Mesh Saliency

Chang Ha Lee Amitabh Varshney David W. Jacobs

Department of Computer Science
University of Maryland, College Park, MD 20742, USA
{chlee, varshney, djacobs}@cs.umd.edu

Abstract

Research over the last decade has built a solid mathematical foundation for representation and analysis of 3D meshes in graphics and geometric modeling. Much of this work however does not explicitly incorporate models of low-level human visual attention. In this paper we introduce the idea of *mesh saliency* as a measure of regional importance for graphics meshes. Our notion of saliency is inspired by low-level human visual system cues. We define mesh saliency in a scale-dependent manner using a center-surround operator on Gaussian-weighted mean curvatures. We observe that such a definition of mesh saliency is able to capture what most would classify as visually interesting regions on a mesh. The human-perception-inspired importance measure computed by our mesh saliency operator results in more visually pleasing results in processing and viewing of 3D meshes, compared to using a purely geometric measure of shape, such as curvature. We discuss how mesh saliency can be incorporated in graphics applications such as mesh simplification and viewpoint selection and present examples that show visually appealing results from using mesh saliency.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling; I.3.m [Computer Graphics]: Perception; I.3.m [Computer Graphics]: Applications

Keywords: saliency, visual attention, perception, simplification, viewpoint selection

1 Introduction

We have witnessed significant advances in the theory and practice of 3D graphics meshes over the last decade. These advances include efficient and progressive representation [Hoppe 1996; Karni and Gotsman 2000], analysis [Taubin 1995; Kobbelt et al. 1998; Meyer et al. 2003], transmission [Al-Regib et al. 2005], and rendering [Luebke et al. 2003] of very large meshes. Much of this work has focussed on using mathematical measures of shape, such as curvature. The rapid growth in the number and quality of graphics meshes and their ubiquitous use in a large number of human-centered visual computing applications, suggest the need for incorporating insights from human perception into mesh processing. Although excellent work has been done in incorporating principles of perception in managing level of detail for rendering meshes [Luebke and Hallen 2001; Reddy. 2001; Watson et al. 2004], there has been less attention paid to the use of perception-inspired metrics for processing of meshes.

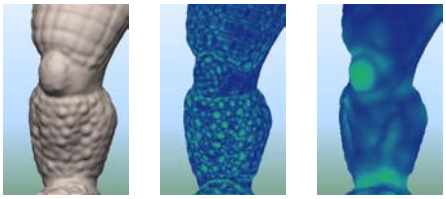
(a) (b)

Figure 1: Mesh Saliency: Image (a) shows the Stanford Armadillo model, and image (b) shows its mesh saliency.

Our goal in this paper is to bring perception-based metrics to bear on the problem of processing and viewing 3D meshes. Purely geometric measures of shape such as curvature have a rich history of use in the mesh processing literature. For instance, Heckbert and Garland [1999] show that their quadric error metric is directly related to the surface curvature. Mesh simplifications resulting from minimizing the quadric errors result in provably optimum aspect ratio of triangles in the L_2 norm, as the triangle areas approach zero. However, a purely curvature-based metric may not necessarily be a good metric of perceptual importance. For example, a high-curvature spike in the middle of a largely flat region will be likely perceived to be important. However, it is also likely that a flat region in the middle of densely repeated high-curvature bumps will be perceived to be important as well. Repeated patterns, even if high in curvature, are visually monotonous. It is the unusual or unexpected that delights and interests. As an example, the textured region with repeated bumps in the leg of the Armadillo shown in Figure 2(a) is arguably visually less interesting than an isolated but smooth feature such as its knee (Figure 2(c)).

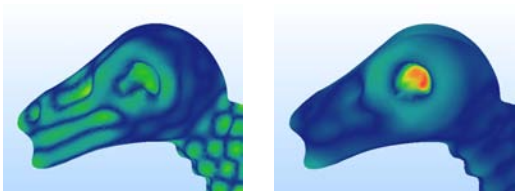
In this paper, we introduce the concept of *mesh saliency*, a measure of regional importance, for 3D meshes, and present a method to compute it. Our method to compute mesh saliency uses a center-surround mechanism. We use the center-surround mechanism because it has the intuitive appeal of being able to identify regions that are different from their surrounding context. We are also encouraged by the success of these mechanisms on 2D problems.

We expect a good model of saliency to operate at multiple scales, since what is interesting at one scale need not remain so at a different scale. A good saliency map should capture the interesting features at all perceptually meaningful scales. Figure 3(a) shows a saliency map at a fine scale where small features such as the nose and mouth have high saliency, while a saliency map at a larger scale (Figure 3(b)) shows the eye to have a higher saliency. We use these observations to define a multi-scale model of mesh saliency using the center-surround mechanism in Section 3. A number of tasks in graphics can benefit from a computational model of mesh saliency.



(a) (b) (c)

Figure 2: Curvature alone is inadequate for assessing saliency since it does not adequately consider the local context. Image (a) shows a part of the right leg of the Stanford Armadillo model. Image (b) visualizes the magnitude of mean curvatures and (c) shows our saliency values. While (b) captures repeated textures and fails to capture the knee, (c) successfully highlights the knee.



(a) (b)

Figure 3: Saliency is relative to the scale. Image (a) shows the saliency map of the Cyberware Dinosaur head at a small scale, and image (b) shows the map of its saliency at a larger scale. In image (a), the small-scale saliency highlights the small features such as nose and mouth and in image (b), the large-scale saliency identifies a larger feature such as the eye.

In this paper we explore the application of mesh saliency to mesh simplification and view selection in Sections 4 and 5.

The main contributions of this paper are:

1. **Saliency Computation:** There can be a number of definitions of saliency for meshes. We outline one such method for graphics meshes based on the Gaussian-weighted center-surround evaluation of surface curvatures. Our method has given us very promising results on several 3D meshes.
2. **Salient Simplification:** We discuss how traditional mesh simplification methods can be modified to accommodate saliency in the simplification process. Our results show that saliency-guided simplification can easily preserve visually salient regions in meshes that conventional simplification methods typically do not.
3. **Salient Viewpoint Selection:** As databases of 3D models evolve to very large collections, it becomes important to automatically select viewpoints that capture the most salient attributes of objects. We present a saliency-guided method for viewpoint selection that maximizes visible saliency.

We foresee the computation and use of mesh saliency as an increasingly important area in 3D graphics. As we engage in image synthesis and analysis for ever larger graphics datasets and as the gap between processing capabilities and memory-access times grows ever wider, the need for prioritizing and selectively processing graphics datasets will increase. Saliency can provide an effective tool to help achieve this.

2 Related Work

Low-level cues influence where in an image people will look and pay attention. Many computational models of this have been proposed. Koch and Ullman's [1985] early model suggested that salient image locations will be distinct from their surroundings. Our approach is explicitly based on the model of Itti *et al.* [1998]. They combine information from center-surround mechanisms applied to different feature maps, computed at different scales, to compute a *saliency map* that assigns a saliency value to each image pixel. Tsotsos *et al.* [1995], Milanese *et al.* [1994], Rosenholtz [1999], and many others describe other interesting saliency models. Among their many applications, 2D saliency maps have been applied to selectively compress [Privitera and Stark 1999] or shrink [Chen et al. 2003; Suh et al. 2003] images. DeCarlo and Santella [2002] use saliency determined from a person's eye movements to simplify an image producing a non-photorealistic, painterly rendering.

More recently, saliency algorithms have been applied to views of 3D models. Yee *et al.* [2001] use Itti *et al.*'s algorithm to compute a saliency map of a coarsely rendered 2D projection of a 3D dynamic scene. They use this to help decide where to focus computational resources in producing a more accurate rendering. Mantiuk *et al.* [2003] use a real-time, 2D saliency algorithm to guide MPEG compression of an animation of a 3D scene. Frintrop *et al.* [2004] use a saliency map to speed up the detection of objects in 3D data. They combine saliency maps computed from 2D images representing scene depth and intensities. Howlett [2004] demonstrate the potential value of saliency for the simplification of 3D models. Their work captures saliency by using an eye-tracker to record where a person has looked at a 2D image of a 3D model.

These prior works determine saliency for a 3D model by finding saliency in its 2D projection. There is little work that determines saliency directly from 3D structure. Guy and Medioni [1996] proposed a method for computing a saliency map for edges in a 2D image, (such edge-based saliency maps were previously explored by Shashua and Ullman [1988]). In [Medioni and Guy 1997] they extend this framework to apply to 3D data. However, their approach is mainly designed to smoothly interpolate sparse, noisy 3D data to find surfaces. They do not compute an analog to the saliency map for a 3D object. Watanabe and Belyaev [2001] have proposed a method to identify regions in meshes where principal curvatures have locally maximal values along one of the principal directions (typically along ridges and ravines). Hisada *et al.* [2002] have proposed a method to detect salient ridges and ravines by computing the 3D skeleton and finding non-manifold points on the skeletal edges and associated surface points.

3 Mesh Saliency Computation

Itti *et al.* [1998]'s method is one of the most effective techniques for computing saliency for 2D images. Our method for computing saliency for 3D meshes uses their center-surround operation. Unlike images, where color is the most important attribute, we consider geometry of meshes to be the most important contributor to saliency. At present our method for mesh saliency uses only geometry, but it should be easy to incorporate other surface appearance attributes into it as well. There are several possible characteristics of mesh geometry that could be used for saliency. Before we decide on one let us compare the desiderata of saliency in a 2D image with the saliency of a 3D object. Zero saliency in an image corresponds to a region with uniform intensity. The motivation behind this is that the key image property whose variations are critical is the intensity. In an image, intensity is a function of shape and lighting. For 3D objects however, we have the opportunity to determine the saliency

based on shape, independent of lighting. For 3D objects, we feel that a sphere is the canonical zero-saliency feature. This is in spite of the fact that depending on the lighting, a sphere may not produce a uniform intensity image. In the case of the sphere the property that is invariant is the curvature. Therefore we are guided by the intuition that it is changes in the curvature that lead to saliency or non-saliency. This has led us to formulate mesh saliency in terms of the mean curvature used with the center-surround mechanism. Figure 4 gives an overview of our saliency computation.

The first step of our saliency computation involves computing surface curvatures. There are a number of excellent approaches that generalize differential-geometry-based definition of curvatures to discrete meshes [Taubin 1995; Meyer et al. 2003]. One can use any of these to compute the curvature of a mesh at a vertex v. Let the curvature map $\mathscr{C}$ define a mapping from each vertex of a mesh to its mean curvature, i.e. let $\mathscr{C}(v)$ denote the mean curvature of vertex v. We use Taubin [1995]'s method for curvature computation. Let the neighborhood $N(v,\sigma)$ for a vertex v, be the set of points within a distance σ. One can consider several distance functions to define the neighborhood, such as the geodesic or the Euclidean. We have tried both and found that the Euclidean distance gave us better results and that is what we use here. Thus, $N(v,\sigma) = \{x | \|x-v\| < \sigma,$ x is a mesh point$\}$. Let $G(\mathscr{C}(v),\sigma)$ denote the Gaussian-weighted average of the mean curvature. We compute this as:

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

Note that with the above formulation, we are assuming a cut-off for the Gaussian filter at a distance 2σ. We compute the saliency $\mathscr{S}(v)$ of a vertex v as the absolute difference between the Gaussian-weighted averages computed at fine and coarse scales. We currently use the standard deviation for the coarse scale as twice that of the fine scale:

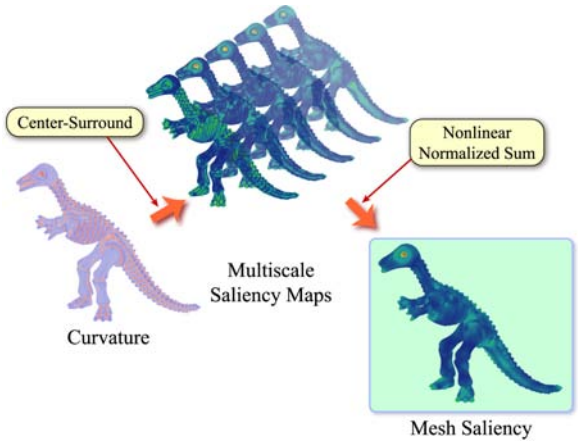


Figure 4: Mesh Saliency Computation: We first compute mean curvature at mesh vertices. For each vertex, saliency is computed as the difference between mean curvatures filtered with a narrow and a broad Gaussian. For each Gaussian, we compute the Gaussian-weighted average of the curvatures of vertices within a radius 2σ, where σ is Gaussian's standard deviation. We compute saliency at different scales by varying σ. The final saliency is the aggregate of the saliency at all scales with a non-linear normalization.

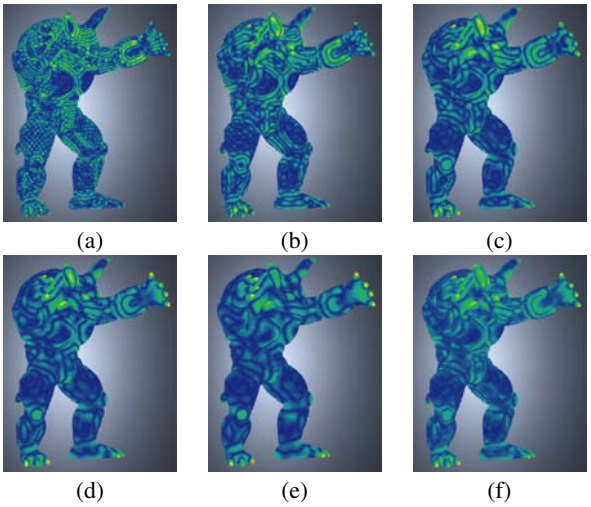


Figure 5: Images (a)–(e) show the saliency at scales of 2ε, 3ε, 4ε, 5ε, and 6ε. Image (f) shows the final mesh saliency after aggregating the saliency over multiple scales. Here, ε is 0.3% of the length of the diagonal of the bounding box of the model.

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

To compute mesh saliency at multiple scales, we define the saliency of a vertex v at a scale level i as $\mathscr{S}_i(v)$:

$$\mathscr{S}_i(v) = |G(\mathscr{C}(v),\sigma_i) - G(\mathscr{C}(v),2\sigma_i)|$$

where, σ_i is the standard deviation of the Gaussian filter at scale i. For all the results in this paper we have used five scales $\sigma_i \in \{2\varepsilon, 3\varepsilon, 4\varepsilon, 5\varepsilon, 6\varepsilon\}$, where ε is 0.3% of the length of the diagonal of the bounding box of the model.

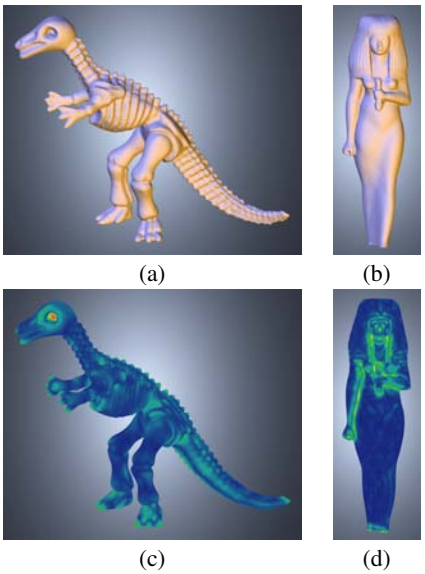
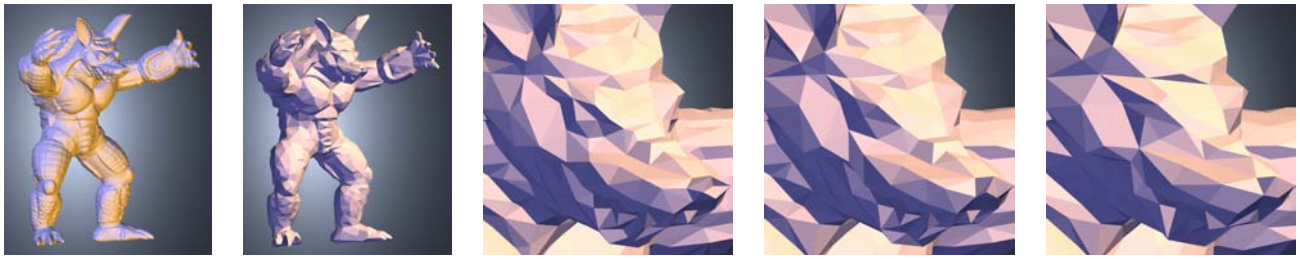
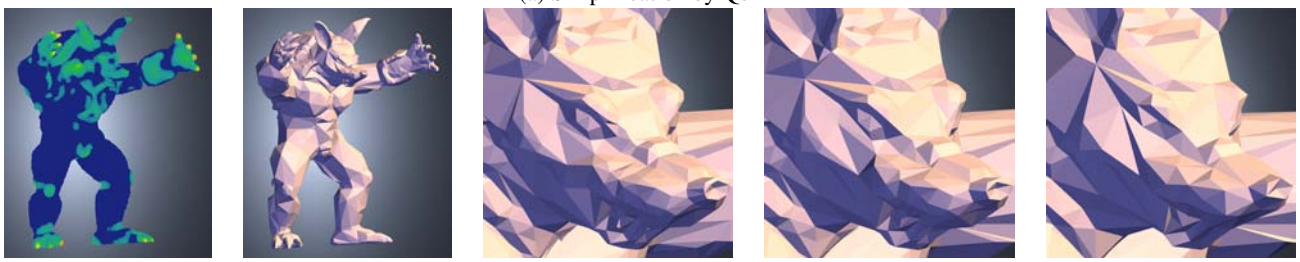


Figure 6: We show mesh saliency for the Cyberware Dinosaur model (a) in figure (c) and for the Cyberware Isis model (b) in figure (d). Warmer colors (reds and yellows) show high saliency and cooler colors (greens and blues) show low saliency.



Original (346K triangles) | 99% simplification (3.5K triangles) | 98% simplification (6.9K triangles) | 98.5% simplification (5.2K triangles) | 99% simplification (3.5K triangles)

(a) Simplification by Qslim



Saliency | 99% simplification (3.5K triangles) | 98% simplification (6.9K triangles) | 98.5% simplification (5.2K triangles) | 99% simplification (3.5K triangles)

(b) Simplification guided by saliency

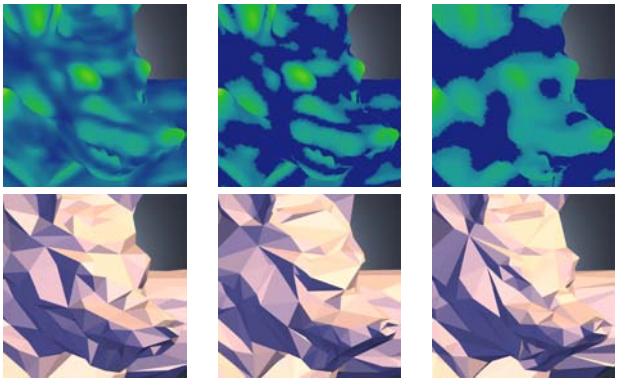
Figure 7: Simplification results for the Stanford Armadillo: (a) shows simplified models using Qslim and (b) shows different levels of simplification using saliency. The three right columns show the zoomed-in face of the Armadillo. The eyes and the nose are preserved better with our method while the bumps on the legs are smoothed faster.

For combining saliency maps $\mathscr{S}_i$ at different scales, we apply a non-linear suppression operator S similar to the one proposed by Itti *et al.* [1998]. This suppression operator promotes saliency maps with a small number of high peaks (Figure 5(e)) while suppressing saliency maps with a large number of similar peaks (Figure 5(a)). Thus, non-linear suppression helps us in reducing the number of salient points. If we do not use suppression, we get far too many regions being flagged as salient. We believe, therefore, that this suppression helps to define what makes something unique, and therefore potentially salient. For each saliency map $\mathscr{S}_i$, we first normalize $\mathscr{S}_i$. We then compute the maximum saliency value M_i and the average $\bar{m}_i$ of the local maxima excluding the global maximum at that scale. Finally, we multiply $\mathscr{S}_i$ by the factor $(M_i - \bar{m}_i)^2$. The final mesh saliency $\mathscr{S}$ is computed by adding the saliency maps at all scales after applying the non-linear normalization of suppression: $\mathscr{S} = \sum_i S(\mathscr{S}_i)$

4 Salient Simplification

There is a large and growing body of literature on simplification of meshes using a diverse set of error metrics and simplification operators [Luebke et al. 2003]. Several simplification approaches use estimates of mesh curvature to guide the simplification process and achieve high geometric fidelity for a given triangle budget [Turk 1992; Kim et al. 2002]. Other simplification approaches, such as QSlim [Garland and Heckbert 1997], use error metrics that while not directly computing curvature, are related to curvature [Heckbert and Garland 1999]. Curvature has also been directly used to identify salient regions on meshes. Watanabe and Belyaev [2001] classify extrema of the principal curvatures as salient features and preserve them better during simplification. Their method however, does not use a center-surround mechanism to identify regions on a mesh that are different from their local context.

For evaluating the effectiveness of our mesh saliency method, we have modified the quadrics-based simplification method (Qslim) of Garland and Heckbert [1997] by weighting the quadrics with mesh saliency. However, it should be equally easy to integrate our mesh saliency with any other mesh simplification scheme. Garland and Heckbert's method simplifies a mesh by repeatedly contracting vertex pairs ordered by increasing quadric errors. Let P be the set of planes of triangles incident at a vertex v, where the plane $p \in P$ defined by the equation $ax + by + cz + d = 0$, $a^2 + b^2 + c^2 = 1$, is represented as $(a\ b\ c\ d)^T$. Then the quadric for the plane p is de-



(a) (b) (c)

Figure 8: We show the saliency-based weights and the quality of the 99% simplification ($3.5K$ triangles) for the Stanford Armadillo model for three choices of the simplification weights: (a) the original mesh saliency ($\mathscr{W} = \mathscr{S}$) (b) the amplified mesh saliency ($\mathscr{W} = A\mathscr{S}$), and (c) the smoothed and amplified mesh saliency ($\mathscr{W} = A(G(\mathscr{S}, 3\varepsilon))$).

Original (606K tris) | 4K triangles (Smooth Shading) | 4K triangles | 2K triangles | 1K triangles

(a) Simplification by Qslim

Saliency | 4K triangles (Smooth Shading) | 4K triangles | 2K triangles | 1K triangles

(b) Simplification guided by saliency

Figure 9: Simplification results for the Cyberware Male: (a) shows simplifications by Garland and Heckbert's method, and (b) shows simplifications by our method using saliency. The eyes, nose, ears, and mouth are preserved better with our method.

fined as $Q_p = pp^T$. They define the error of v with respect to p as the squared distance of v to p which is computed by $v^T Q_p v$. The quadric Q of v is the sum of all the quadrics of neighboring planes: $Q = \sum_{p \in P} Q_p$. After computing quadrics of all vertices, they compute the optimal contraction point $\bar{v}$ for each pair (v_i, v_j) which minimizes the quadric error $\bar{v}^T (Q_i + Q_j)\bar{v}$ where Q_i and Q_j are quadrics of v_i and v_j, respectively. The algorithm iteratively contracts the pair with the minimum contraction cost $\bar{v}^T (Q_i + Q_j)\bar{v}$. After a pair is contracted, the quadric for the new point $\bar{v}$ is computed simply by adding the two quadrics $Q_i + Q_j$.

We guide the order of simplification contractions using a weight map $\mathscr{W}$ derived from the mesh saliency map $\mathscr{S}$. We have found that using the simplification weights based on a non-linear amplification of the saliency gives us good results. We believe that the reason behind this is that by amplifying the high saliency vertices we are ensuring that they are preserved longer than the non-salient vertices with high contraction costs. Specifically, we define a saliency amplification operator A using a threshold α and an amplifying parameter λ, such that we amplify the saliency values that are greater than or equal to α by a factor λ. Thus, the simplification weight map $\mathscr{W}$ using the saliency amplification operator A is specified as:

$$\mathscr{W}(v) = A(\mathscr{S}(v), \alpha, \lambda) = \begin{cases} \lambda \mathscr{S}(v) & \text{if } \mathscr{S}(v) >= \alpha \\ \mathscr{S}(v) & \text{if } \mathscr{S}(v) < \alpha \end{cases}$$

For all the saliency-based simplification results in this paper, we use $\lambda = 100$ and $\alpha = 30^{th}$ percentile saliency. At the initialization stage of computing the quadric Q for each vertex v, we multiply Q by its simplification weight $\mathscr{W}(v)$ derived from the saliency of v: $Q \leftarrow \mathscr{W}(v)Q$. Analogous to the computation of a quadric after a vertex-pair collapse, the simplification weight $\mathscr{W}(v)$ for the new vertex v is the sum of the weights for the pair of vertices being collapsed $\mathscr{W}(v_i) + \mathscr{W}(v_j)$.

Obviously, the quality of simplification increases when we apply the saliency amplifying operator. However, we have observed that even when we directly use the saliency as the weighting factor without the amplifying operator, i.e. with $\lambda = 1$, the interesting features are preserved longer than with the original quadric-based method.

We have also observed that blurring the saliency map before computing the amplified saliency gives us fewer salient regions and allows the simplification process to focus more on these selected regions. We use $\sigma = 3\varepsilon$ for blurring, i.e. $\mathscr{W} = A(G(\mathscr{S}, 3\varepsilon))$, This is shown in Figure 8. We compute the saliency map just once and do not modify it during simplification so that we can always stay true to the original model's saliency.

5 Salient Viewpoint Selection

With advances in 3D model acquisition technologies, databases of 3D models are evolving to very large collections. Accordingly, the importance of automatically crafting *best* views that maximally elucidate the most important features of an object has also grown for high-quality representative first views, or sequence of views. A number of papers have addressed the problem of automatically selecting a viewpoint for looking at an object. Kamada and Kawai [1988] describe a method for selecting views in which surfaces are imaged non-obliquely relative to their normals, using parallel projection. Stoev and Straßer [2002] consider a different approach that is more suitable to viewing terrains, in which most surface normals in the scene are similar, and visible scene depth should be maximized. In the context of computer vision, Weinshall and Werman [1997] show an equivalence between the most stable and most likely view of an object, and show that this is the view in which an object is flattest. Finding the optimal set of views of an object for purposes of image-based rendering has also been considered, using measures such as those providing best coverage of the scene [Fleishman et al. 1999], and those that provide the most information [Vázquez et al. 2002].

Blanz *et al.* [1999] have conducted user studies to determine the factors that influence the preferred views for 3D objects. They conclude that selection of a preferred view is a result of complex interactions between task, object geometry, and object familiarity. Their studies support visibility (and occlusion) of salient features of an object as one of the factors influencing the selection of a preferred view. Gooch *et al.* [2001] have built a system that uses art-inspired principles and some of the factors suggested by Blanz *et al.* [1999]

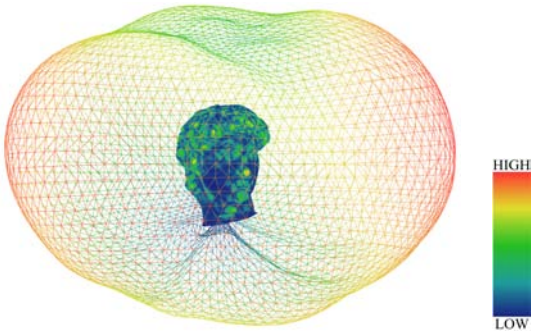


Figure 10: For viewpoint selection, we find the viewpoint that maximizes the visible saliency sum. Here, the wireframe mesh around the David's head model shows the magnitude of the visible saliency sum when the model is seen from each direction. The color of the mesh is also mapped from the visible saliency sum. Our method selects the view-direction with the highest magnitude.

to automatically compute initial viewpoints for 3D objects. Systems such as these can greatly benefit from a computational model of mesh saliency.

We have developed a method for automatically selecting viewpoint so as to visualize the most salient object features. Our method selects the viewpoint that maximizes the sum of the saliency for visible regions of the object. For a given viewpoint v, let $F(v)$ be the set of surface points visible from v, and let $\mathscr{S}$ be the mesh saliency. We compute the saliency visible from v as: $U(v) = \sum_{x \in F(v)} \mathscr{S}(x)$. Then the viewpoint with maximum visible saliency v_m is defined as $v_m = \underset{v}{argmax}\ U(v)$. One possible solution here is to exhaustively compute the maximum visible saliency over all viewpoints. This is shown in Figure 10. This, however, could get computationally intensive as the amount and complexity of 3D content rises.

Instead, we use a gradient-descent-based optimization heuristic to help us select good viewpoints. The optimization variables are the longitude and latitude, (θ, ϕ) and the objective function is the visible saliency $U(\theta, \phi)$. We start from a random view direction and use the iterative gradient-descent method to find the local maxima. We compute the local gradient by probing the saliency at neighboring view points. We use a randomized algorithm to find the global maximum by repeating this procedure with multiple randomly selected starting points. We can see the results of this approach for Stanford's David model in Figure 11. It is interesting to see that our approach identified a side of the face whereas a purely curvature-based approach has identified a view looking straight down at the back of David's head.

6 Results and Discussion

We have developed a model for mesh saliency, discussed its computation, and shown its applicability to mesh simplification and viewpoint selection. Figure 6 shows the mesh saliency for the Cyberware Dinosaur and the Cyberware Isis models. Repeating patterns are usually not classified as salient by our approach. Notice that although the curvature of the Dinosaur's ribs in Figure 6 is high, their saliency is low. For other examples, consider the repeated bumps on the legs of the Armadillo model in Figure 7, David's hair in Figure 11, or patterns in Isis's wig in Figure 6. Our approach assigns a low saliency to such local repeating patterns.

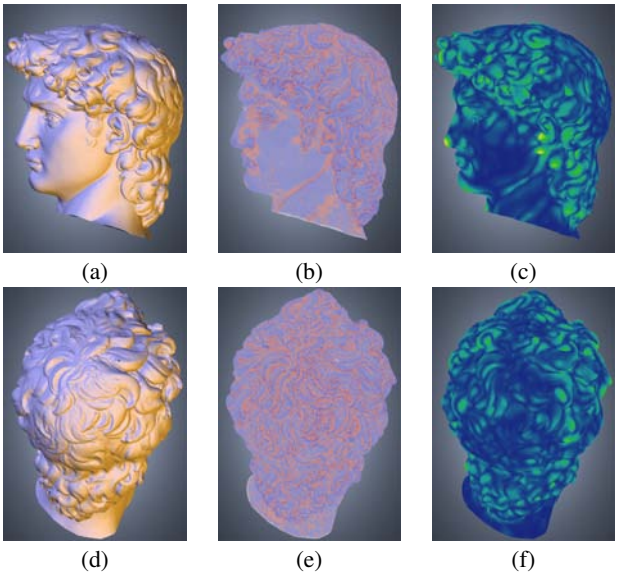


Figure 11: Image (a) shows a viewpoint selected by maximizing visible saliency, and image (d) shows a viewpoint selected by maximizing visible mean curvature. Images (b) and (e) show the mean curvature for the two selected viewpoints, and images (c) and (f) show the saliency. Since saliency negates the repeated hair texture in image (e), the method based on saliency selects the more interesting region of face instead of the top of the head.

The application of our saliency models to guide simplification of meshes have also given us very effective results. Consider for instance the Cyberware Male in Figure 9. Notice how our saliency-based simplification retains more triangles around the ears, nose, lips, and eyes than previous methods. Although in this case, salient simplification preserves the desirable high curvature regions, it can also selectively ignore the undesirable high curvature regions, such as in the simplification of the Armadillo's legs (Figure 7) or in ignoring David's hair for viewpoint selection (Figure 11).

The time to compute saliency depends on the scale at which it is computed. Larger scales require identification and processing of a larger number of neighborhood vertices and therefore are more time consuming. Spatial data-structures such as a grid or an octree can greatly improve the running time for establishing the neighborhood at a given scale. Table 1 shows the time for saliency computation on a 3.0 GHz Pentium IV PC with 2 GB RAM using a regular grid.

Table 1: Run Times for Computing Mesh Saliency

		Time for each scale (sec)				
Model	**#verts**	2ε	3ε	4ε	5ε	6ε
Dinosaur	56K	1.6	3.4	4.8	6.7	9.0
Armadillo	172K	7.6	15.4	20.5	29.8	41.1
Male	303K	20.7	35.2	50.6	71.2	95.2
Dragon	437K	34.8	72.8	93.8	131.9	178.9
David's Head	2M	593.7	1097.2	1407.4	1968.6	2619.7

Our mesh saliency computation approach is based on a center-surround operator, which is present in many models of human vision. We use this approach primarily because it is a straightforward way of finding regions that are unique relative to their surroundings. For this reason, it is plausible that mesh saliency may capture the regions of 3D models that humans will also find salient. Our experiments provide preliminary indications that this may be true.

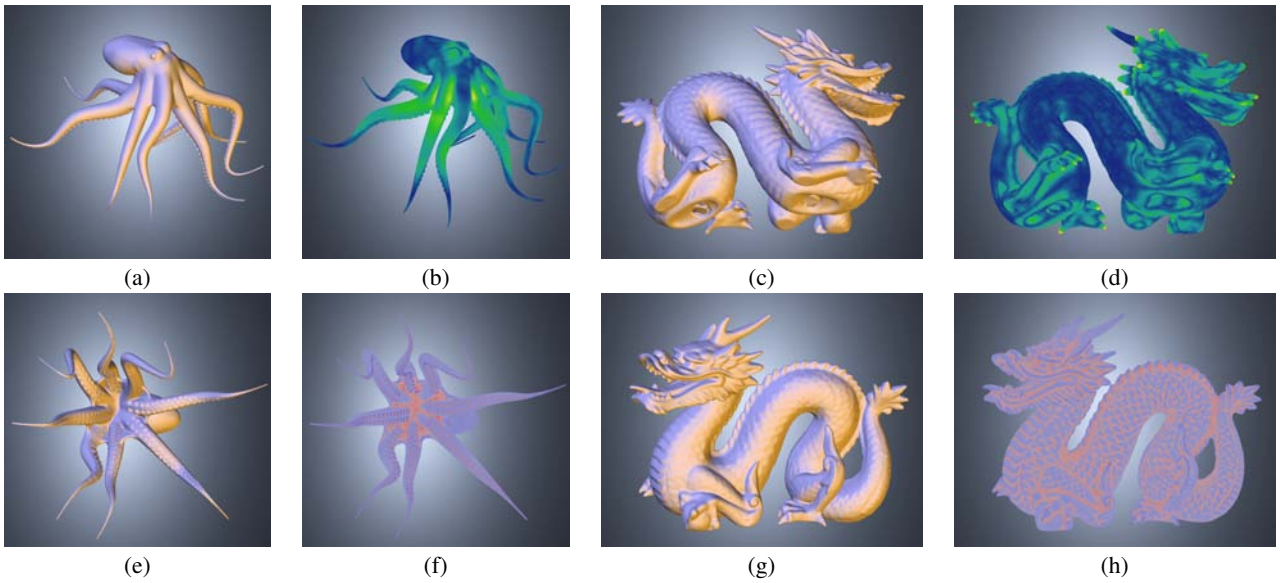


Figure 12: Viewpoint selection for the Octopus and the Stanford Dragon models. Images (a)–(d) show viewpoints selected by maximizing visible saliency, and images (e)–(h) show viewpoints selected by maximizing visible mean curvature. Images (b) and (d) show the saliency, and images (f) and (h) show the mean curvature. Compared with a curvature-based viewpoint selection method, the saliency-based method picks a more pleasing view for models with repeated textures such as (a) the octopus but not for (c) the Dragon. Our method for saliency-guided view selection for the Dragon selects the view from below instead of from the side since the Dragon's feet have a very high saliency.

7 Conclusions and Future Work

We have developed a model of mesh saliency using center-surround filters with Gaussian-weighted curvatures. We have shown how incorporating mesh saliency can visually enhance the results of several graphics tasks such as mesh simplification and viewpoint selection. For a number of examples we have shown in this paper, one can see that our model of saliency is able to capture what most of us would classify as interesting regions in meshes. Not all such regions necessarily have high curvature. While we do not claim that our saliency measure is superior to mesh curvature in all respects, we believe that mesh saliency is a good start in merging perceptual criteria inspired by low-level human visual system cues with mathematical measures based on discrete differential geometry for graphics meshes.

Mesh saliency promises to be a rich area for further research. We are currently defining mesh saliency using mean curvature. It should be possible to improve this by using better measures of shape, such as principal curvatures. Our current definition of mesh saliency considers only geometry. Generalizing mesh saliency to encompass other appearance attributes such as color, texture, and reflectance, should be an important direction for further research. Current methods for lighting design [Lee et al. 2004] do not incorporate any notion of perceptual saliency in deciding how and where to illuminate a scene. Saliency-based lighting design is likely to emerge as an important area for further research. Our current method of computing saliency takes a long time. It should be possible to significantly speed it up by using a multiresolution mesh hierarchy to accelerate filtering at coarser scales. Mesh segmentation [Katz and Tal 2003], like mesh simplification, is another mesh processing operation that could benefit from a saliency map that assigns different priorities to different regions of a mesh. It will also be an interesting exercise to use eye-tracking to determine the regions on 3D objects that elicit greater visual attention and contrast this with their computed saliency using methods such as ours.

8 Acknowledgements

We would like to thank the anonymous reviewers for their exceptionally thorough and careful reviews, voluminous advice, and valuable suggestions. We greatly appreciate their time and effort in helping make this paper significantly better. We will to thank Youngmin Kim for his help at various stages of this project. We would also like to acknowledge Stanford Graphics Lab and Cyberware Inc. for providing the models for generating the images in this paper. This work has been supported in part by the NSF grants: IIS 00-81847, ITR 03-25867, CCF 04-29753, and CNS 04-03313.

References

AL-REGIB, G., ALTUNBASAK, Y., AND ROSSIGNAC, J. 2005. Error-resilient transmission of 3D models. *ACM Transactions on Graphics 24*, 2, 182–208.

BLANZ, V., TARR, M. J., AND BÜLTHOFF, H. H. 1999. What object attributes determine canonical views? *Perception 28*, 5, 575–599.

CHEN, L., XIE, X., FAN, X., MA, W., ZHANG, H., AND ZHOU, H. 2003. A visual attention model for adapting images on small displays. *ACM Multimedia Systems Journal 9*, 4, 353–364.

DECARLO, D., AND SANTELLA, A. 2002. Stylization and abstraction of photographs. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2002) 21*, 3, 769–776.

FLEISHMAN, S., COHEN-OR, D., AND LISCHINSKI, D. 1999. Automatic camera placement for image-based modeling. In *Proceedings of the 7th Pacific Conference on Computer Graphics and Applications (PG 1999)*, 12–20.

FRINTROP, S., NÜCHTER, A., AND SURMANN, H. 2004. Visual attention for object recognition in spatial 3D data. In *2nd International Workshop on Attention and Performance in Computational Vision (WAPCV 2004)*, 75–82.

GARLAND, M., AND HECKBERT, P. 1997. Surface simplification using quadric error metrics. In *Proceedings of ACM SIGGRAPH*, 209–216.

GOOCH, B., REINHARD, E., MOULDING, C., AND SHIRLEY, P. 2001. Artistic composition for image creation. In *Proceedings of Eurographics Workshop on Rendering Techniques*, 83–88.

GUY, G., AND MEDIONI, G. 1996. Inferring global perceptual contours from local features. *International Journal of Computer Vision 20*, 1–2, 113–133.

HECKBERT, P. S., AND GARLAND, M. 1999. Optimal triangulation and quadric-based surface simplification. *Computational Geometry 14*, 49–65.

HISADA, M., BELYAEV, A. G., AND KUNII, T. L. 2002. A skeleton-based approach for detection of perceptually salient features on polygonal surfaces. *Computer Graphics Forum 21*, 4, 689–700.

HOPPE, H. 1996. Progressive meshes. *Proceedings of ACM SIGGRAPH*, 99–108.

HOWLETT, S., HAMILL, J., AND O'SULLIVAN, C. 2004. An experimental approach to predicting saliency for simplified polygonal models. In *Proceedings of the 1st Symposium on Applied Perception in Graphics and Visualization*, 57–64.

ITTI, L., KOCH, C., AND NIEBUR, E. 1998. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 11, 1254–1259.

KAMADA, T., AND KAWAI, S. 1988. A simple method for computing general position in displaying three-dimensional objects. *Computer Vision, Graphics, Image Processing 41*, 1, 43–56.

KARNI, Z., AND GOTSMAN, C. 2000. Spectral compression of mesh geometry. In *Proceedings of ACM SIGGRAPH*, 279–286.

KATZ, S., AND TAL, A. 2003. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH 2003) 22*, 3.

KIM, S.-J., KIM, S.-K., AND KIM, C.-H. 2002. Discrete differential error metric for surface simplification. In *Proceedings of 10th Pacific Conference on Computer Graphics and Applications (PG 2002)*, 276 – 283.

KOBBELT, L., CAMPAGNA, S., VORSATZ, J., AND SEIDEL, H.-P. 1998. Interactive multi-resolution modeling on arbitrary meshes. In *Proceedings of ACM SIGGRAPH*, 105–114.

KOCH, C., AND ULLMAN, S. 1985. Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology 4*, 219–227.

LEE, C. H., HAO, X., AND VARSHNEY, A. 2004. Light collages: Lighting design for effective visualization. In *Proceedings of IEEE Visualization*, 281–288.

LUEBKE, D., AND HALLEN, B. 2001. Perceptually driven simplification for interactive rendering. In *Proceedings of Eurographics Workshop on Rendering Techniques*, 223 – 234.

LUEBKE, D., REDDY, M., COHEN, J., VARSHNEY, A., WATSON, B., AND HUEBNER, R. 2003. *Level of Detail for 3D Graphics*. Morgan Kaufman.

MANTIUK, R., MYSZKOWSKI, K., AND PATTANAIK, S. 2003. Attention guided MPEG compression for computer animations. In *Proceedings of the 19th Spring Conference on Computer Graphics*, 239–244.

MEDIONI, G., AND GUY, G. 1997. Inference of surfaces, curves and junctions from sparse, noisy 3-D data. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 11, 1265–1277.

MEYER, M., DESBRUN, M., SCHRÖDER, P., AND BARR, A. H. 2003. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III (Proceedings of VisMath 2002)*, Springer Verlag, Berlin (Germany), 35–54.

MILANESE, R., WECHSLER, H., GIL, S., BOST, J., AND PUN, T. 1994. Integration of bottom-up and top-down cues for visual attention using non-linear relaxation. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, 781–785.

PRIVITERA, C., AND STARK, L. 1999. Focused JPEG encoding based upon automatic preidentified regions of interest. In *Proceedings of SPIE, Human Vision and Electronic Imaging IV*, 552–558.

REDDY., M. 2001. Perceptually optimized 3D graphics. *IEEE Computer Graphics and Applications 21*, 5, 68–75.

ROSENHOLTZ, R. 1999. A simple saliency model predicts a number of motion popout phenomena. *Vision Research 39*, 19, 3157–3163.

SHASHUA, A., AND ULLMAN, S. 1988. Structural saliency: The detection of globally salient structures using a locally connected network. In *Proceedings of IEEE International Conference on Computer Vision*, 321–327.

STOEV, S., AND STRAßER, W. 2002. A case study on automatic camera placement and motion for visualizing historical data. In *Proceedings of IEEE Visualization*, 545–548.

SUH, B., LING, H., BEDERSON, B. B., AND JACOBS, D. W. 2003. Automatic thumbnail cropping and its effectiveness. *CHI Letters (UIST 2003) 5*, 2, 95–104.

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

TSOTSOS, J., CULHANE, S., WAI, W., LAI, Y., DAVIS, N., AND NUFLO, F. 1995. Modeling visual-attention via selective tuning. *Artificial Intelligence 78*, 1-2, 507–545.

TURK, G. 1992. Re-tiling polygon surfaces. In *Proceedings of ACM SIGGRAPH*, 55–64.

VÁZQUEZ, P.-P., FEIXAS, M., SBERT, M., AND LLOBET, A. 2002. Viewpoint entropy: a new tool for obtaining good views of molecules. In *Proceedings of the Symposium on Data Visualisation (VISSYM 2002)*, 183–188.

WATANABE, K., AND BELYAEV, A. G. 2001. Detection of salient curvature features on polygonal surfaces. *Computer Graphics Forum (Eurographics 2001) 20*, 3, 385–392.

WATSON, B., WALKER, N., AND HODGES, L. F. 2004. Suprathreshold control of peripheral LOD. *ACM Transactions on Graphics 23*, 3, 750–759.

WEINSHALL, D., AND WERMAN, M. 1997. On view likelihood and stability. *IEEE Transactions on Pattern Analysis and Machine Intelligence 19*, 2, 97–108.

YEE, H., PATTANAIK, S., AND GREENBERG, D. P. 2001. Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. *ACM Transactions on Graphics 20*, 1, 39–65.