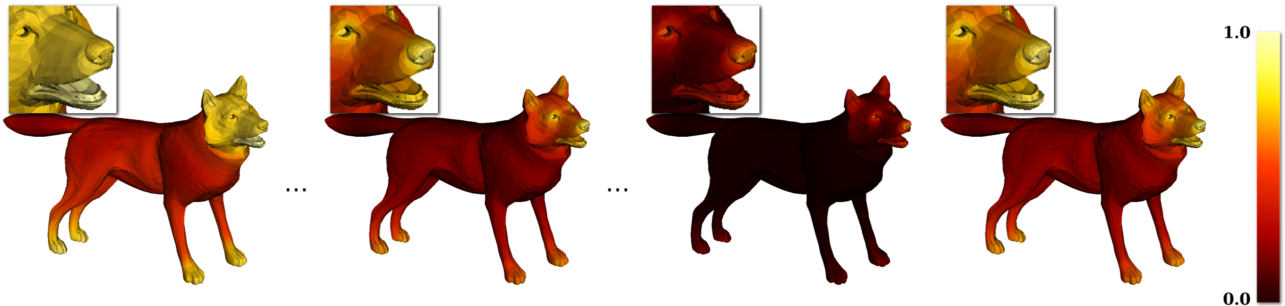
EUROGRAPHICS 2016/ T. Bashford-Rogers and L. P. Santos *Short Paper*

Mesh Saliency Analysis via Local Curvature Entropy

M. Limper[1,2], A. Kuijper[1,2] and D. W. Fellner[1,2,3]

[1] TU Darmstadt [2] Fraunhofer IGD [3] TU Graz



(a) l_0 (b) l_2 (c) l_4 (d) $l_0 ... l_4$ combined

Figure 1: *Multi-scale saliency estimation. We can efficiently captures different kinds of salient features, which are then combined into a single saliency map. Our novel approach is more than an order of magnitude faster than other methods, flexible, effective, and easy to implement.*

Abstract

We present a novel approach for estimating mesh saliency. Our method is fast, flexible, and easy to implement. By applying the well-known concept of Shannon entropy to 3D mesh data, we obtain an efficient method to determine mesh saliency. Comparing our method to the most recent, state-of-the-art approach, we show that results of at least similar quality can be achieved within a fraction of the original computation time. We present saliency-guided mesh simplification as a possible application.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems

1. Introduction

The analysis of *Mesh Saliency* aims at detecting *distinctive* regions of a 3D mesh. Applications include saliency-aware simplification, which strives to preserve the most important features, automated best viewpoint selection, detection of interest points, and 3D scan registration, to name a few [LVJ05,FSG09,DCG12,SLMR14]. One open challenge is the implementation complexity of current methods, which impedes their integration into existing mesh processing pipelines. In addition, long computation times may also limit the applicability of saliency detection in practice.

We present a novel approach towards mesh saliency detection, based on local curvature entropy. Being based on information theory, our method classifies regions on a 3D surface according to the amount of information they contain. Compared to the most recent approach from the field, our method is able to deliver results of similar quality, while being over one order of magnitude faster.

2. Methods for Mesh Saliency Analysis

Within this section, we briefly summarize the most important work related to our saliency detection method for 3D meshes. For a more detailed survey of the state of the art, the interested reader is referred to the recent work of Song et al. [SLMR14].

The seminal paper of Itti et al. introduced a method to compute saliency on images [IKN98]. In this context, saliency is defined as a measure of how strongly a part of an image will draw a human's immediate attention. The method incorporates knowledge about the human visual system, employing a 2D neural network to simulate the possible movement of a human's focus of attention. Lee et al. have transferred the basic concept to the domain of 3D surface meshes [LVJ05]. One of their key findings is that, within this domain, curvature is the main source of information. Saliency is computed at multiple scales from filtered mean curvature values, using a Difference-of-Gaussians (DoG) method. The authors pro-

DOI: 10.2312/egsh.20161003

pose a non-linear weighting scheme for the different levels, such that maps with a few distinctive, high peaks are than promoted over ones with many similar peaks. Demonstrated applications are mesh simplification and best viewpoint selection. Like Lee et al., we base our method on mean curvature, to measure, at each vertex, deviations from a local neighborhood. However, our information-theoretic definition works well even on a single scale. Moreover, it also prevents us from having to explicitly suppress frequently occurring values inside the saliency maps.

Page et al. introduced the concept of *Shape Information* [PKS*03]. The aim of the method is to measure how much information is contained within a 3D surface mesh. The authors reason that Shannon entropy of discrete surface curvature values is well-suited for this purpose. Therefore, the method is conceptually related to our saliency measure. However, we do not use a single value to classify the whole surface. Instead, our method is able to compute per-vertex saliency scores, and it is able to operate at multiple scales. Leifmann et al. use *Spin Images* to measure how different a vertex is from his neighborhood [LST12]. The main application is best viewpoint selection. Our method is similar in the sense that it also uses a local descriptor at each vertex, to measure local distinctiveness. However, while spin images are more complex than our measure, they fail to detect salient features in some cases (cf. [SLMR14]). Feixas at el. compute saliency by considering the information channel between a set of viewpoints and the polygons of the mesh [FSG09]. The approach is therefore primarily designed for best viewpoint selection, which is demonstrated as main application. In contrast, our goal is to use mesh saliency as a geometric tool for a wider variety of tasks, therefore we generally prefer a view-independent measure.

Recently, Song et al. presented a spectral method for saliency analysis [SLMR14]. Saliency is determined via deviations from a local average in the frequency domain, using the log-Laplacian spectrum. The method is shown to detect most of the typically salient features on several test meshes. As applications, the authors present saliency-driven mesh simplification and registration of 3D scans. Due to the spectral decomposition, the computational costs of the approach limit its direct application to very small meshes. However, the authors show that, for many cases, satisfying results can be obtained using simplified versions. Computed results are propagated back to the original, high-resolution meshes, using closest point search and a subsequent smoothing step.

3. Algorithm

3.1. Saliency via Local Curvature Entropy

Shannon entropy is defined as the expected value of the information within a message. More precisely, it allows to compute the number of bits $H(X)$ that are needed to encode a message X (not necessarily an integer number), without any loss of information. Assuming that X is composed using n different kinds of symbols with probabilities p_i, its entropy is defined as $H(X) = -\sum_i^n p_i \log_2(p_i)$.

Curvature has been found to be the main source of information on a 3D surface, since a shape can be locally characterized solely via its principal curvatures [PKS*03]. Therefore, we can estimate mesh saliency locally, using the mean curvature at each vertex v_i (cf. [LVJ05]). By considering curvature values within the geodesic r-neighborhood $\varphi = \{w_0, ..., w_m\}$ of v_i a discrete message, we are able to compute the local curvature entropy (LCE). This measure can directly be regarded as the local amount of information at the given vertex. Like Page et al., we obtain a discrete range of possible symbols from the continuous curvature values using a uniform sampling into a fixed number of n bins, resulting in a discrete range of possible symbols $\sigma_0, ..., \sigma_n$ [PKS*03]. The local curvature entropy at vertex v_i can then be computed as

$$H(v_i) = -\sum_j^n p_\varphi(\sigma_j)\, \log_2(p_\varphi(\sigma_j)),$$

using the local probabilities $p_\varphi(\sigma_i)$ of each symbol.

3.2. Area Weighting

Symbol probabilities could simply be computed with respect to the amount of vertices within the local neighborhood. However, if the triangle distribution is irregular, large, possibly stretched triangles might heavily influence distant neighbors, for example. Since such effects are not desired, we mitigate the problem to a certain amount by weighting vertices by their area of influence, using *mixed voronoi cells*, providing a complete, overlap-free tiling of the overall mesh surface (see [BKP*10], for example). Using the area weights $A_0, ..., A_m$ of each vertex, the probability of a given symbol within the local neighborhood φ is then defined as

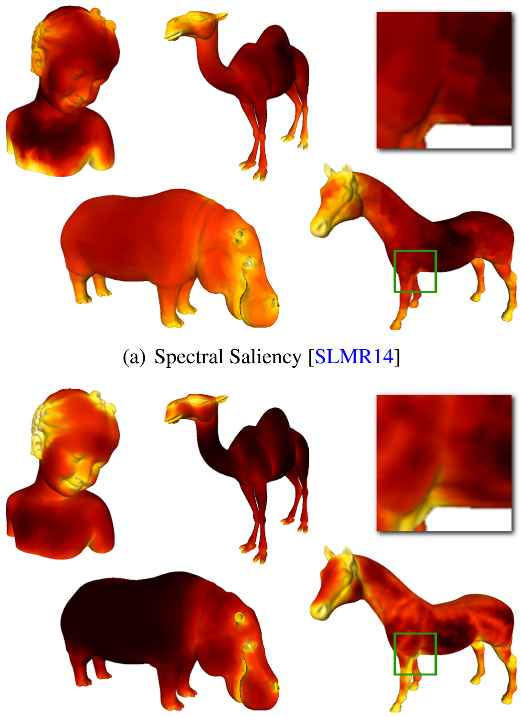
$$\begin{aligned} p_\varphi(\sigma) &= \frac{\sum_k^m A_k\, \chi_k(\sigma)}{\sum_k^m A_k}, \\ \chi_k(\sigma) &= \begin{cases} 1, & \text{if } \mathit{binnedCurvature}(w_k) = \sigma, \\ 0, & \text{otherwise} \end{cases} . \end{aligned}$$

3.3. Multi-Scale Saliency Detection

With the radius parameter r, we are able to control the size of features our algorithm will detect. However, it is usually desirable to detect features at multiple scales, so that neither small, distinctive details, nor larger, interesting regions are missed. To achieve this, we use a multi-scale detection by varying the radius parameter up to a fixed maximum r_{max}. Saliency maps are then computed at multiple levels $l_0, ..., l_{n-1}$, where the radius parameter for each level l_i is simply defined as $r_i = 2^{-i} r_{max}$.

4. Results

Throughout our experiments, we have used a maximum radius parameter of $r_{max} = 0.08\sqrt{A_M}$, where A_M is the surface area of the mesh, and we have computed saliency at five different levels. Choosing a largest possible feature size is necessary for all feature detection algorithms, and our particular choice for r_{max} matches well with the class of objects we used for our experiments (cf. Fig 1). The number of bins for discretizing curvature values was always set to 256. We have also found that using a uniform weighting scheme to combine these levels into a single, resulting saliency map provided satisfying results. Figure 1 shows an example. Clearly, using other weighting schemes is possible. In general, a different weighting scheme can be used to boost salient features on a certain scale, if desired.



(a) Spectral Saliency [SLMR14]

(b) LCE

Figure 2: *Results obtained using spectral saliency (top) and using our method (bottom).*

Figure 2 shows a visual comparison of the results for some test meshes, comparing our method to the spectral approach of Song et al. [SLMR14]. To visually evaluate the results more in depth, the interested reader is referred to the supplemental material of this paper, which contains the results for our test meshes in PLY format, with saliency values mapped to vertex colors using two different transfer functions (including the classical*rainbow* color map). Saliency values are also directly provided as a custom, scalar *quality* property.

As can be seen from Fig. 2, our method captures all important features that would typically be considered salient, such as feet or facial features. Results for the spectral method have been computed with the suggested setting of initial simplification to 3,000 faces per mesh. As can be seen from the inset within the figure, the upsampling of the results from the simplified mesh to the original one, using closest point correspondences, leads to patch-like structures, which are also shown and discussed by the authors [SLMR14]. Although we used the original implementation, including a smoothing step, some patch-like structures remain visible. In contrast, our method operates directly on the high-resolution mesh, hence the result is free of such artifacts.

4.1. Computation Times

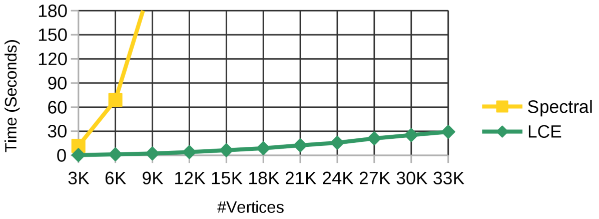


Figure 3: *Runtime performance, using different resolutions of the Stanford bunny. Our LCE method processes the original mesh within about 30 seconds.*

			[SLMR14]		LCE	
Model	#vert.	#Δ	3KΔ	5KΔ	3KΔ	orig. #Δ
Wolf	4,344	8,684	10.26	44.97	0.27	0.82
Woman	5,641	11,278	9.20	37.37	0.21	1.38
Camel	9,757	19,510	10.72	43.84	0.36	3.95
Cup	15,070	30,140	9.73	46.81	0.28	5.40
Bimba	15,516	31,028	10.36	42.81	0.52	9.30
Hippo	23,132	46,202	11.19	44.77	0.53	31.21
M. Planck	27,726	55,448	11.23	44.90	0.22	16.47
Bunny	34,834	69,451	11.28	43.29	0.21	29.29
Horse	48,485	96,966	11.27	44.16	0.22	47.75

Table 1: *Computation times, in seconds, for spectral saliency [SLMR14] and for our method. For the spectral method, each model has been previously simplified to 3,000 and 5,000 triangles.*

To evaluate the runtime performance of our algorithm, we have tested it with several meshes from the SHREC 2007 watertight track, as well as with a few other popular test meshes. We have also compared the results with the ones generated using the spectral method. Throughout our experiments, a test machine with 3.4 GHz i7-3770 CPU and 32 GB RAM was used.

Figure 3 shows a comparison of runtime performance for both evaluated methods, using the Stanford bunny at different resolutions. The spectral method uses an eigenvalue decomposition of the mesh Laplacian, which is the dominant factor of the computation and has a runtime behavior of $\mathcal{O}(n^3)$. Therefore, it can become computationally too expensive if an input a mesh consists of more than a few thousand vertices. Because of this, the computation times of the spectral method, as shown in Table 1, have been measured using an initial simplification to 3,000 faces (recommended default setting), as well as to 5,000 faces. As can be seen, our method is more than fifty times faster when executed on the same, simplified meshes. Moreover, for test meshes of moderate size, it even remains more than one order of magnitude faster when operating on the original mesh.

4.2. Saliency-Aware Simplification

We have evaluated saliency-aware simplification as a possible application. Our implementation uses the well-established OpenMesh library [BSBK02]. In addition to the standard quadric module, we have used a custom saliency module, using a threshold percentile to protect the most salient regions of the mesh. When no more collapses are allowed, we iteratively relax the saliency threshold until the process converges to the desired number of vertices.

Similar to Song et al., we have evaluated the geometric root

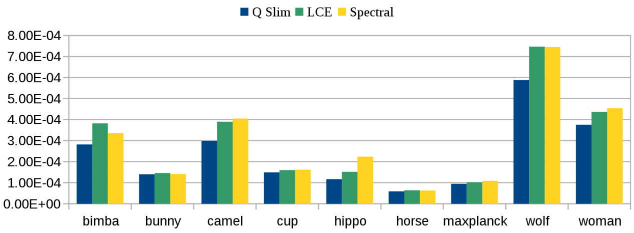
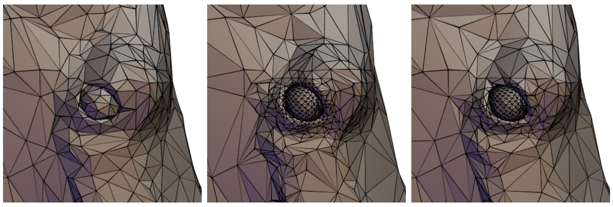


Figure 4: *Geometric error (RMSE) between the original meshes and 50% reduced versions. Saliency-aware simplification trades in geometric precision for better preservation of important features.*



(a) [GH97]
RMSE: $1.17 \cdot 10^3$

(b) [SLMR14]
RMSE: $2.24 \cdot 10^3$

(c) LCE
RMSE: $1.52 \cdot 10^3$

Figure 5: *Detail view onto the eye of the hippo model, after simplification to 50% of the original vertices.*

mean square error (RMSE) introduced by our saliency-guided simplification, compared with the standard quadric-based approach [SLMR14, GH97]. For direct comparison, we have also included results for their spectral approach. For both saliency detection methods, the same simplification framework was used.

Figure 4 shows the results on our test data sets, using a reduction to 50% of the original vertices. As can be seen, both of the saliency-based methods perform mostly equally well. For some cases, our method performs slightly better. One reason for this is that we can use the vertex budget more efficiently in small, isolated salient regions, such as the eye of the hippo mesh, as shown in Fig. 5. The spectral approach uses values propagated from a simplified mesh, hence the classification is more blurry. This leads to an unnecessary high amount of vertices around small, isolated features. For the bimba mesh, however, using the spectral method resulted in a lower RMSE. This is due to the many fine, small-scale structures within the hair of the character, which are more dominant in the high-resolution mesh used by our algorithm. Preserving that many details consumes most of the vertex budget, therefore the overall geometric error of the mesh suffers more in this case. If desired, this could be changed using a different weighting scheme, boosting large-scale features (cp. Fig. 1). Another approach would be to smooth the saliency map. Similarly, since our method is applied to the unmodified mesh, small geometric changes, such as noise, will directly impact the result. However, we have found that a smoothing step helps to overcome this sensitivity.

5. Conclusion

We have presented a novel method for estimation of mesh saliency, entitled Local Curvature Entropy (LCE). Our approach computes per-vertex entropy scores from binned curvature values within a geodesic neighborhood. We perform the analysis at multiple scales and combine the results into a single saliency map.

We have compared our results to the most recent approach, which is based on spectral decomposition. During our experiments, we found that our method is able to significantly outperform this approach in terms of computation time (more than an order of magnitude). We have also compared the results of both methods in a saliency-aware simplification application, demonstrating that LCE provides competitive results. Specifically, operating on the original mesh allows us to detect arbitrarily small features, if desired.

There are several promising directions for future work. We have used a simple implementation of Dijkstra's algorithm for geodesic neighborhood search, which is currently dominating the overall computational effort ($>$ 90% of time). As more optimized methods exist, we expect a significant speedup by improving this part of our algorithm. Furthermore, the saliency computation of each vertex happens independently from the others, therefore a parallelized implementation also seems promising. Finally, we also would like to explore other possible applications of our method, such as detection of feature points for shape matching and registration.

Acknowledgements. Test meshes are courtesy of the Stanford Computer Graphics Laboratory and of the SHREC 2007 *Watertight* track [GBP07]. We would also like to thank the anonymous reviewers for their helpful comments and suggestions.

References

[BKP*10] Botsch M., Kobbelt L., Pauly M., Alliez P., Levy B.: *Polygon Mesh Processing*. AK Peters, 2010. 2

[BSBK02] Botsch M., Steinberg S., Bischoff S., Kobbelt L.: OpenMesh: A Generic and Efficient Polygon Mesh Data Structure. In *OpenSG Symposium 2002* (2002). 3

[DCG12] Dutagaci H., Cheung C., Godil A.: Evaluation of 3d interest point detection techniques via human-generated ground truth. *The Visual Computer 28*, 9 (2012), 901–917. 1

[FSG09] Feixas M., Sbert M., González F.: A unified information-theoretic framework for viewpoint selection and mesh saliency. *ACM Trans. Appl. Percept. 6*, 1 (Feb. 2009), 1:1–1:23. 1, 2

[GBP07] Giorgi D., Biasotti S., Paraboschi L.: Shape retrieval contest 2007: Watertight models track, 2007. 4

[GH97] Garland M., Heckbert P. S.: Surface simplification using quadric error metrics. In *Proc. SIGGRAPH* (1997), ACM Press/Addison-Wesley Publishing Co., pp. 209–216. 4

[IKN98] Itti L., Koch C., Niebur E.: A model of saliency-based visual attention for rapid scene analysis. *TPAMI 20*, 11 (Nov 1998), 1254–1259. 1

[LST12] Leifman G., Shtrom E., Tal A.: Surface regions of interest for viewpoint selection. In *Proc. CVPR* (June 2012), pp. 414–421. 2

[LVJ05] Lee C. H., Varshney A., Jacobs D. W.: Mesh saliency. In *Proc. SIGGRAPH* (2005), ACM, pp. 659–666. 1, 2

[PKS*03] Page D., Koschan A., Sukumar S., Roui-Abidi B., Abidi M.: Shape analysis algorithm based on information theory. In *Proc. ICIP* (Sept 2003), vol. 1, pp. I–229–32 vol.1. 2

[SLMR14] Song R., Liu Y., Martin R. R., Rosin P. L.: Mesh saliency via spectral processing. *ACM Trans. Graph. 33*, 1 (Feb. 2014), 6:1–6:17. 1, 2, 3, 4