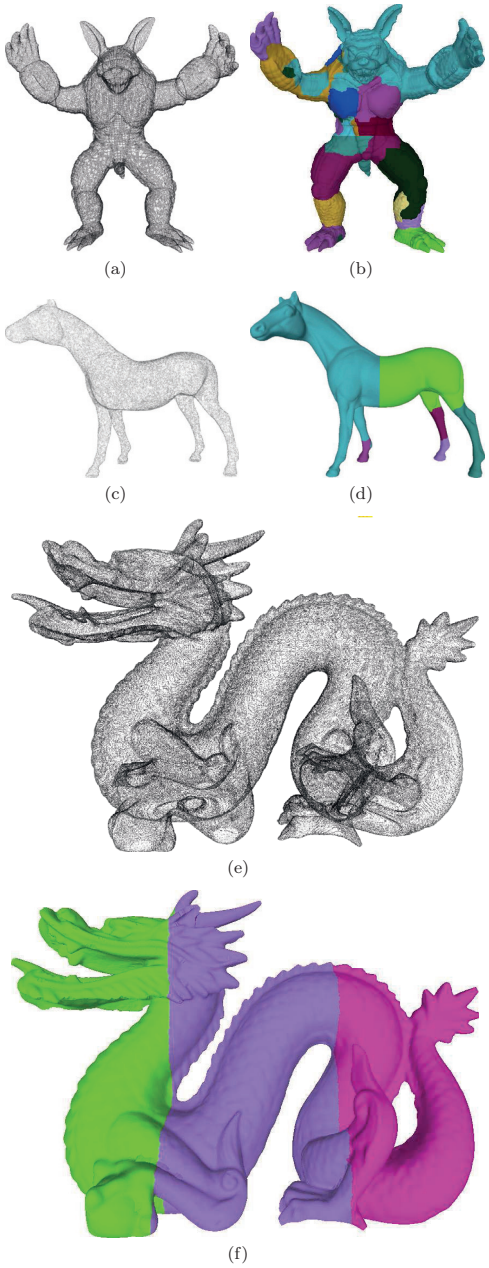


Fig. 25 Three classical examples approached by domain decomposition

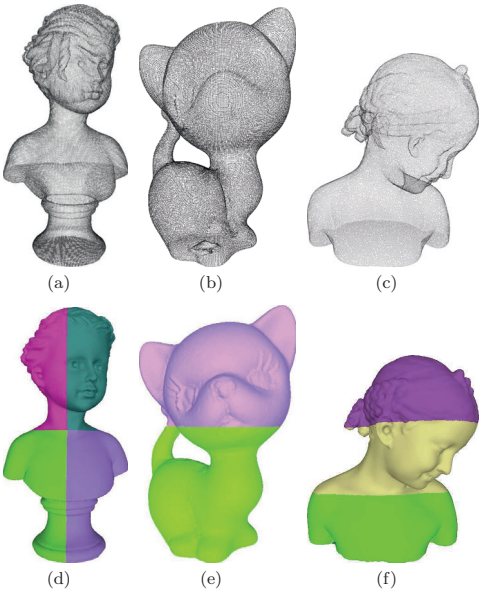


Fig. 26 Three statuettes approached in different decomposition schemes

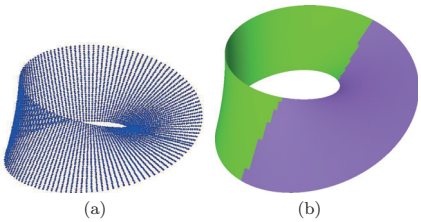


Fig. 27 Mobius strip approached by domain decomposition

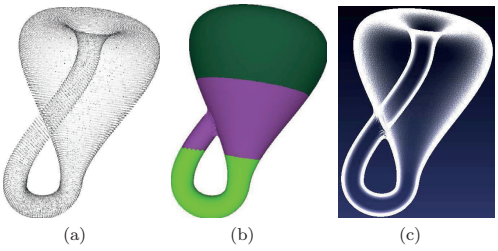


Fig. 28 Klein bottle approached by domain decomposition

Table 8 Statistics of domain decomposition examples

Example	Data Set	Mesh Generation Time	Graph Built Time	Graph Cut Time
Armadillo Total	172974	191.58093	4.74328	3.936398
Subdomain 1	76525	73.61781	1.78773	3.06553
Subdomain 2	119073	117.96312	2.95555	0.870868
Horse Total	494195	843.2701	16.12855	12.67107
Subdomain 1	300357	315.7326	7.78682	5.00724
Subdomain 2	299595	527.5375	8.34173	7.66383
Dragon Total	437645	605.4991	15.40466	2.239658
Subdomain 1	190871	259.2044	4.87826	0.151977
Subdomain 2	265931	346.2947	6.66699	1.9627
Subdomain 3	155873	194.8477	3.85941	0.124981

whole domain has been translated to a watertight surface problem in a restricted region. The phase detection based on region growing algorithms hence can be applied and so can the graph techniques.

Furthermore, a surface reconstruction method based on domain decomposition is presented as an important application of open surface reconstruction. First, the domain decomposition is a powerful tool to tackle the difficulty in non-orientable surfaces. Since locally all the non-orientable surfaces can be decomposed into orientable patches, the reconstruction based on domain decomposition would be an effective approach without loss of generality. Another motivation is parallel surface reconstruction. The overlapping decomposition scheme as well as our fix-the-boundary method could effectively eliminates the cracks and conflicts.

However, the independence between subdomains are sacrificed to eliminate the conflicts and cracks. This loss of parallel potential may be largely compensated if we adopt proper decomposition scheme. By the decomposition scheme in this paper, if the numbers of processing units and subdomains are both larger than eight, the parallel efficiency is as high as a subdomain-independent algorithm. Parallel implementation of this domain decomposition method and investigation of its efficiency is one of our future research interests.

We assume all data sets in this paper are sufficiently sampled. To some extent our method is subject to the difficulty of undersampling. The crust defined in this paper may still fail to separate the domain when facing severely undersampling cases. Possible solutions such as dilation of data points are still under investigation.

Besides, our method is subject to noises and outliers near the boundary as mentioned in Section 5.2. Because the trimming operation is crucial to the success of converting open cases to watertight cases. When facing the difficulty such noises and outliers around the boundary, possible filtering could be performed on the rawly detected boundary before the trimming operation. This is also one of our future research interests.

Acknowledgements The authors would like to thank the referees for valuable suggestions on the improvement of the paper. This research is sponsored by Sinapore MOE ARC 29/07 T207B2202, MOE RG 59/08 M52110092, and NRF 2007IDM-IDM 002-010. The CRUST code (Giaccari, 2010) from the MATLAB community is used to obtain the result in Fig. 1 and we appreciate the code writer.

References

(1997) CGAL, Computational Geometry Algorithms Library. Http://www.cgal.org

Adams R, Bischof L (1994) Seeded region growing. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(6):641–647

Adamy U, Giesen J, John M (2000) New techniques for topologically correct surface reconstruction. In: Proceedings of the conference on Visualization'00, IEEE Computer Society Press Los Alamitos, CA, USA, pp 373–380

Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva C (2001) Point set surfaces. In: IEEE Visualization, vol 1, pp 21–28

Amenta N, Bern M, Kamvysselis M (1998) A new Voronoi-based surface reconstruction algorithm. In: Proceedings of the 25th annual conference on Computer graphics and interactive techniques, ACM New York, NY, USA, pp 415–421

Amenta N, Choi S, Dey T, Leekha N (2000) A simple algorithm for homeomorphic surface reconstruction. In: Proceedings of the sixteenth annual symposium on Computational geometry, ACM New York, NY, USA, pp 213–222

Bae E, Tai XC (2009) Graph cut optimization for the piecewise constant level set method applied to multiphase image segmentation. In: Tai XC, Morken K, Lysaker M, Lie KA (eds) SSVM, Springer, Lecture Notes in Computer Science, vol 5567, pp 1–13

Bertalmío M, Sapiro G, Randall G (1999) Region tracking on surfaces deforming via level-sets methods. Lecture notes in computer science pp 330–338

Boissonnat J, Cazals F (2000) Smooth surface reconstruction via natural neighbour interpolation of distance functions. In: Proceedings of the sixteenth annual symposium on Computational geometry, ACM New York, NY, USA, pp 223–232

Boykov Y, Kolmogorov V (2004) An experimental comparison of min-cut/max-flow algorithms for energy

minimization in vision. IEEE Transactions on Pattern Analysis and Machine Intelligence pp 1124–1137

Burchard P, Cheng L, Merriman B, Osher S (2001) Motion of curves in three spatial dimensions using a level set approach. Journal of Computational Physics 170(2):720–741

Caselles V, Kimmel R, Sapiro G (1997a) Geodesic active contours. International journal of computer vision 22(1):61–79

Caselles V, Kimmel R, Sapiro G, Sbert C (1997b) Minimal surfaces based object segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(4):394–398

Cheng L, Burchard P, Merriman B, Osher S (2002) Motion of curves constrained on surfaces using a level-set approach. Journal of Computational Physics 175(2):604–644

Curless B, Levoy M (1996) A volumetric method for building complex models from range images", SIGGRAPH96. In: Computer Graphics Proceedings

Dey T (2007) Curve and surface reconstruction: algorithms with mathematical analysis, Cambridge Univ Pr, pp 6–7

Dey TK, Goswami S (2003) Tight cocone: a water-tight surface reconstructor. In: SM '03: Proceedings of the eighth ACM symposium on Solid modeling and applications, pp 127–134

Du Q, Wang D (2002) Tetrahedral mesh generation and optimization based on centroidal Voronoi tesselations. Int J Numer Meth Eng 56:1355–1373

Edelsbrunner H, Mucke E (1992) Three-dimensional alpha shapes. In: Proceedings of the 1992 workshop on Volume visualization, ACM New York, NY, USA, pp 75–82

Faugeras O, Gomes J (2000) Dynamic shapes of arbitrary dimension: the vector distance functions. In: Proceedings of the Ninth IMA Conference on Mathematics of Surfaces. The Mathematics of Surfaces IX, Springer, Citeseer

Franchini E, Morigi S, Sgallari F (2010a) Implicit shape reconstruction of unorganized points using PDE-based deformable 3D manifolds. Numerical Mathematics: Theory, Methods and Applications

Franchini E, Morigi S, Sgallari F (2010b) Segmentation of 3D Tubular Structures by a PDE-Based Anisotropic Diffusion Model. Mathematical Methods for Curves and Surfaces pp 224–241

George P, Borouchaki H (1998) Delaunay triangulation and meshing: application to finite elements. Kogan Page

Giaccari L (2010) Surface reconstruction from scattered points cloud part1. URL http://www.mathworks.com/matlabcentral/fileexchange/22185, [Online; accessed 22-10-2010]

Gray R, Neuhoff D (2002) Quantization. Information Theory, IEEE Transactions on 44(6):2325–2383

Hodneland E, Tai X, Gerdes H (2009) Four-Color Theorem and Level Set Methods for Watershed Segmentation. International Journal of Computer Vision 82(3):264–283

Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W (1992) Surface reconstruction from unorganized points. In: SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques, pp 71–78

Hornung A, Kobbelt L (2006a) Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol 1

Hornung A, Kobbelt L (2006b) Robust reconstruction of watertight 3d models from non-uniformly sampled point clouds without normal information. In: Geometry Processing 2006: Fourth Eurographics Symposium on Geometry Processing, Cagliari, Sardinia, Italy, June 26-28, 2006, Eurographics, p 41

Ishikawa H (2003) Exact optimization for markov random fields with convex priors. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(10):1333–1336

Kohlberger T, Schnörr C, Bruhn A, Weickert J (2003) Domain decomposition for parallel variational optical flow computation. Pattern Recognition pp 196–203

Kohlberger T, Schnörr C, Bruhn A, Weickert J (2004) Parallel variational motion estimation by domain decomposition and cluster computing. Computer Vision-ECCV 2004 pp 205–216

Kohlberger T, Schnörr C, Bruhn A, Weickert J (2005) Domain Decomposition for Nonlinear Problems: A Control-Theoretic Approach. Technical Report, Computer Science Series

Kolmogorov V, Zabin R (2004) What energy functions can be minimized via graph cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence 26(2):147–159

Kuo C, Yau H (2005) A Delaunay-based region-growing approach to surface reconstruction from unorganized points. Computer-Aided Design 37(8):825–835

Labelle F, Shewchuk J (2007) Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. In: ACM SIGGRAPH 2007 papers, ACM, p 57

Lempitsky VS, Boykov Y (2007) Global optimization for shape fitting. In: CVPR, IEEE Computer Society

Lie J, Lysaker M, Tai X (2006) A variant of the level set method and applications to image segmentation. Mathematics of computation 75(255):1155–1174

Ohtake Y, Belyaev A, Alexa M, Turk G, Seidel H (2005) Multi-level partition of unity implicits. In: International Conference on Computer Graphics and Interactive Techniques, ACM New York, NY, USA

Osher S, Fedkiw R (2002) Level set methods and dynamic implicit surfaces. Springer Verlag

Paris S, Sillion F, Quan L (2006) A surface reconstruction method using global graph cut optimization. International Journal of Computer Vision 66(2):141–161

Smereka P (2000) Spiral crystal growth. Physica D: Nonlinear Phenomena 138(3-4):282–301

Solem J, Heyden A (2004) Reconstructing open surfaces from unorganized data points. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society; 1999, vol 2

Solem J, Heyden A (2006) Reconstructing open surfaces from image data. International Journal of Computer Vision 69(3):267–275

Solem J, Kahl F (2004) Surface reconstruction from the projection of points, curves and contours. In: 2nd Int. Symposium on 3D Data Processing, Visualization and Transmission, Thessaloniki, Greece

Solem J, Kahl F (2005) Surface reconstruction using learned shape models. Advances in Neural Information Processing Systems 17:1

Solem J, Overgaard N (2005) A geometric formulation of gradient descent for variational problems with moving surfaces. In: International Conference on Scale Space and PDE Methods in Computer Vision, Springer, pp 419–430

Strandmark P, Kahl F (2010) Parallel and distributed graph cuts by dual decomposition. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, pp 2085–2092

Tai X, Duan Y (2009) Domain decomposition methods with Graph cuts algotithms for image segmentation. UCLA CAM Report pp 9–54

Tai X, Xu J (2002) Global and uniform convergence of subspace correction methods for some convex optimization problems. Mathematics of Computation 71(237):105–124

Wan M, Wang Y, Wang D (to appear) Variational Surface Reconstruction Based on Delaunay Triangulation and Graph Cut. Int J Numer Meth Eng DOI 10.1002/nme.2965

Ye J, Bresson X, Goldstein T, Osher S (2010) A Fast Variational Method for Surface Reconstruction from Sets of Scattered Points. UCLA CAM Report

Yu Y (1999) Surface reconstruction from unorganized points using self-organizing neural networks. In: proceedings of IEEE Visualization, Citeseer, vol 99, pp 61–64

Zhao H, Osher S, Fedkiw R (2001) Fast surface reconstruction using the level set method. In: Proceedings of the IEEE Workshop on Variational and Level Set Methods (VLSM'01), IEEE Computer Society Washington, DC, USA, p 194

Zhao M (2000) Implicit and nonparametric shape reconstruction from unorganized data using a variational level set method. Computer Vision and Image Understanding 80(3)