Salient Critical Points for Meshes

Yu-Shen Liu* Min Liu† Daisuke Kihara† Karthik Ramani†
Purdue University, West Lafayette, Indiana, USA

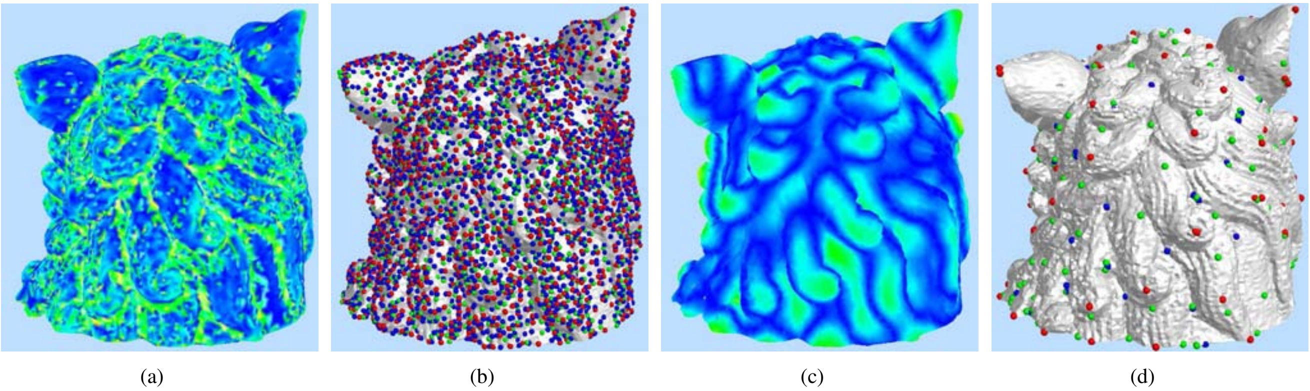

Figure 1: Salient critical points (the blue, red, and green points are minimum, maximum, and saddles, respectively). (a) The back of the lion head model with large noise and small hair textures, and the corresponding mean curvature visualization. (b) The mean curvature function yields 7,629 critical points due to the curvature function's sensitivity to noise. (c) The corresponding mesh saliency. (d) Salient critical points with lower number. Since mesh saliency in (c) captures the hair texture and negates the noisy curvature in (a), our method based on saliency selects the more interesting critical points in the important region. In color images shown in this paper, warmer colors (reds and yellows) show high curvature or saliency and cooler colors (blues) show low curvature or saliency.

Abstract

A novel method for extracting the salient critical points of meshes, possibly with noise, is presented by combining mesh saliency with Morse theory. In this paper, we use the idea of mesh saliency as a measure of regional importance for meshes. The proposed method defines the salient critical points in a scalar function space using a center-surround filter operator on Gaussian-weighted average of the scalar of vertices. Compared to using a purely geometric measure of shape, such as curvature, our method yields more satisfactory results with the lower number of critical points. We demonstrate the effectiveness of this approach by comparing our results with the results of the conventional approaches in a number of examples. Furthermore, this work has a variety of potential applications. We give a direct application to the hierarchical topological representation for meshes by combining the salient critical points with the Morse-Smale complex.

Keywords: critical points, saliency, Morse theory

1 Introduction

Morse theory is a powerful mathematical tool for determining the topology of a manifold from the critical points of one suitable scalar function on the manifold. Recently, discrete Morse theory [Banchoff 1970; Edelsbrunner et al. 2003] on a triangulated manifold has also became an active research area in computer graphics and computational geometry using different real scalar functions [Edelsbrunner et al. 2003; Bremer et al. 2004; Dong et al. 2006; Natarajan et al. 2006; Ni et al. 2004]. In discrete Morse theory, extracting the critical points of 3D meshes is an important problem. However, a poor choice of this real function can lead to a complex configuration of a high number of critical points due to noise. In addition, some salient critical points on important regions might also be missed when some methods of smoothing (or fairing) Morse function are used. Recently, mesh saliency [Lee et al. 2005], as a measure of regional importance for meshes, has been derived from 2D image techniques. In this paper, we focus on the problem of extracting the salient critical points of meshes by combining mesh saliency with Morse theory.

Morse theory, which was originally devised for smooth functions on manifolds, connects the differential geometry of a surface with its algebraic topology. Morse theory has been extended to piecewise linear functions on triangulated meshes [Banchoff 1970]. Given a real-valued function over some shape, discrete Morse theory describes the connectedness of the shape from the configuration of the points where the function's gradient vanishes, its so-called *critical points* (e.g. minima, maxima, saddles). Extracting the critical points of 3D meshes is an important problem in discrete Morse theory. Some applications are strongly dependent on the quality of the critical points, such as quadrilateral remeshing [Dong et al. 2006] and surface segmentation [Natarajan et al. 2006]. However, there are still two problems in extracting critical points from meshes due to noise. One problem is that a poor choice of Morse function can yield many more critical points due to the scalar function's sensitivity to surface noise [Ni et al. 2004]. Figure 1(b) shows an example for the lion head model with large noise, in which a mean curvature function yields 7,629 critical points. These extra critical points are caused by the poor curvature function's sensitivity to noise and hair textures on the back of this model. The other problem is that some

*e-mail: liuyushen00@gmail.com
†e-mail: {liu66, dkihara, ramani}@purdue.edu

SPM 2007, Beijing, China, June 04 – 06, 2007.
© 2007 ACM 978-1-59593-666-0/07/0006 $5.00

salient critical points in important regions might be missed when some smoothing methods are used. For example, when the isotropically smoothing Morse function is used for this goal, some salient critical points might be diffused and skipped.

There are several methods for resolving the above two problems, including directly smoothing the surface, simplifying the topology by cancelling pairs of critical points [Bremer et al. 2004; Edelsbrunner et al. 2003], and smoothing the Morse function using the Laplacian operation in the scalar function space [Dong et al. 2006; Ni et al. 2004]. Directly smoothing the original surface will change the surface and destroy the original surface position. Topology simplifying needs to build the persistence [Edelsbrunner et al. 2002] and hierarchy [Bremer et al. 2004; Edelsbrunner et al. 2003], and requires the expenditure of large amounts of time and space if the number of vertices on the meshes is large. The method of smoothing the Morse function leapfrogs this persistence organization and removes unwanted critical points in a single step. Ni et al. [2004] use the Laplacian operator for smoothing the Morse function to cancel many unnecessary critical points. Dong et al. [2006] apply their work to surface remeshing. Laplacian smoothing can get a lower number of critical points, but it is hard to extract the salient critical points in some interesting and important regions. Laplacian smoothing is isotropic, and therefore it also diffuses shape features in the scalar function space and might skip some salient critical points when smoothing the Morse function. Furthermore, Ni et al.'s method needs to solve eigenvectors for a linear system of Laplace equation to find a smooth Morse function. Solving a number of eigenfunctions requires the expenditure of large amounts of time, if the number of vertices for a mesh is large [Ni et al. 2004]. In this paper, we follow some ideas from Ni et al.'s works [Ni et al. 2004] using mesh saliency instead of Laplacian smoothing. In contrast, our method uses an iterative strategy instead of solving eigenvectors for a linear system of Laplace equation.

The purpose of the saliency map is to assign a saliency value to each image pixel, which is introduced in 2D image processing. More recently, it has been extended to 3D mesh processing in different ways (e.g. Refs. [Gal and Cohen-Or 2006; Lee et al. 2005; Yamauchi et al. 2006]). Based on the theory of saliency of visual parts, Gal et al. [2006] propose a method for defining the salient geometric features for partial shape matching. Lee et al. extend the 2D saliency map to 3D meshes through the center-surround operation on Gaussian-weighted mean curvatures, and apply mesh saliency to mesh simplification and view selection. Yamauchi el al. combine similarity and Lee et al.'s saliency approaches for selecting stable and salient representative views of 3D shapes. Since the algorithm of mesh saliency proposed by Lee et al. [2005] is simple, fast, and feature-preserving in the scalar function space, it is a potential choice for extracting the critical points. In fact, Lee et al.'s mesh saliency computation is a geometry filter (smoothing) operation in terms of the mean curvature used with the center-surround mechanism. For combining saliency maps at different scales, the final mesh saliency is computed by applying the non-linear normalization of suppression to all scales.

Our approach is built on the technique of mesh saliency proposed by Lee et al. [2005]. In some sense, our algorithm can be considered an application of mesh saliency based on Lee et al.'s algorithm [Lee et al. 2005]. We call the extracted critical points based on mesh saliency *salient critical points*. We also find that the original mesh saliency algorithm can not provide the best results for our application in extracting critical points by testing some examples. In this paper, we explore the improved method for meshing saliency for different models possibly with small or large noise. Our improved method combines the different anisotropic bilateral filter operation [Fleishman et al. 2003; Jones et al. 2003], and an iterative procedure. Our strategy can bring the salient critical points in important regions to a lower number. In addition, our saliency method is suitable for most scalar functions, such as geodesic and atomic density [Natarajan et al. 2006], in addition to curvature and height functions. The direct application of our method is a hierarchical topological representation for meshes by combining the Morse-Smale complex.

2 Background

In this section, we review the necessary background about critical points and mesh saliency.

2.1 Critical Points

Let M denote a compact 2-manifold without boundary and $f : M \rightarrow \mathbb{R}$ denote a real-valued smooth function on M. Supposing a local coordinate system at a point $\mathbf{p} \in M$, point $\mathbf{p}$ is *critical* if the gradient of $\mathbf{p}$ vanishes; otherwise, it is *regular*. In general, critical points are classified as maxima (f decreases in all directions), minima (f increases in all directions), and saddles (f switches between decreasing and increasing four times around $\mathbf{p}$).

In this paper, we only consider that M is a triangulated mesh possibly with noise. Ni et al. [2004] have extended Morse theory to meshed manifolds with boundary. Here f is a piecewise-linear real function. Its values are defined on the vertices of the mesh M, and linearly interpolated within the edges and triangles of the mesh. Typically, we also call f the Morse function of M. Suppose each edge $\langle \mathbf{v}_1, \mathbf{v}_2 \rangle \in M$ that $f(\mathbf{v}_1) \neq f(\mathbf{v}_2)$. Therefore, the gradient is constant, non-zero, and well defined across the interiors of the faces and edges; critical points occur at the vertices [Ni et al. 2004]. The degenerate flat edge limitation can be overcome by perturbation or Conley index theory [Ni et al. 2004]. We follow the general method [Banchoff 1970; Ni et al. 2004] using one neighborhood for defining the local neighborhood of a vertex $\mathbf{v} \in M$. A vertex is labelled a maximum/minimum if its function value is higher/lower than that of its neighbors, regular if its lower neighbors form a connected chain, and a saddle, otherwise.

2.2 Mesh Saliency

Lee et al. [2005] introduced the idea of mesh saliency as a measure of regional importance for graphics meshes, and explored the applications of mesh saliency to mesh simplification and view selection. The basic idea is to filter the curvatures of vertices in meshes using a center-surround operator on Gaussian-weighted mean curvatures. We first summarize the algorithm of mesh saliency as follows.

1. Compute the curvature of each vertex $\mathbf{v} \in M$ using Taubin's method [Taubin 1995]. Let $\mathscr{C}(\mathbf{v})$ denote the mean curvature of $\mathbf{v}$.

2. Compute the Gaussian-weighted average of the mean curvature at each vertex $\mathbf{v}$ as follows:

$$G(\mathscr{C}(\mathbf{v}), \sigma) = \frac{\sum\limits_{\mathbf{x} \in N(\mathbf{v}, 2\sigma)} \mathscr{C}(\mathbf{x}) \exp[-\|\mathbf{x} - \mathbf{v}\|^2/(2\sigma^2)]}{\sum\limits_{\mathbf{x} \in N(\mathbf{v}, 2\sigma)} \exp[-\|\mathbf{x} - \mathbf{v}\|^2/(2\sigma^2)]}, \quad (1)$$

   where $N(\mathbf{v}, \sigma)$ denotes a neighborhood for a vertex $\mathbf{v}$ and includes the set of vertices within a distance σ with $\mathbf{v}$.

3. Compute the saliency $\mathscr{S}(\mathbf{v})$ of a vertex $\mathbf{v}$ as the absolute difference of the Gaussian-weighted average (i.e. Eq. (1)) between fine and coarse scales:

$$\mathscr{S}(\mathbf{v}) = |G(\mathscr{C}(\mathbf{v}), \sigma) - G(\mathscr{C}(\mathbf{v}), 2\sigma)|. \quad (2)$$

4. The final mesh saliency is computed by adding the saliency maps (i.e. Eq. (2)) at five scales after applying a non-linear normalization of suppression.

The mesh saliency algorithm is essentially an anisotropic filter or smoothing operation for the mean curvature function. Our goal is to improve the mesh saliency algorithm for applying to critical points extraction.

3 Salient Critical Points

3.1 Disadvantages of Mesh Saliency

By testing some examples, we find that the original mesh saliency algorithm can not offer the best results in our application for extracting critical points. By analyzing the original algorithm, we find two disadvantages for our application.

One disadvantage is that Eq. (2) might make the same saliency for two opposite and symmetric vertices because of using the absolute difference between the Gaussian-weighted average. For instance, the standard example in Morse theory is the height (e.g. z coordinate) function over a torus standing on its side. In this case, there are one maximum and one minimum in the outer ring, and two saddles in the inner ring. Suppose the torus is centered at the original point with a height of $2h$; the maximum (h) and minimum ($-h$) have the opposite height value and the same neighborhood structure. However, the saliency values of the two vertices are same if using the absolute difference in Eq. (2). Similarly, more incorrect critical points are introduced because of the symmetry of the torus, where the lowest point is incorrectly classified as maximum. Therefore, the use of the absolute difference in Eq. (2) is not appropriate for extracting the critical points of a 3D mesh because it will change the type of critical points.

The other disadvantage is that Step 4 is difficult for controlling the number of critical points because it only combines saliency maps of five scales. If the chosen scales are not appropriate, the Morse function might be extremely non-smooth. For example, an inappropriate scale yields some extra critical points for a smooth torus surface. Conversely, if more close scales are utilized in saliency computation, over-smoothing might also occur, resulting in some salient critical points being missed. Ni et al. [2004] solve a relaxed form of Laplace equation to find a smooth Morse function with a user-controlled number of critical points. It is non-trivial to build a similar equation system for Gaussian operation. To overcome this problem, we use an anisotropic smoothing operation with an iterative procedure instead of using five scales and solving an equation system. The saliency of critical points is preserved by an appropriate anisotropic filter operation, and the number of critical points is controlled by the iterative number. The more the iterative number, the less the number of salient critical points. We will introduce the improved algorithm in the next sections.

3.2 Anisotropically Smoothing Morse Function

Filtering is a fundamental operation of image processing and computer vision. It means that the value of the filtered image at a given location is a function of the values of the input image in a small neighborhood of the same location. Similarly, we can regard the Morse function value of a vertex on a mesh as the gray value of a pixel in an image, and the connective neighborhood of the vertex as the neighborhood of the pixel. A part of the research in this paper is an extension from image smoothing to smoothing Morse function on meshes.

Smoothing a Morse function with a Laplacian filter [Ni et al. 2004] is an efficient technique for cancelling many pairs of unnecessary critical points. However, this technique is isotropic, and therefore has indiscriminately smooth noise and salient features, so this might skip some salient critical points. The idea of our approach is to modify the Laplacian diffusion equation to make it anisotropic. In particular, the Gaussian filter used for computing the mesh saliency computes a weighted average of values in the neighborhood, in which the weights decrease with the distance from the neighborhood center. Alternatively, the bilateral filter, introduced by Tomasi and Manduchi [1998], is another anisotropic filter derived from the Gaussian filter, with a feature preservation term that decreases the weight of pixels as a function of intensity difference. In this section, we will describe the above smoothing operation of bilateral filter for anisotropically smoothing the Morse function.

Similar to the second step in mesh saliency, we select the bilateral filter operation [Fleishman et al. 2003; Jones et al. 2003] instead of the simple Gaussian filter. We give the equation of the bilateral smoothing operation for the real function $f(\mathbf{v})$ on each vertex $\mathbf{v}$ as follows:

$$B(f(\mathbf{v}),\sigma)=\frac{\sum\limits_{\mathbf{x}\in N(\mathbf{v},2\sigma)} f(\mathbf{x})W_c(\|\mathbf{x}-\mathbf{v}\|)W_s(|f(\mathbf{x})-f(\mathbf{v})|)}{\sum\limits_{\mathbf{x}\in N(\mathbf{v},2\sigma)} W_c(\|\mathbf{x}-\mathbf{v}\|)W_s(|f(\mathbf{x})-f(\mathbf{v})|)}, \quad (3)$$

where the closeness smoothing filter is the standard Gaussian filter with parameter $\sigma_c : W_c(x) = \exp[-x^2/(2\sigma_c^2)]$, and a feature-preserving weight function with parameter σ_s that penalizes large variation in intensity is: $W_s(x) = \exp[-x^2/(2\sigma_s^2)]$. Compared with the Gaussian operation, the output of Eq. (3) on a vertex $\mathbf{v}$ is also a weighted average of the surrounding vertices, but the weight depends not only on the spatial distance $\|\mathbf{x}-\mathbf{v}\|$, but also on the scalar function difference $|f(\mathbf{x})-f(\mathbf{v})|$. The bilateral smoothing may be regarded as an anisotropic filter consisting of two Gaussian operations both on the spatial distance $\|\mathbf{x}-\mathbf{v}\|$ and the scalar function difference $|f(\mathbf{x})-f(\mathbf{v})|$.

3.3 Saliency Computation

Being different from Step 3 in mesh saliency, we do not adopt the absolute difference between the Gaussian-weighted average of the mean curvature as the vertex's saliency. The disadvantage of the absolute difference has been discussed in Section 3.1. We define the saliency on a vertex $\mathbf{v}$ as the Gaussian-weighted average of the scalar function difference between its neighboring vertices and $\mathbf{v}$, where the weight is similar to the weight in Eq. (3). By combining the bilateral filter in Eq. (3), we also get the equation of saliency computation:

$$\mathscr{S}_B(f(\mathbf{v}),\sigma)=\frac{\sum\limits_{\mathbf{x}\in N(\mathbf{v},2\sigma)} (f(\mathbf{x})-f(\mathbf{v}))W_c(\|\mathbf{x}-\mathbf{v}\|)W_s(|f(\mathbf{x})-f(\mathbf{v})|)}{\sum\limits_{\mathbf{x}\in N(\mathbf{v},2\sigma)} W_c(\|\mathbf{x}-\mathbf{v}\|)W_s(|f(\mathbf{x})-f(\mathbf{v})|)}. \quad (4)$$

3.4 The Algorithm Implementation

In this section, we give the whole algorithm of extracting salient critical points. First, we apply a bilateral filter to a single vertex $\mathbf{v}$ for computing its saliency value s. After finishing the saliency computation on all vertices, update the real function value on each vertex $\mathbf{v}$ as: $\hat{f} = f + s$. Finally, after several iterative saliency computations, we utilize the computed saliency for extracting the critical points by Banchoff's method [Banchoff 1970; Ni et al. 2004]. The following is the pseudo-code for the algorithm of extracting the salient critical points:

Listing 1: The fundamental **salient critical points extraction** algorithm.

```
Procedure SalientCriticalPoints($M, \sigma, iteration$);
  Input:
    $M$: the given mesh
    $\sigma$: the radius of the neighborhood
    $iterations$: the number of iterations
  Output:
    salient critical points
  Local variables:
    $N$: the size of $M$
    $\mathbf{v}_i$: the $i$ vertex of $M$
    $f_i$: the real function value on $\mathbf{v}_i$
    $s_i$: the computed saliency value on $\mathbf{v}_i$
begin
  for($j$=1 to $iterations$)
    $\{\mathbf{v}_i\}$=vertices($M$);
    $N$=$|\{\mathbf{v}_i\}|$;
    for($i$=1 to $N$)
      $f_i$=function($\mathbf{v}_i$);
      $s_i$=computate $\mathbf{v}_i$'s saliency using Eq.(4);
    end
    update vertices functions as: $f_i = f_i + s_i$;
  end
  classify vertices using ClassifyVertex;
  return extracting salient critical points;
end
```

The **ClassifyVertex** algorithm referred to in Listing 1 is a direct implementation of vertex classification [Ni et al. 2004]. The types of returning vertices in the algorithm include maximum, minimum, and saddle. The reader may consult the reference (Section 3.1 in [Ni et al. 2004]) for detailed expositions of the algorithm.

Parameters. The parameters of the algorithm are: the radius of neighborhood σ, σ_c, σ_s, and the number of iterations. Similar to Lee et al.'s strategy [Lee et al. 2005] for choosing the neighborhood radius σ, we choose $\sigma = 8.0\varepsilon$, where ε is 0.3% of the length of the diagonal of the bounding box of the model. We use $\sigma_c = \sigma$ as the standard deviation of the Gaussian filter at distance. In addition, we suggest two methods for choosing σ_s. One method is that σ_s is simply equal to σ_c. The other method is that σ_s is equal to the maximal difference among real functions of vertices in the neighborhood of a vertex. We have tried both and found that the first method gives us better results, so we choose $\sigma_c = \sigma_s = \sigma$ in our implementation. One may choose a large σ and perform a few iterations, or choose a narrow filter and increase the number of iterations. In all results shown in this paper, we use 10 to 20 iterations. We find that a small number of iterations is sufficient and advantageous both for the speed of computation and for the number of salient critical points. Furthermore, we use a *kd*-tree to accelerate the searching speed of points in a sphere with the radius σ and the center at vertices of the mesh.

4 Results and Discussion

Critical points always depend on a Morse function f, which is a real-valued function defined on a surface. The definition of the Morse function for a more general surface is non-trivial, and it depends closely on applications. If a different Morse function is used as f, the critical points might change. There are many different definitions for the Morse function. For instance, in terrain modeling applications, the height function has been a useful function f [Bremer et al. 2004]. However, the height function is not invariant to transformations such as object rotation. The curvature function may provide invariance in a rotation, but it is sensitive to the noise of the surface. For segmenting molecular surfaces, Natarajan et al. [2006] chose an atomic density function. This paper does not focus on how to define new Morse functions. In all the results shown in this paper, we adopt the mean curvature function as a Morse function for testing the effectiveness of our method because of its invariance in a rotation. We use Taubin's method [Taubin 1995] for curvature computation. In fact, any other Morse function can also be suitable for our method.

Table 1: Comparisons of time and the number of critical points.

Model	Fig.	#Verts	#Iters	Time(s)	σ	#CP
Lion	1(b)	24K	-	-	-	7629
	1(d)	24K	10	3.9	8.0ε	493
Squirrel	3(b)	10K	-	-	-	2864
	3(d)	10K	20	1.4	8.0ε	152
Protein	4(a)	8K	1	0.13	8.0ε	741
	4(b)	8K	10	0.63	8.0ε	257

We have implemented the algorithm of extracting salient critical points as described in the previous section and applied it to some models. The algorithm described above is implemented in C++. The execution time is given in seconds on a Pentium IV 1.70GHz processor with 512M RAM excluding the time of loading meshes. This section investigates the effectiveness of our method by comparing our results with the results of the conventional critical points extraction algorithm [Banchoff 1970; Ni et al. 2004], and the cancelled critical points algorithm based on topological persistence and simplification (TPS) [Edelsbrunner et al. 2002]. Finally, we also give an example for showing a direct application to a hierarchical topological representation.

Table 1 gives the time in seconds for some meshes referred to in this paper, where "#Verts" is the number of vertices of the models, "#Iters" is the number of iterations, and "#CP" is the number of critical points. In all the results shown in this section, we use 10 to 20 iterations. We found a small number of iterations is sufficient both for the speed of computation and for the numerical stability.

Example 1 Comparisons with the conventional critical points extraction algorithm [Banchoff 1970] (Refer to Figures 1, 2). The back of the lion head model includes large noise and many small hair textures. The mean curvature function yields many critical points due to the curvature function's sensitivity to noise (see Figure 1(b)). Our method based on mesh saliency can filter some noise and capture the important features, and yields the salient critical points in the more interesting region of hairs with a lower number after 10 iterations, as shown in Figure 1(d)).

For showing the quality of critical points, we also illustrate the topological segmentation of salient critical points using the Morse-Smale (MS) complex. The MS complex [Edelsbrunner et al. 2003] is a cellular decomposition of a scalar function over a manifold, defined formally as the refinement of its ascending manifolds by its descending manifolds [Edelsbrunner et al. 2003; Dong et al. 2006]. The MS complex is a power tool, and has been applied to many graphics techniques, such as topological simplification and hierarchy [Bremer et al. 2004; Edelsbrunner et al. 2002; Edelsbrunner et al. 2003; Natarajan et al. 2006], cutting a surface into a Disk [Ni et al. 2004], and surface quadrangulation [Dong et al. 2006]. Figure 2 gives the MS complex comparison corresponding to the lion head model in Figure 1. The original MS complex (see Figure 2(a)) is too complex for some applications due to noise. Salient critical points using our method can generate a more simplified and uni-

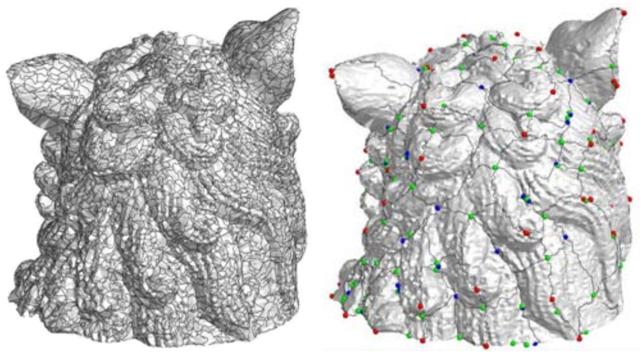
(a) (b)

Figure 2: The MS complex comparison with the conventional critical points extraction algorithm. (a) The full MS complex of the original model in Figure 1(b). (b) The MS complex corresponding to Figure 1(d). Note our method generates a more simplified MS complex due to the reduction of critical points.

form MS complex (see Figure 2(b)). Our method can also serve for the above applications, including surface segmentation [Natarajan et al. 2006], cutting [Ni et al. 2004], and remeshing [Dong et al. 2006].

Example 2 Comparison with TPS (Refer to Figure 3). The algorithm of topological persistence and simplification, i.e. TPS, is an alternative method for cancelling critical points [Edelsbrunner et al. 2002]. In general, the persistence of a critical point pair is defined as the absolute difference in the values of f between the two points [Edelsbrunner et al. 2003]. Figure 3 gives an example for comparing our method with TPS. A mean curvature function yields a mass of critical points due to noise, and the full MS complex is generated in Figure 3(a). In Figure 3(b), the TPS method is used for removing all critical points with persistence less than 0.1% of the mean curvature range. Figure 3(c) shows the salient critical points after 20 iterations using our method with the close number of critical points generated by TPS. Observe the difference in the area of the left eye in Figures 3(d) and 3(e). Note that TPS still has some critical points remaining around this area because of the high noise points in this region with a persistence of more than 0.1%. Conversely, our method can only leave one salient critical point around the same area.

Example 3 Hierarchical topological representation (Refer to Figure 4). The efficient construction of topologically simplified models is an important problem in computer graphics and geometric modeling [Edelsbrunner et al. 2002]. Its goal is to remove topological noise and leave the topological features. Edelsbrunner et al. [2002] formalized a notion of topological simplification within the framework of a filtration, and presented a topological simplification algorithm based on persistence. Based on the proposed persistence computation algorithm for cancelling pairs of critical points, Edelsbrunner et al. [2003] built hierarchical MS complexes for simplifying the topological structure of piecewise linear 2-manifolds. Similar to their work, our method can also provide a hierarchical topological representation for meshes at each different iterative level by combining the MS complex. The difference between our method and the method based on cancelling critical pairs is that our smoothing process naturally gives the hierarchical critical points. The set of MS complexes at different iterations introduces a hierarchical topological representation for meshes. The proposed hierarchical topological structure based on salient critical points and iterations can be regarded as a beneficial supplement of the known algorithms. Figure 4 shows a hierarchical example of a molecular surface including 1, 5, and 10 iterations. In this example, we use the structure of an enzyme, phospholipase A2 (Protein Data Bank code: 2bp2). Enzymes carry out their catalytic reaction on a local site of the molecular surface. Therefore, developing a methodology for segmenting a molecular surface in a biologically meaningful way is very useful for characterizing and also predicting the function of proteins. Detecting salient critical points is suitable for application to protein surface, because fewer (hence larger) and significant segments are obtained, which would be more tolerant to small conformational changes of molecular surface due to the intrinsic flexibility of proteins. Our method might move positions of critical points after several iterations.

5 Conclusions

We have presented a simple, robust, and efficient algorithm for extracting the salient critical points on meshes by combining mesh saliency with Morse theory. The basic idea of the proposed method is first to assign a saliency value to each vertex of meshes as the saliency map using the anisotropic bilateral filter, and then extract the salient critical points using Morse theory. We also find that the original mesh saliency method can not be applied directly to our method due to several drawbacks, so an improved iterative implementation of the proposed algorithm is provided. Our method is essentially an anisotropic filter on the Morse function. In addition, we compared our results with the results of the conventional methods, and the comparisons show our method can offer more satisfactory results, especially for the noisy meshes. Finally, we demonstrated a direct application to a hierarchical topological representation for meshes by combining the Morse-Smale complex.

The major drawback in our current implementation is that the number of salient critical points is dependent on the iteration number and can not be accurately controlled. Ni et al. [2004] can control the number of critical points by solving a linear Laplace equation system. However, it is not easy to build a similar linear Gaussian equation system and solve it effectively. In the future we plan to explore research of the bounded number of critical points for the Gaussian equation system. In some sense, the new method still has the same drawbacks as the other iterative methods. That is, the termination condition is hard to control. In our current implementation, the iteration number is the only condition for termination. In the future work we plan to add the other termination conditions to make our method more effective.

Acknowledgements This work was supported in part by the National Institute of General Medical Sciences of the National Institutes of Health (R01 GM075004) awarded to Professor Kihara and Professor Ramani.

References

BANCHOFF, T. 1970. Critical points and curvature for embedded polyhedral surfaces. *American Mathematical Monthly 77*, 5, 475–485.

BREMER, P.-T., EDELSBRUNNER, H., HAMANN, B., AND PASCUCCI, V. 2004. A topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics 10*, 4, 385–396.

DONG, S., BREMER, P.-T., GARLAND, M., PASCUCCI, V., AND HART, J. C. 2006. Spectral surface quadrangulation. In *Proceedings of SIGGRAPH 2006*, 1057–1066.

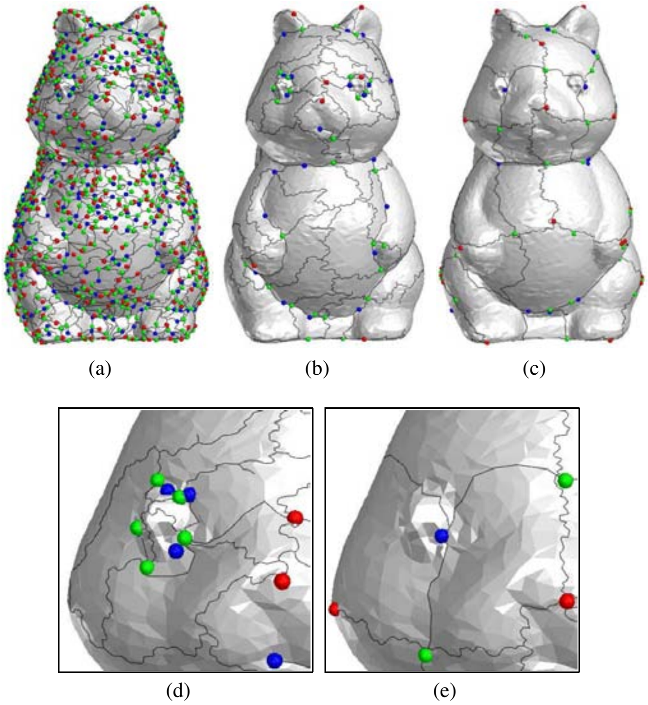

Figure 3: Comparison with TPS of a squirrel model with light noise. (a) A mass of critical points yielded by a mean curvature function, and the full MS complex. (b) Cancelling some critical points using TPS, and the simplified MS complex. (c) Salient critical points after 20 iterations using our method, and the corresponding MS complex. (d) The magnified view of (b) on the left eye of the model. (e) The magnified view of (c). There is a close number of critical points in both (b) and (c), where (d) has 141 critical points and (e) has 152 critical points.

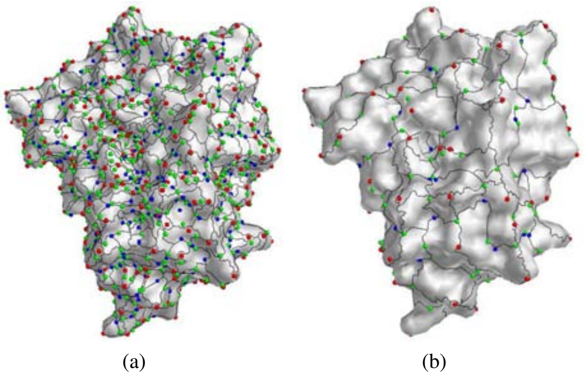

Figure 4: A hierarchical topological representation for a molecular surface by combining our method with the MS complex. (a) Salient critical points after one iteration and the corresponding MS complex. (b) 10 iterations.

EDELSBRUNNER, H., LETSCHER, D., AND ZOMORODIAN, A. 2002. Topological persistence and simplification. *Discrete & Computational Geometry 28*, 4, 511–533.

EDELSBRUNNER, H., HARER, J., AND ZOMORODIAN, A. 2003. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete & Computational Geometry 30*, 1, 87–107.

FLEISHMAN, S., DRORI, I., AND COHEN-OR, D. 2003. Bilateral mesh denoising. In *Proceedings of SIGGRAPH 2003*, 950–953.

GAL, R., AND COHEN-OR, D. 2006. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics 25*, 1, 130–150.

JONES, T., DURAND, F., AND DESBRUN, M. 2003. Non-iterative, feature-preserving mesh smoothing. In *Proceedings of SIGGRAPH 2003*, 943–949.

LEE, C.-H., VARSHNEY, A., AND JACOBS, D. 2005. Mesh saliency. In *Proceedings of SIGGRAPH 2005*, 659–666.

NATARAJAN, V., WANG, Y., BREMER, P.-T., PASCUCCI, V., AND HAMANN, B. 2006. Segmenting molecular surfaces. *Computer Aided Geometric Design 23*, 6, 495–509.

NI, X., GARLAND, M., AND HART, J. C. 2004. Fair Morse functions for extracting the topological structure of a surface mesh. In *Proceedings of SIGGRAPH 2004*, 613–622.

TAUBIN, G. 1995. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Proceedings of International Conference on Computer Vision*, 902–907.

TOMASI, C., AND MANDUCHI, R. 1998. Bilateral filtering for gray and color images. In *Proceedings of IEEE International Conference on Computer Vision*, 836–846.

YAMAUCHI, H., SALEEM, W., YOSHIZAWA, S., KARNI, Z., BELYAEV, A., AND SEIDEL, H.-P. 2006. Towards stable and salient multi-view representation of 3D shapes. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, 265–270.

YAMAZAKI, I., NATARAJAN, V., BAI, Z., AND HAMANN, B. 2006. Segmenting point sets. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, 4–13.