TOPOLOGICALLY ENHANCED SLICING OF MLS SURFACES

Pinghai Yang **Kang Li** **Xiaoping Qian** †

DEPARTMENT OF MECHANICAL, MATERIALS AND
AEROSPACE ENGINEERING
ILLINOIS INSTITUTE OF TECHNOLOGY
CHICAGO, IL, 60616,USA

ABSTRACT

Growing use of massive scan data in various engineering applications has necessitated research on point-set surfaces. A point-set surface is a continuous surface defined directly with a set of discrete points.

This paper presents a new approach that extends our earlier work on slicing point-set surfaces into planar contours for rapid prototyping usage. This extended approach can decompose a point-set surface into slices with guaranteed topology. Such topological guarantee stems from the use of Morse theory based topological analysis of the slicing operation.

The Morse function for slicing is a height function restricted to the point-set surface, an implicitly defined moving least-squares (MLS) surface. We introduce a Lagrangian multiplier formulation for critical point identification from the restricted surface. Integral lines are constructed to form Morse-Smale complex and the enhanced Reeb graph. This graph is then used to provide seed points for forming slicing contours, with the guarantee that the sliced model has the same topology as the input point-set surface. The extension of this approach to degenerate functions on point-set surface is also discussed.

Keywords: Morse theory, moving least-squares, Reeb graph, Morse-Smale complex, slicing

INTRODUCTION

Growing use of massive scan data in diverse engineering applications has necessitated research in efficient and robust algorithms for processing scanned data. A point-set surface is a computer representation of a continuous surface defined directly with a set of discrete points [3][5][19], a canonical representation of data output from various 3D scanning systems. The moving least-squares (MLS) based point-set surfaces are defined implicitly by a projection operator. The MLS surface in essence provides a weighted average of data points with the weight varying from a target point based on a Gaussian function. It has been used in a host of graphics [1] and shape modeling [22] applications. We have successfully applied the MLS surface in applications such as computer-aided design [29], rapid prototyping [28], and NC machining [30].

This paper presents a new approach that extends our earlier work on slicing point-set surfaces into planar contours for rapid prototyping usage. This extended approach can decompose a point-set surface into slices with guaranteed topology. The slicing algorithm [28][29] forms the planar contour essentially by tracing plane-MLS surface intersection points. The tracing is through successive marching along the intersection curve. Starting points are required for the marching process to avoid any missing loops. Our earlier work assumes starting points can be properly found. This paper presents a topologically enhanced slicing approach that can guarantee the handling of topological events during the marching, such as loop merging, splitting, appearance and disappearance in each planar contour.

Such topological guarantee stems from the use of Morse theory based topological analysis of the slicing operation. The overall approach is shown in Figure 1, where a double torus is sliced. The Morse function for slicing is a height function restricted to the point-set surface, an implicitly defined MLS surface. We introduce a Lagrangian multiplier formulation for identification and classification of critical points on such a restricted surface. These critical points signal topological changes of the intersection curves between slicing planes and the point-set surface, including the creation of a new closed contour, the merge of two closed contours, the split of a closed contour, and the removal of a closed contour. Integral lines are constructed to form Morse-Smale complex, from which the redundant arcs are then pruned to form an enhanced Reeb graph. Such a topological graph provides necessary topological information for slicing the point-set surface, including the starting points for contour marching, and the knowledge of contour appearance and disappearance at different heights. The extension of this approach to degenerate functions on point-set surface is also discussed.

†Corresponding author. Email: qian@iit.edu

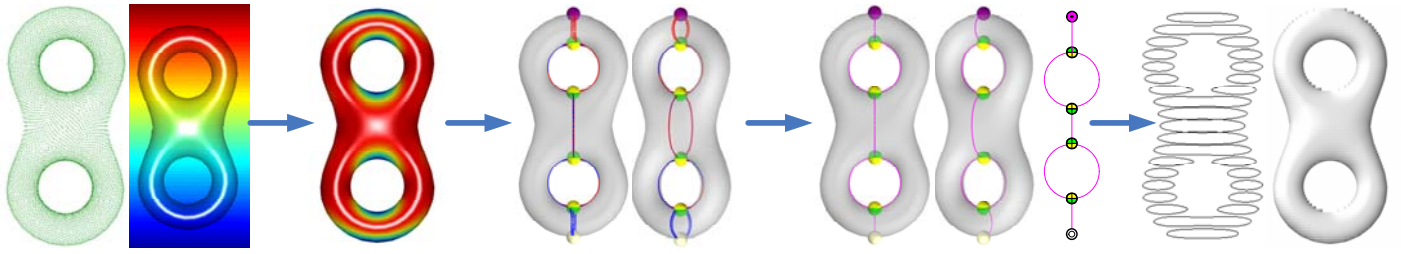


Figure 1. OVERVIEW OF THE PROPOSED APPROACH. (A) POINT CLOUD. (B) MORSE FUNCTION ON THE MLS SURFACE. (C) CRITICAL POINT GENERATION. (D) MORSE-SMALE COMPLEX. (E) ENHANCED REEB GRAPH. (F) SLICED MODEL. THE MAGENTA, GREEN & YELLOW AND WHITE DOTS REPRESENT THE MAXIMUM, (TOP AND BOTTOM) SADDLE AND MINIMUM CRITICAL POINTS RESPECTIVELY. FOR DETAILS PLEASE REFER TO THE LEGEND SHOWN IN FIUGRE 4.

The novelty of this work includes the following.

- A Lagrange multiplier approach is developed for indentifying critical points, where the Morse function is restricted to an implicit moving least-squares (MLS) surface. This ensures all critical points of a Morse function, under proper sampling conditions, lie on the restricted surface (i.e. the MLS surface). To the authors' best knowledge, this is the first reported work on extracting critical points of Morse functions on restricted surfaces.
- A method for constructing Morse-Smale complex on the MLS surface is presented. Due to the use of analytical equations for differential geometrical analysis on the MLS surface, it results in efficient and accurate construction of the complex.
- A method for robust intersection of a point-set surface with a set of parallel planes through enhanced Reeb graph is introduced. It guarantees the sliced model has the same topology with the input point-set surface. This enhanced Reeb graph is constructed from the Morse-Smale complex by pruning the redundant arcs. This use of Reeb graph to guide slicing is novel and it contrasts with the typical scenarios involving Reeb graph where the Reeb graph is generated from model slicing [17],[26].

LITERATURE REVIEW

MLS Surface

MLS based point-set surfaces have proven to be powerful and convenient in point based geometric processing and graphics applications [18],[19]. Salient advantages of such MLS surfaces include its ability of handling noise, up-sampling, down-sampling, etc. Moreover, based on a more general definition of a projection process [3],[5],[6], a mathematical proof of the convergence of the projection procedure has been presented [9],[10]. The resulting MLS surface is proven to be isotopic to the original sampled surface. Analytical equations for directly computing surface curvatures from MLS surfaces are available in [27].

A method for detecting ridges and valleys has also been developed, based on a modified MLS approximation technique to estimate the local differential information [16]. This method is similar to the one by Ohtake [21] but it gains better computation efficiency due to the use of local MLS approximation as opposed to the global RBF approximation in [21]. However both methods were implemented only on triangular meshes and the extracted feature lines are disconnected. In our approach, the Morse complex paths are connected at critical points and built directly on the continuous MLS surface.

Morse Theory

Morse theory studies how the critical points of a function are related to the topological shape of the manifold on which the function is defined. Over the past several decades, it has become a versatile tool for examining topological features of manifolds of interest in a variety of applications.

Helman and Hesselink [14],[15] proposed methods to simplify the visualization of two- and three-dimensional vector field topology. Critical points are first identified as loci where vector flow vanishes and a group of tangent curves connecting the critical points are then used to decompose the vector flow to form a simplified representation of the data.

A theoretical framework of classical Morse theory has been extended in [12] to the domain of implicit surfaces; this extended Morse theory is useful in detecting the global topological characteristics of implicit surfaces and thus very desirable for real time manipulation during interactive modeling process. However, it does not address critical points of constrained Morse functions.

Critical points sometimes may take other form. Algorithms have been developed to compute intersection curves of parametric surface with plane sets by using normal surfaces and topology transition points; these points function quite similarly as critical points.

Morse theory has also been be extended to discrete piecewise linear manifold as well. The underlying triangular mesh of the PL manifold carries all the topology information to generate hierarchical Morse-Smale complex [11].

This paper uses Morse theory to construct Morse-Smale complex to generate enhanced Reeb graph for a topologically guaranteed slicing. It is worth noting that our Morse-Smale

complex is different from that in [11] because it is pre-completely based on pure point set data without any already existing topology information such as polygonized mesh.

TOPOLOGICAL ANALYSIS FOR MORSE FUNCTIONS DEFINED ON MLS SURFACES

In this section, we present the pipeline of the topological analysis for Morse functions (i.e. height functions) for slicing defined on MLS surfaces. It includes four steps: 1) Critical point identification with a Lagrangian multiplier formulation; 2) Critical point classification with the bordered Hessian; 3) Morse-Smale complex construction through a marching process along the integral lines and restricted on a MLS surface; 4) Enhanced Reeb graph extraction by pruning the Morse-Smale complex.

Introduction of projection based MLS surfaces

Levin [18],[19] defined an MLS surface M as the stationary set of a projection operator ψ_P, i.e.,

$$M = \left\{ \mathbf{x} \in R^3 \mid \psi_P(\mathbf{x}) = \mathbf{x} \right\}. \tag{1}$$

Such projection based MLS surfaces are referred to as projection MLS surfaces. Amenta and Kil [5],[6] further proved that the MLS surface M is actually an implicit surface given by the zero-level set of the implicit function

$$g(\mathbf{x}) \equiv \mathbf{n}(\mathbf{x})^T \left(\frac{\partial e(\mathbf{y}, \mathbf{n}(\mathbf{x}))}{\partial \mathbf{y}} \Big|_{\mathbf{y}=\mathbf{x}} \right) = 0 . \tag{2}$$

In Eqn. (2), $\mathbf{n} : R^3 \to R^3$ represents a vector field defined as

$$\mathbf{n}(\mathbf{x}) = \frac{\sum_{\mathbf{q}_i \in \mathbf{Q}} \mathbf{v}_i \theta(\mathbf{x}, \mathbf{q}_i)}{\left\| \sum_{\mathbf{q}_i \in \mathbf{Q}} \mathbf{v}_i \theta(\mathbf{x}, \mathbf{q}_i) \right\|}, \tag{3}$$

where $\mathbf{v}_i$ are normal vectors assigned to each point $\mathbf{q}_i \in R^3$ of an input point set $\mathbf{Q}$ and $\theta(\mathbf{x}, \mathbf{q}_i)$ is a Gaussian weighting function. And $e : R^3 \times R^3 \to R$ represents an energy function defined as

$$e(\mathbf{y}, \mathbf{n}(\mathbf{x})) = \sum_{\mathbf{q}_i \in \mathbf{Q}} \left((\mathbf{y} - \mathbf{q}_i)^T \mathbf{n}(\mathbf{x}) \right)^2 \theta(\mathbf{y}, \mathbf{q}_i) . \tag{4}$$

An example of such an MLS surface and the corresponding point cloud of a double torus are shown in Figure 1(a) and (b).

Identification of Critical Points

In Morse theory, the identification of critical points for a given Morse function is the first and key step for topological analysis. Giving a smooth Morse function $f : R^3 \to R, f \in C^2$, the definition of its critical points is quite straight-forward: a point $\mathbf{x} \in R^3$ is a critical point if and only if its derivatives vanish:

$$\nabla f(\mathbf{x}) \equiv \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z} \right)^T = 0 .$$

Based on this definition, the problem of critical point identification has been addressed by many previous researchers [12]. However, suppose the Morse function f is a scalar function with a restriction, i.e., $f_M := f\big|_M$ defined on a smooth, compact 2-manifold M. Then identifying its critical points becomes more complicated since it has become a constrained problem. Some researchers have solved this problem for a special case that M is represented by a parametrical surface [4].

In this section, we introduce a Lagrangian multiplier approach for critical point identification, where M is represented by an implicit surface, i.e., the MLS surface defined by Eqn. (2). In this approach, we adopt height functions as the Morse function f, which can be represented as $f(\mathbf{x}) = Ax + By + Cz$. One special case of these height functions is $f(\mathbf{x}) = z$, as shown in Figure 1(b). However, the method is applicable for other smooth scalar functions, e.g., blobby functions introduced in [12].

We wish to determine the critical points of the function $f_M := f\big|_M$ which is the restriction of f. For a multiplier λ, we consider the Lagrangian function on the manifold $M \equiv g(\mathbf{x}) = 0$, that is,

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda g(\mathbf{x}).$$

The critical points are obtained by solving the equations

$$\nabla L(\mathbf{x}, \lambda) = 0 . \tag{5}$$

where

$$\nabla L(\mathbf{x}, \lambda) = \begin{pmatrix} \nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) \\ \nabla_{\lambda} L(\mathbf{x}, \lambda) \end{pmatrix} = \begin{pmatrix} f(\mathbf{x}) + \lambda \nabla g(\mathbf{x}) \\ g(\mathbf{x}) \end{pmatrix}.$$

Substituting it into Eqn. (5), we obtain a system of equations

$$\begin{cases} \nabla f(\mathbf{x}) + \lambda \nabla g(\mathbf{x}) = 0 \\ g(\mathbf{x}) \equiv \mathbf{n}(\mathbf{x})^T \left(\dfrac{\partial e(\mathbf{y}, \mathbf{n}(\mathbf{x}))}{\partial \mathbf{y}} \Big|_{\mathbf{y}=\mathbf{x}} \right) = 0 \end{cases} . \tag{6}$$

From Eqn. (6), we observe that, according to the value of λ, the critical points of f_M can be classified into two groups:

1) if $\lambda = 0$, then any $\mathbf{x} \in M$ satisfies $\nabla f(\mathbf{x}) = 0$ is a critical point of f_M, which is also the critical point of the Morse function f;

2) if $\lambda \neq 0$, then any $\mathbf{x} \in M$ satisfies

$$\varphi(\mathbf{x}) \equiv 1 - \left(\frac{\nabla f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|} \cdot \frac{\nabla g(\mathbf{x})}{\|\nabla g(\mathbf{x})\|} \right)^2 = 0 , \tag{7}$$

is a critical point of f_M, which can be comprehended as that the gradient of the Morse function f is parallel to the normal of the MLS surface.

In this paper, we term the critical points satisfying the first or second condition respectively as the first or second kind of critical points. Many previous researchers have addressed the problem of identifying the first kind of critical points [12]. However, none has explored the issue of identifying the second kind of critical points. In the following, we will give a two-step procedure to identify the second kind of critical points:

1. Critical point candidate generation: Project each point $\mathbf{q}_i \in R^3$ of the input point set $\mathbf{Q}$ onto the MLS surface and obtain a point $\mathbf{q}_i'$. If $\varphi(\mathbf{q}_i') < \varepsilon$, where ε is a given small positive value. Then classify $\mathbf{q}_i'$ as a candidate point and $\varphi(\mathbf{q}_i')$ is defined by Eqn. (7). Figure 1(c) illustrates the distribution of $\varphi(\mathbf{q}_i')$ for the height Morse function on the MLS surface.

2. Critical point refinement: The candidate points generated from the first step are located on the input MLS surface but, in general, not precisely satisfy Eqn. (6). A refinement must be performed by solving the non-linear system of Eqn. (6).

Note the Jacobian matrix of Eqn. (5) over $\mathbf{x}$ can be analytically obtained by

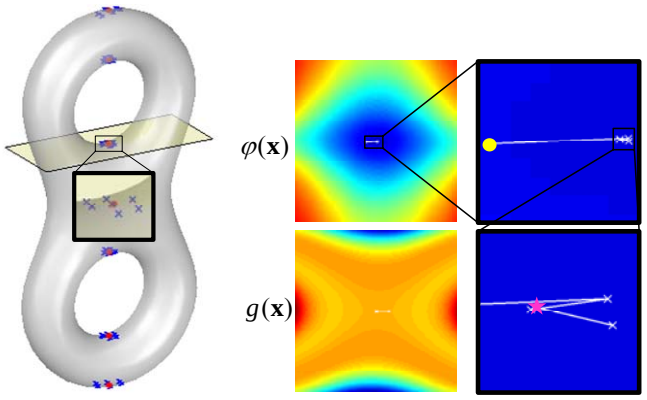
$$\mathbf{J}(\mathbf{x},\lambda)=\begin{pmatrix}\mathbf{H}(f(\mathbf{x}))+\lambda\mathbf{H}(g(\mathbf{x})) & \nabla g(\mathbf{x})\\ \nabla^T g(\mathbf{x}) & 0\end{pmatrix} \tag{8}$$

where $\mathbf{H}(f(\mathbf{x}))$ is the Hessian matrix of $f(\mathbf{x})$, which can be analytically calculated, since we assume $f(\mathbf{x})$ is at least C^2 continuous. And $\nabla g(\mathbf{x})$ (respectively, $\mathbf{H}(g(\mathbf{x}))$) is the gradient vector (respectively, Hessian matrix) of $g(\mathbf{x})$. For further expansion of $\nabla g(\mathbf{x})$ and $\mathbf{H}(g(\mathbf{x}))$, please refer to [27].

Once we have obtained the expression of $\mathbf{J}(\mathbf{x})$, we can solve the non-linear system of Eqn. (6) using the Gauss-Newton optimization method [8], where the refined coordinate values $\mathbf{x}_k$ can be iteratively calculated by

$$\begin{pmatrix}\mathbf{x}_{k+1}\\ \lambda_{k+1}\end{pmatrix}=\left(\mathbf{J}_k^T\mathbf{J}_k\right)^{-1}\mathbf{J}_k^T\left(\mathbf{J}_k\begin{pmatrix}\mathbf{x}_k\\ \lambda_k\end{pmatrix}-\begin{pmatrix}\nabla f(\mathbf{x}_k)+\lambda_k\nabla g(\mathbf{x}_k)\\ g(\mathbf{x}_k)\end{pmatrix}\right)$$

where k is the iteration number. Figure 2 illustrates an example of the critical point refinement process.



(a) (b)

Figure 2. CRITICAL POINT REFINEMENT. (A) GENERATION OF CRITICAL POINTS (RED DOT) BY REFINING THE CANDIDATE POINTS (BLUE CROSS), GEOMETRICAL CONFIGURATION OF THE DOUBLE TORUS SURFACE AND A SLICING PLANE PASSING A CRITICAL POINT. (B) DISTRIBUTION OF $\varphi(\mathbf{x})$ AND $g(\mathbf{x})$ ON THIS SLICING PLANE AND THE REFINEMENT PROCESS FROM A CANDIDATE POINT (YELLOW DOT) TO THE CRITICAL POINT (MAGENTA STAR).

Classification of Critical Points

To determine the types of a critical point $\mathbf{x}$ of the function $f_M = f|_M$, we use the number of negative eigenvalues of the Hessian matrix $\mathbf{H}(f_M(\mathbf{x}))$ [12], [20]. The relationship between the type of critical points and the sign of eigenvalues of the Hessian matrix is given in Table 1. Let the two eigenvalues of $\mathbf{H}(f_M(\mathbf{x}))$ be $\lambda_1 \le \lambda_2$, then the number of negative eigenvalues of $\mathbf{H}(f_M(\mathbf{x}))$ is the index of a critical point. A critical point of index 0, 1 or 2 is called a minimum, saddle, or maximum, respectively.

Table 1. THE RELATIONSHIP BETWEEN TYPE OF CRITICAL POINTS AND SIGN OF EIGENVALUES.

Sign of Eigenvalues ($\lambda_1 \le \lambda_2$)		*Matrix signature*	*Type of critical point*
λ_1	λ_2		
-	-	-2	Maximum (Index-2)
-	+	0	Saddle (Index-1)
+	+	2	Minimum (Index-0)

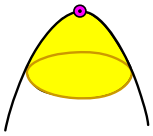
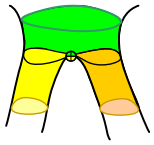
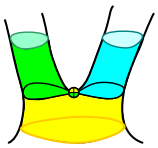
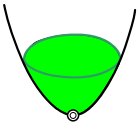
The Hessian matrix $\mathbf{H}(f_M(\mathbf{p}))$ is not easy to calculate directly for a scalar function constrained on another implicit surface. However, the Hessian matrix $\mathbf{H}(f_M(\mathbf{p}))$ has the same signature as $\mathbf{H}_L(\mathbf{p},\lambda)$ [13]; the latter is the bordered Hessian matrix of the Lagrange function L at the critical point $(\mathbf{p},\lambda)$ and it is defined as

$$\mathbf{H}_L(\mathbf{x},\lambda)=\begin{pmatrix}\mathbf{H}(f(\mathbf{x}))+\lambda\mathbf{H}(g(\mathbf{x})) & \nabla g(\mathbf{x})\\ \nabla^T g(\mathbf{x}) & 0\end{pmatrix} \tag{9}$$

where λ is the Lagrangian multiplier, $\nabla g(\mathbf{x})$ and $\mathbf{H}(g(\mathbf{x}))$ are the gradient and Hessian matrix of $g(\mathbf{x})$.

Meanwhile, depending on the sign of λ, the saddle points can be categorized into two subtypes: if $\text{sgn}(\lambda)=1$, then the saddle point is a top saddle; if $\text{sgn}(\lambda)=-1$, then the saddle point is a bottom saddle. The basic relationship between the four types of critical points (maximum, top saddle, bottom saddle and minimum) and the topology change when a sectioning plane moves up through a critical point is given in Table 2.

Table 2. THE RELATIONSHIP BETWEEN CRITICAL POINT TYPE AND TOPOLOGY CHANGE.

Critical point type	*Topology at critical point*	*Topology change*	*Number of contours*	*Graphical illustration*
Maximum	An isolated point	Removal of a closed contour	-1	
Top saddle	A self-intersection of the contour	Merge of two closed contours	-1	
Bottom saddle	A self-intersection of the contour	Split of a closed contour	+1	
Minimum	An isolated point	Creation of a new closed contour	+1	

Construction of Morse-Smale Complex

After the identification and classification of the critical points, these points will be linked to form a Morse-Smale complex. The connection of critical points can start from any saddle point by building two steepest ascending integral lines and two steepest descending integral lines. A steepest ascending integral line is generated by tracing steepest ascending directions until it approaches a maximum critical point; Reversely, a steepest descending integral line is generated by tracing steepest descending directions until it approaches a minimum critical point.

In this section, we apply the Runge-Kutta-Fehlberg method [24] to trace along an integral line $\mathbf{c}(t)\subset M$, which is defined as a maximal path in M whose tangent vectors agree with the gradient of f_M at every point of the path:

$$\mathbf{c}'(t)=\nabla f_M(\mathbf{c}(t)). \tag{10}$$

where

$$\nabla f_M(\mathbf{c}(t))=\frac{\nabla g(\mathbf{c}(t))\times(\nabla f(\mathbf{c}(t))\times\nabla g(\mathbf{c}(t)))}{\|\nabla g(\mathbf{c}(t))\|^2}, \tag{11}$$

where $\nabla f(\mathbf{c}(t))$ and $\nabla g(\mathbf{c}(t))$ are respectively the gradient of Morse function f and implicit MLS surface function g. The above initial value problem can be solved by the forth/fifth order Runge-Kutta-Fehlberg method with an adaptive step size, which helps to balance the computational efficiency and accuracy.

The initial position of this algorithm is given by perturbing the source saddle point along the direction determined by the eigenvectors of the Hessian matrix of f_M, which is defined as

$$H(f_M(u,v,w(u,v)))=-\frac{f_w}{g_w}\begin{bmatrix} g_{uu} & g_{uv} \\ g_{uv} & g_{vv}\end{bmatrix},$$

where $(u,\ v,\ w)^T$ is a coordinate under a local coordinate system defined by the tangent plane T of the MLS surface at point $\mathbf{p}=\mathbf{c}(t)$, f_w, g_w, g_{uu}, g_{vv}, and g_{uv} are local spatial derivatives. These spatial derivatives can be obtained analytically through a coordinate transformation of the derivatives $\nabla g(\mathbf{x})$ and $\mathbf{H}(g(\mathbf{x}))$ [27].

Then the Runge-Kutta-Fehlberg method iteratively calculates the next point on the integral line with an adaptive step size until it reaches a maximum/minimum critical point within a certain threshold.

Generation of Enhanced Reeb Graph

A standard Reeb graph is a graphical representation of the connectivity of an object between critical points [25]. More specifically, for a Morse function f_M, the Reeb graph $G(V,E)$ is defined as a graph whose nodes V are the critical points of f_M and whose arcs E are connected components of the level sets of f_M. However, this standard Reeb graph contains only topological information and all geometrical relationships are lost. In this paper, we extend the standard Reeb graph by assign additional geometrical information to the nodes V and the arcs E: for a node $v_i\in V$, we replace it with $\mathbf{p}_i$, i.e., the 3D coordinate of the corresponding critical point; for an arc $(v_i,v_j)\in E$, we replace it with a polyline, whose endpoints are $\mathbf{p}_i$ and $\mathbf{p}_j$ and middle points are generated by integrating Eqn. (10) with an initial step from $\mathbf{p}_i$. This attached geometrical information will later be used to help generate slicing contours robustly and efficiently. A comparison of the standard and the enhanced Reeb graphs are shown in Figure 1(e), which illustrates that the enhanced Reeb graph is a Reeb graph associated with its projection onto the MLS surface.

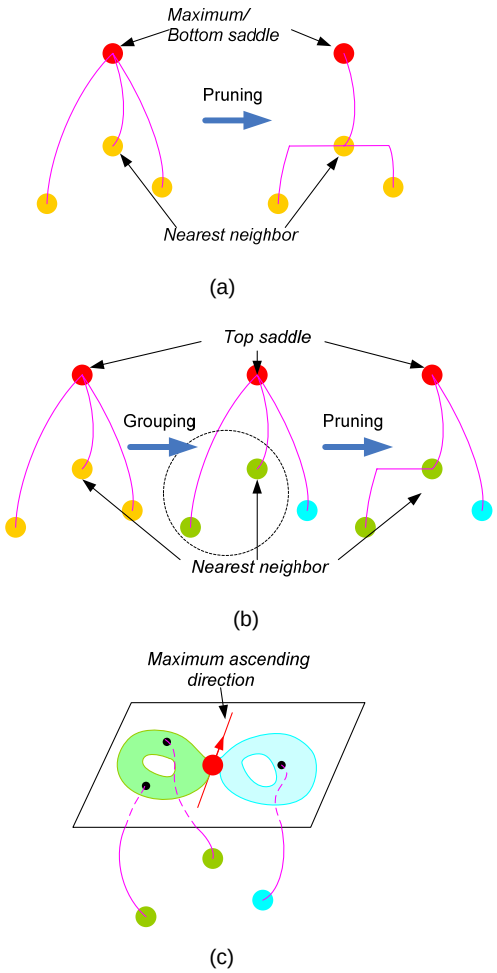


Figure 3. RULES OF GENERATING AN ENHANCED REEB GRAPH FROM THE MORSE-SMALE COMPLEX. (A) RULE #1 FOR MAXIMUM AND BOTTOM SADDLE. (B) RULE #2 FOR TOP SADDLE. (C) GROUPING THE LOWER NEIGHBOR POINTS IN RULE #2.

As an inverse of the traditional Reeb graph generation method where the Reeb graph is generated from model slicing, we create an enhanced Reeb graph from the Morse-Smale complex by pruning the redundant arcs and then use it to guide the slicing process. Before we address the key challenges in the pruning operation for enhanced Reeb graph generation, let us first re-examine Table 2, where we find that the number and the types of arcs for each node of the Reeb graph are determined by the types of the critical point:

- A maximum point makes a closed contour collapse. Therefore, in the Reeb graph, there is only one downward arc that links this maximum point with another lower critical point.
- A top saddle point makes two closed contours merge into one. Therefore, there are two downward arcs and one upward arc that link this top saddle point with another higher critical point.
- A bottom saddle point makes a closed contour split into two contours. Therefore, there are one downward arc and two upward arcs.
- A minimum creates a new closed contour. Therefore, there is only one upward arc.

Based on the above four principles, we have the following strategy in generating an enhanced Reeb graph from the Morse-Smale complex. We first sort all the critical points according to their heights. Then we process them from the highest point to the lowest points according to the following three rules: 1) *Maximum or bottom saddle*: if it has more than one downward arcs, we keep the arc between this critical point and its nearest, lower neighbors (we term it adjacent point), which has the closest, lower height value to the critical point. And prune the other arcs by trimming them at the height of the adjacent point and re-linking them to it, as shown in Figure 3 (a); 2) *Top saddle*: we first classify all its lower neighbors into two groups. This is achieved with the aim of the intersection regions of this top saddle, which is defined as a set of points $A = \{\mathbf{q} \mid -\varepsilon_c < g(\mathbf{q}) < \varepsilon_c \cap \mathbf{q} \in S_{top}\}$, where ε_c is a small positive value and S_{top} is the slicing plane that contains the top saddle. This region (set A) is further divided into two sub-regions, i.e.,

$$\begin{cases} A_l = \{\mathbf{q} \mid (\mathbf{q} - \mathbf{p}_{top}) \cdot (\mathbf{v}_{\max} - \mathbf{n}) > 0 \cap \mathbf{q} \in A\} \\ A_r = \{\mathbf{q} \mid (\mathbf{q} - \mathbf{p}_{top}) \cdot (\mathbf{v}_{\max} - \mathbf{n}) \le 0 \cap \mathbf{q} \in A\} \end{cases},$$

where $\mathbf{p}_{top}$ represents the top saddle point, $\mathbf{v}_{\max}$ is the maximum ascending direction of $\mathbf{p}_{top}$ and $\mathbf{n}$ is the normal of the slicing plane S_{top}. Then these two sub-regions are used to classify the lower neighbors: if an arc intersects A_l (resp., A_r), we insert the corresponding neighboring point into Group One (resp., Group Two), as shown in Figure 3(c). After we finish the grouping process, we apply the same pruning procedure as described in rule #1 for each of these two groups, as shown in Figure 3 (b); 3) *Minimum*: we just remove all downward arcs.

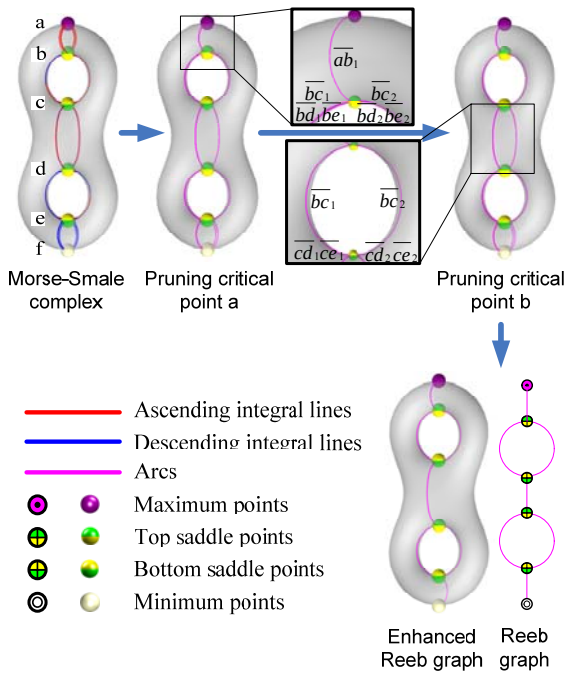


Figure 4. ILLUSTRATION OF THE GENERATION OF AN ENHANCED REEB GRAPH FROM THE MORSE-SMALE COMPLEX WITH THE EXAMPLE OF DOUBLE TORUS.

Figure 4 illustrates the generation of an enhanced Reeb graph from the Morse-Smale complex with a double torus. From Figure 4, we can see that before pruning critical point ***a***, it has eight arcs linked with the four saddle points ***b***, ***c***, ***d*** and ***e***. Since ***a*** is a maximum point, we apply the rule #1 for pruning. And in the resulting graph, point ***a*** has only one arc $\overline{ab}_1$ and the other arcs are so trimmed that they are now only connected to critical point ***b***. That is, point ***b*** has six arcs, $\overline{bc}_1$, $\overline{bc}_2$, $\overline{bd}_1$, $\overline{bd}_2$, $\overline{be}_1$ and $\overline{be}_2$. For the top saddle point ***b***, we apply the rule #2 for pruning: we first group the six arcs into two sets: $\{\overline{bc}_1, \overline{bd}_1, \overline{be}_1\}$ and $\{\overline{bc}_2, \overline{bd}_2, \overline{be}_2\}$. For each of these two groups, we perform the pruning operation. Then we get only two arcs for point ***b***, i.e., $\overline{bc}_1$, $\overline{bc}_2$ and point ***c*** will have four arcs $\overline{cd}_1$, $\overline{cd}_2$, $\overline{ce}_1$ and $\overline{ce}_2$. Keeping processing all the critical points from ***a*** to ***f*** similarly, we will finally obtain the enhanced Reeb graph as shown in Figure 4.

TOPOLOGICALLY ENHANCED SLICING ALGORITHM

Intersection contours are generated by marching. Therefore, when the intersecting plane contains critical points, we should make the marching algorithm topology-aware so as to avoid marching failure and obtain the correct contour.

Our original marching algorithm discussed in [28], [29] cannot handle this robustly. It requires a marching normal direction n_P determined by:

$$n_P = n_H \times (n_S \times n_H).$$

where n_H is the normal of the slicing plane and n_S is the normal of MLS surface S. Rotating n_P by a right angle on the slicing plane gives the step length direction t_P at each slice. It is actually the tangent vector at the surface point and it guides marching step to the next contour point. At critical points location (E, F in Figure 5), both n_P and t_P vanish as shown in Figure 5(c) and marching suddenly loses the guiding direction, causing marching to fail.

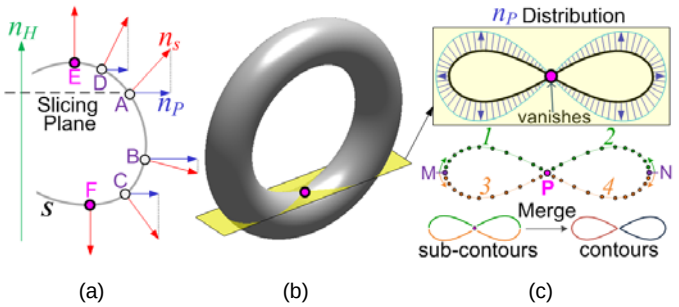


Figure 5: MARCHING BASED SLICING (A) MARCHING NORMAL DIRECTION n_P AND SURFACE NORMAL n_S. (B) SLICING AT HEIGHT OF TORUS' SADDLE (C) MARCHING NORMAL DISTRIBUTION AND CONTOUR MERGING AT SADDLE

The new topologically enhanced algorithm works as follows. First, seed points (M, N in Figure 5(c)) far away from saddle P are chosen as marching starting points. Merging sub-contours generated by marching clockwise and counterclockwise from each seed point then yields each contour. This modified marching will naturally stop when it reaches sufficiently close to a non-degenerate critical point on the slicing plane.

As for degenerate region, however, sub-contours will not meet to form each contour loop because degenerate regions are essentially composed of spatially adjacent critical points that are not isolated. The contour gap between the sub-contour ending points should be filled based on the boundary of the degenerate region; this will be discussed in the later section.

In this paper, our new slicing algorithm extends the original one in [28],[29] via the following topological enhancements.

- The seed points for contour marching are determined by intersecting all the integral lines of the extended Reeb graph with the slicing plane.
- The topology of the MLS surface is guaranteed to be preserved in the sliced contours under adequate sampling, that is, the MLS-represented feature size is greater than the maximum point spacing of point-cloud data.
- The number of contours can also be derived from intersecting enhanced Reeb graph with the slicing plane. Here we give a formula that not only checks the validity of the generated Reeb graph but also computes the number of contours at a specific height. We define contour topology characteristic as:

$$\varsigma(z) = M + S_B - S_T - m,$$

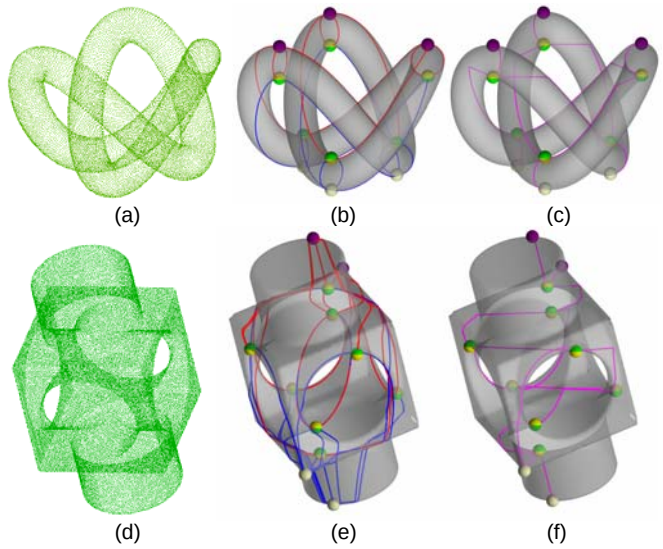
where M , S_B, S_T and m is the number of maximum, bottom saddle, top saddle and minimum entities that lie below the current height value z . This formula comes from the topology changes listed in Table 2.

IMPLEMENTATION AND EXAMPLES

We have implemented the above algorithms, including critical points and integral line extractions, Morse-Smale complex and enhanced Reeb graph construction, and topologically enhanced slicing algorithm with Matlab. Examples are presented below to illustrate and validate the proposed approach.

Enhanced Reeb Graph Generation

In this section, we present three examples performed with the enhanced Reeb graph generation algorithm described previously. In the first example, the input data is sampled from a knot surface, which contains 90,000 points, as shown in Figure 6 (a). In Figure 6 (b), nine critical points are identified and classified: three magenta balls represent the maximum critical points, six green & yellow balls represent the (top and bottom) saddle points and three white balls represent the minimum critical points. Meanwhile, the ascending integral lines (red curves in Figure 6 (b)) and descending integral lines (blue curves in Figure 6 (b)) are generated and used to setup the Morse-Smale complex. Figure 6 (c) illustrates the resulting enhanced Reeb graph, where the red and blue integral lines are pruned to form magenta arcs. The second example (the second row of Figure 6) is sampled from a mechanical part with 66,477 points, which demonstrates the ability of our approach in handling models with sharp corners. The third example is a fertility sculpture from AIM@SHAPE, which contains 98,602 points, as shown in Figure 6 (g). The results shown in Figure 6 (h) and (i) demonstrates the robustness of our algorithms in handling noisy, real data.



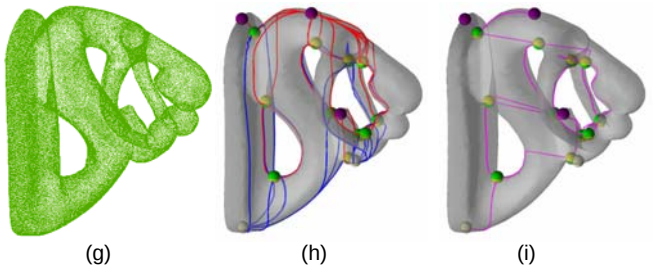


Figure 6. ENHANCED REEB GRAPH GENERATION FOR THREE EXAMPLES: A KNOT SURFACE IN THE FIRST ROW, A MECHANICAL PART IN THE SECOND ROW AND A REAL SCULPTURE DATA IN THE THIRD ROW. IN EACH OF THESE THREE ROWS, THE FIRST COLUMN SHOWS THE CORRESPONDING INPUT POINT DATA, THE SECOND COLUMN SHOWS THE RESULTING MORSE-SMALE COMPLEX AND THE THIRD COLUMN SHOWS THE RESULTING ENHANCED REEB GRAPH.

Topologically Enhanced Slicing

With the above generated enhanced Reeb graphs, we can further extract slicing data of these three models by applying the topologically enhanced slicing algorithm. In comparing with the original algorithm, this algorithm is more robust and capable of handling various models, as shown in Figure 7.

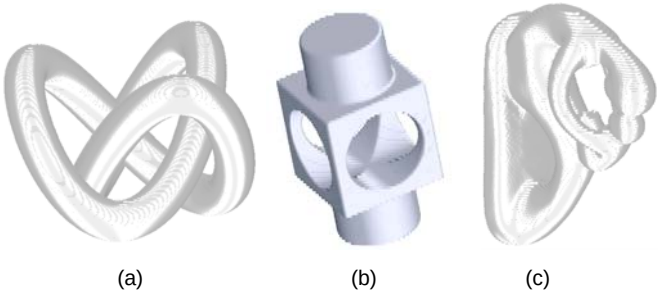


Figure 7. TOPOLOGY ENHANCED SLICING FOR THREE EXAMPLES. (A) THE KNOT SURFACE. (B) THE MECHANICAL PART WITH SHARP CORNERS. (C) THE REAL SCULPTURE DATA WITH NOISE AND HOLES.

For a detailed illustration of the topological robustness for our slicing algorithm, we use a knot surface model as an example and examine two specific slices, i.e., $z = 0.0$ and $z = 1.83$, as shown in Figure 8(a). For better illustration of the sliced data, we adopt the top views of these two slices, as shown in Figure 8(b) and (c), where the red dot points with grey lines represent the output 2D contours. These two slices are generated by two different plane/MLS surface intersection algorithms: 1) the slice at $z = 0.0$, as shown in Figure 8(b), contains no critical points and thus the original marching based algorithm [28], [29] is applied to get the contours. Note that the six blue points in Figure 8(b) represent the initial points used to find the starting points for the marching process, which are captured by intersecting the enhanced Reeb graph with the slicing plane $z = 0.0$. In comparison with the method introduced in [28], [29], the topologically enhanced method has the following advantages: no prescribed value ε is necessary, the number of initial points drops dramatically and all the contours are guaranteed to be found. 2) the slice at

$z = 1.83$, as shown in Figure 8(c), contains one saddle point (the green & yellow ball), where two contours are merging with each other. Thus the initial points for these two merging contours are chosen to be away from the saddle, while the initial points for the other three contours are determined by intersecting the enhanced Reeb graph with the slicing plane $z = 1.83$, as shown in Figure 8(c). The finial sliced model is illustrated in Figure 7(a)

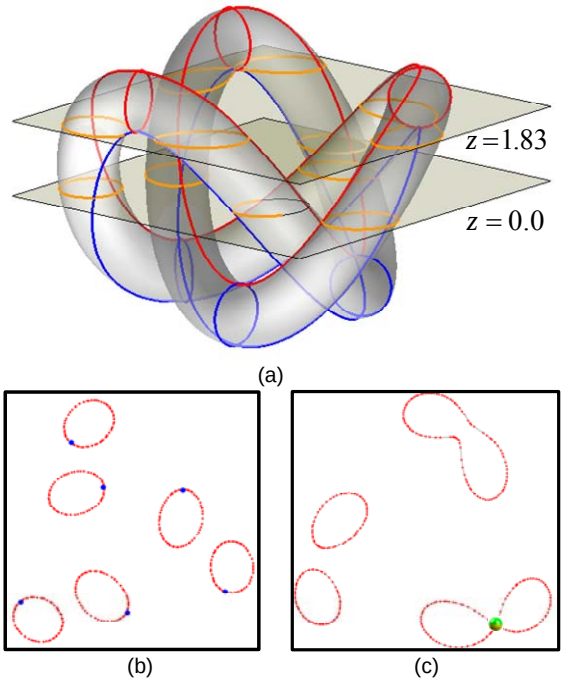


Figure 8. TOPOLOGY ENHANCED SLICING FOR THE KNOT SURFACE. (A) ISO-VIEW OF THE KNOT DATA WITH INTEGRAL LINES. (B) TOP VIEW OF THE SLICE AT $z = 0.0$. (C) TOP VIEW OF THE SLICE AT $z = 1.83$.

Noisy and Unevenly Distributed Point Sets

To validate the robustness of our method in handling noisy and unevenly distributed point sets, we add varying levels of synthetic noise onto the nominal knot surface, as shown in Figure 9. For a better illustration of the varying noise, we pick a typical normal section of the nominal knot surface and the corresponding noisy point data, as shown in Figure 9(c). From the bottom to the top of this circular section, the standard deviation of the noise changes from 0 to 0.1.

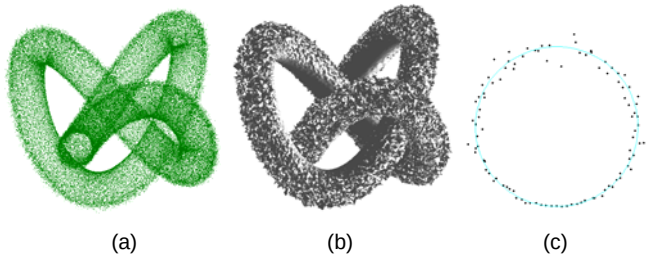


Figure 9. Synthetic noise data sampled from the knot surface. (a) point cloud. (b) rendered triangluar mesh model. (c) a typtical circular section with a varying levels of noises (standard deviation from 0 to 0.1).

We examined this noisy and unevenly distributed data at a specific slice, i.e., $z = 1.83$, as shown in Figure 10(a), where the red lines represent the output contours, the blue line represents the nominal contour and the green cross points represent the input data points with intensity fading away when they are farther away from the slicing plane. For better illustrating the quality of the generated slice, we showed a zoom-in of the left-most contour as well as the corresponding nominal contour in Figure 10(b) and plot the distance (error) between these two contours as a quantitative description of the data quality in Figure 10(c). In Figure 11(c), we find that the maximum error is smaller than the given standard deviation of the noise, i.e., 0.1, which reveals the de-noising ability of our algorithm. Meanwhile, the uneven distribution of the noise is properly handled. For the sake of simplicity, in this paper, we keep the Gaussian factor *h* constant, even though adaptive choice of *h*, e.g. through the scheme in [23] would likely lead to more accurate surface estimates.

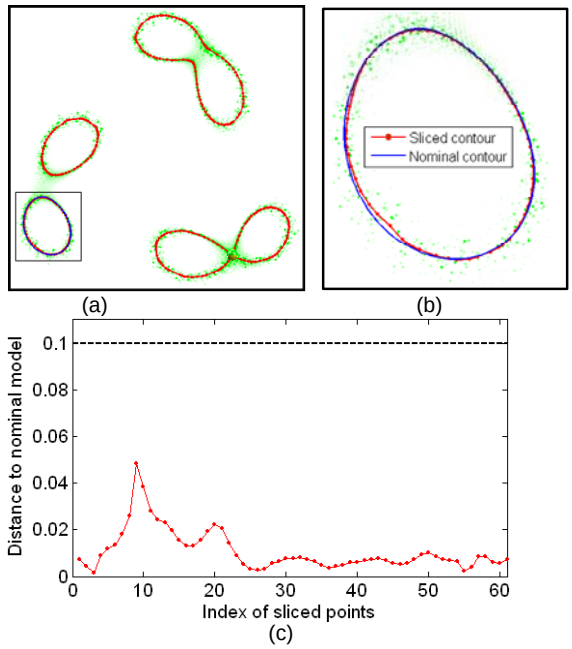


Figure 10. TOPOLOGY ENHANCED SLICING FOR THE NOISY UNEVENLY DISTRIBUTED DATA. (A) TOP VIEW OF THE SLICE AT $z = 1.83$. (B) ZOOM-IN OF (A). (C) DISTANCE BETWEEN THE SLICED POINTS IN (B) AND THE NOMINAL MODEL.

SLICING UNDER DEGENERACY

The aforementioned algorithms have been demonstrated to be able to slice various point-set surfaces with non-degenerate critical points. We now show that the algorithms for Reeb graph construction and slicing can be extended to degenerate cases.

Identify Degenerate Region

Degenerate regions can be identified by clustering spatially adjacent critical points; they can be classified into *degenerate maximum, minimum* and *saddle* if we treat the region like a point. Related concepts such as rings, wedges in [7] could also be easily extended from non-degenerate critical

points to degenerate regions. Figure 11 shows three types of degenerate regions along with their wedges of a degenerate torus.

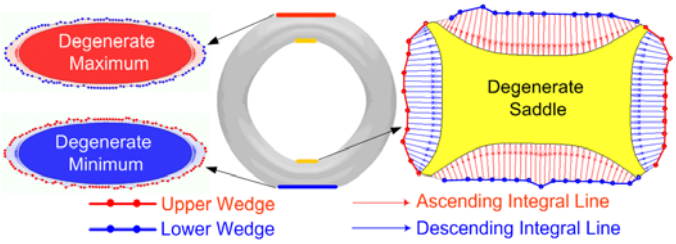
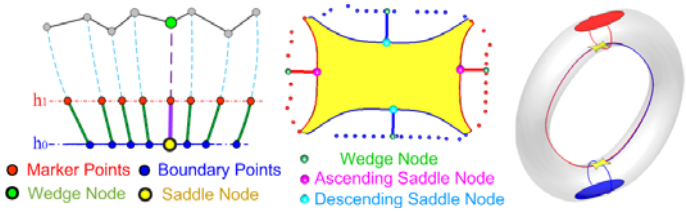


Figure 11: DEGENERATE TORUS AND ITS DEGENERATE MAXIMUM, MINIMUM and SADDLE

If every degenerate region (degenerate type, region boundary) is well-defined and ascending and descending integral lines from degenerate saddles are known, then Morse-Smale complex and extended Reeb Graph could be readily constructed for slicing.

If this information of the degenerate region is not available, we provide a heuristic method as follows. Comparing the height values of regional wedges and the region itself reveals the degenerate type. Tracing integral lines from its wedge back to the height of region could outline the region boundary. Ascending and descending integral lines emanating from degenerate regions could be determined by locating *saddle node* and *wedge node* on region boundary as shown in Figure 12. Integral line that traces from saddle node to wedge node travels the shortest distance between region height h_0 and a prescribed height h_1 very close to h_0; this ensures the obtained integral lines follow the steepest ascending or descending directions.



(a) (b) (c)

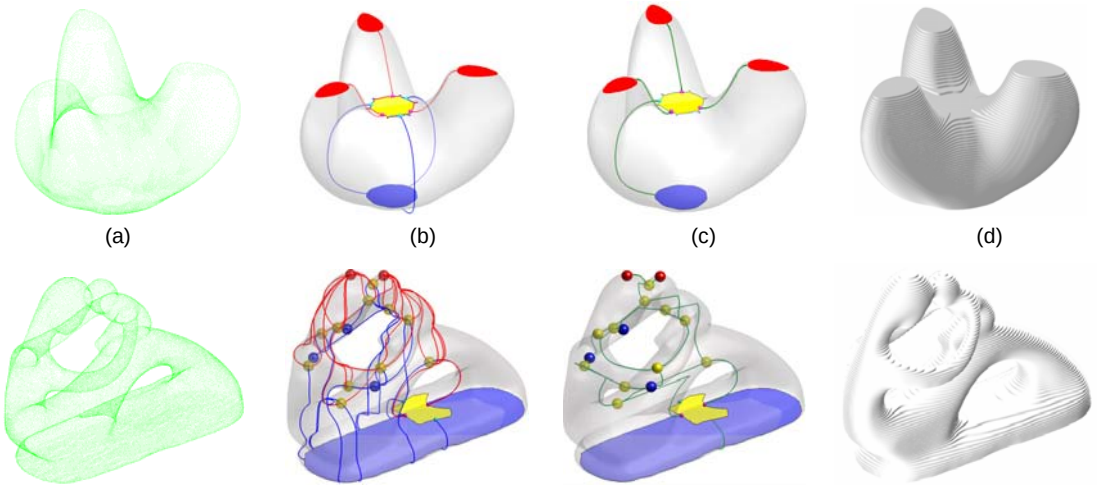
Figure 12: (a) SADDLE NODE DETERMINATION. (b) SADDLE NODES AND WEDGE NODES. (c) GENERATED COMPLEX.

When Slicing Plane contains Degenerate Region

The slicing algorithm is modified when slicing plane contains degenerate regions. If a *degenerate maximum* is being sliced, we do nothing and move on to next layer. If a *degenerate minimum* is being sliced, we use its boundary as contour points for the slices. If a *degenerate saddle* is being sliced, part of its region boundary associated with the upper or lower wedge will be used along with sub-contours generated by marching to form a closed contour loop.

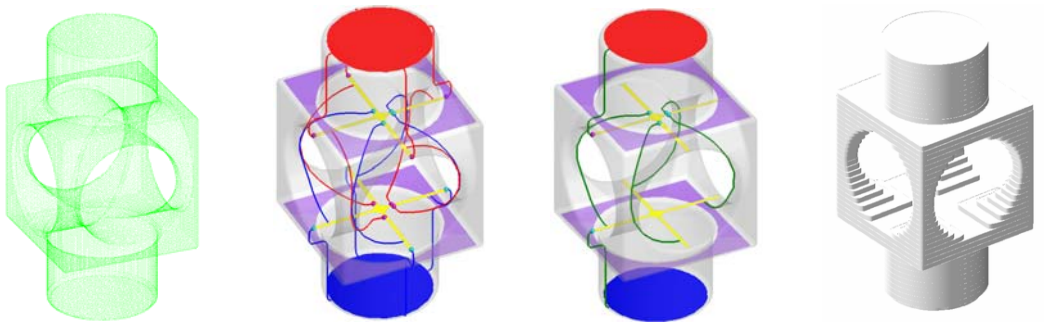
Slicing Results of Degenerate Models

Figure 13(a) is a point-set (48.8K points) that contains a *degenerate monkey saddle* (multiplicity=2) in the middle. Its three ascending and three descending integral lines lead to the three degenerate maxima and one degenerate minimum respectively. Fertility Sculpture (58.2K points) in Figure 13(e) contains a degenerate saddle and a degenerate minimum at the base. As for many mechanical parts, degenerate regions may take the form of lines, which can also be treated like region. Using this technique, the Block part (62.8K points) contains two cross shape *degenerate dog saddles* (multiplicity=3), a maximum and a minimum. These examples demonstrate that our topologically enhanced slicing algorithm is applicable to degenerate model, even for those with high multiplicity.



(a) (b) (c) (d)

(e) (f) (g) (h)



(i) (j) (k) (l)

Figure 13: POINT CLOUD, MORSE COMPLEX AND REEB GRAPH AND SLIDED MODEL OF MONKEY SADDLE POINT-SET (48.8K POINTS), FERTILITY (58.2K POINTS) AND MECHANICAL BLOCK (62.8K POINTS).

CONCLUSION

In this paper, we have presented an approach for slicing point-set surfaces with topological guarantee. This is done through the analysis of a height Morse function restricted to an implicitly defined MLS surface. The critical points are obtained from the Lagrangian multiplier method. Morse-Smale complexes are constructed, from which the Reeb graph is built. The enhanced Reeb graph provides the seed point for contour marching in slicing and the information at topological transition locations. The slicing method has also been extended to degenerate cases.

The Lagrangian multiplier method and the associated bordered Hessian have proved to be useful in identifying critical points of Morse functions restricted to the point-set surface. Other Morse functions such as blobby [12] or curvature functions can also be used. Even though the constraint surface used in this paper is the MLS surface, other forms of implicit surfaces can also be used.

The resulting Morse-Smale complex, enhanced Reeb graph, and the sliced model can be used in applications such as computer-aided design, rapid prototyping and shape understanding.

ACKNOWLEDGMENT

The authors are grateful for financial support from the National Science Foundation.

REFERENCES

[1] Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., and Silva, C. T., 2003, "Computing and rendering point set surfaces," *IEEE Transactions on Visualization and Computer Graphics*, **9**(1), Jan-March, pp. 3-15.

[2] Attene, M., Biasotti, S., Spagnuolo, M., 2003, "Shape understanding by contour-driven retiling," *The Visual Computer*, **19**(2-3), pp. 127-138.

[3] Alexa, M., Behr, J., Cohen-Or, D., Fleishman S., Levin D.,2001, "Point set surfaces," *Proceedings of the Conference on Visualization*, pp. 21-28.

[4] Acar, E. U., Choset, H., Rizzi, A. A., Atkar, P., Hull, D., 2002, "Morse decompositions for coverage tasks," *International Journal of Robotics Research*, **21**(4), pp. 331-344.

[5] Amenta, N., Kil, Y. J., 2004, "Defining point-set surfaces," *ACM Transactions on Graphics*, **23**(3), pp. 264-270.

[6] Amenta, N., Kil, Y. J., 2004, "The domain of a point set surface," *Proceedings of 2004 IEEE/Eurographics Symposium on Point-based Graphics*, pp. 139-147.

[7] Banchoff, T. F., 1970, "Critical points and curvature for embedded polyhedral surfaces," *Am. Math. Monthly 77*, pp. 475-485.

[8] Coleman, T. F., Li, Y., 1993, "An interior trust region approach for nonlinear minimization subject to bounds," *Technical Report*, Ithaca, NY, USA.

[9] Dey, T. K., Sun, J., 2005, "Adaptive MLS surfaces for reconstruction with guarantees," *Proceedings of Eurographics Symposium on Geometry Processing*, pp. 43-52.

[10] Dey, T. K., Goswami, S., Sun, J., 2005, "Extremal surface based projections converge and reconstruct with isotopy," *Technical Report OSU-CISRC-4-05-TR25*.

[11] Edelsbrunner, H., Harer, J., Zomorodian, A., 2003, "Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds," *Discrete and Computational Geometry*, **30**(1), pp. 87-107.

[12] Hart, J. C., 1998, "Morse theory for implicit surface modeling," *Mathematical Visualization, Hege H-C, Polthier K. ed., Springer-Verlag*, pp. 257-268.

[13] Hassell, C., Rees, E.,T., 1993, "he index of a constrained critical point," *The American Mathematical Monthly*, **100**(8), pp. 772-778.

[14] Helman, J. and Hesselink, L., 1989, "Representation and Display of Vector Field Topology in Fluid Flow Data Sets," *IEEE Computer*, **22**(8):27-36.

[15] Helman, J. and Hesselink, L., 1990, "Surface representations of two- and three-dimensional fluid flow topology," *Proceedings of the First IEEE Conference on Visualization '90*, pp. 6-13.

[16] Kim, S. K., Kim, C. H., 2005, "Finding ridges and valleys in a discrete surface using a modified MLS approximation," *Computer-Aided Design*, **37**(14), pp 1533-1542.

[17] Kreveld, M., Oostrum, R., Bajaj, C., Pascucci, V., Schikore, D., 1997, "Contour trees and small seed sets for isosurface traversal," *Proceedings of the thirteenth annual symposium on Computational geometry*, pp.212-220, Nice, France.

[18] Levin, D., 1998, "The approximation power of Moving Least-Squares". *Mathematics of Computation*, vol. **67**(224), pp. 1517-1531.

[19] Levin, D., 2003, "Mesh-independent surface interpolation," *Geometric Modeling for Scientific Visualization, Springer-Verlag*, pp. 37-49.

[20] Milnor, J., 1963, *Morse theory*, Princeton University Press.

[21] Ohtake, Y., Belyaev, A.G., and Seidel, H.-P., 2004, "Ridge-valley lines on meshes via implicit surface fitting," *ACM Trans. Graph.*, **23**(3), pp. 609-612.

[22] Pauly, M., Keiser, R., Kobbelt, L. P., and Gross, M., 2003, "Shape modeling with point-sampled geometry," *ACM Trans. Graph.*, **22**(3), 641-650.

[23] Pauly, M., 2003, "Point Primitives for Interactive Modeling and Processing of 3d Geometry," Ph.D. thesis, ETH Zurich.

[24] Press, W., Flannery, B., Teukolsky, S., Vetterling, W., 1988, *Numerical recipes in C: the art of scientific computing*, Cambridge University Press.

[25] Wood, Z., Hoppe, H., Desbrun, M., Schroder, P., 2004, "Removing excess topology from isosurfaces," *ACM Trans. Graph.*, **23**(2), pp. 190-208.

[26] Wood, Z., Desbrun, M., Schroder, P., Breen, D., 2000, "Semi-regular mesh extraction from volumes," *Proceedings of IEEE Visualization '00*, Los Alamitos California, pp. 275-282.

[27] Yang, P., Qian, X., 2007, "Direct computing of surface curvatures for point-set surfaces," *Proceedings of 2007 IEEE/Eurographics Symposium on Point-based Graphics*, Prague, Czech Republic.

[28] Yang, P., Qian, X., 2008, "Adaptive slicing of Moving Least Squares surfaces: toward direct manufacturing from point cloud data," *ASME Transactions Journal of Computing and Information Science in Engineering*, **8**(3), pp. 433-442.

[29] Yang, P., Qian, X., 2009, "Direct Boolean intersection between acquired and designed geometry," *Computer-Aided Design*, **41**(2), pp. 81-94.

[30] Zhang, D., Yang, P., and Qian, X., 2009, "Adaptive NC Path Generation from Massive Point Data with Bounded Error," *ASME Transactions Journal of Manufacturing Science and Engineering*, **131**(1), pp. 011001-113.