Course Report on Mesh Saliency

Yupan Liu[*]

College of Computer Science and Technology
Zhejiang University

January 9, 2016

The report for the course *Studies and Discussions of Special Subjects: Mixed Reality*, and this course project based on the SIGGRAPH 2005 paper[2]. It devoted to demonstrating normal vector, mean curvature and mesh saliency of an image.

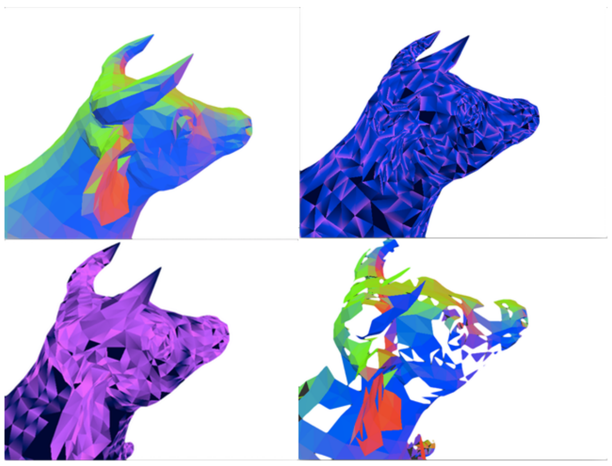

Figure 1: Normal vector, mean curvature and mesh saliency (also with simplication)

An outline of this report is as follows.

- First, we reviewed basic ideas from differential geometry, and then discuss its correspondent version in discrete case.
- Second, we use curvature to describe the mesh saliency, i.e. measure of regional importance for graphics meshes. It is defined in a scale-dependent manner using a center-surround operator on Gaussian-weighted mean curvatures.
- Third, we show some applications on mesh saliency

Now let's start from basic ideas in differential geometry.

1 Basics in Differential Geometry

Now we introduce (First & Second) fundamental form of surfaces and its *shape operator*. The latter is necessary to calculate the mesh saliency.

[*]yp_liu@zju.edu.cn

1.1 First & Second fundamental form

First fundamental form

$$\mathrm{d}s^2 = E\mathrm{d}x^2 + 2F\mathrm{d}x\mathrm{d}y + G\mathrm{d}y^2 = \begin{pmatrix} E & F \\ F & G \end{pmatrix}$$

$$\mathrm{d}A = \frac{1}{2}(EG - F)^2\mathrm{d}x\mathrm{d}y$$

$\mathrm{d}s^2$ is similar to the length of the curve and is part of an invariant quality system for the surface. Also $\mathrm{d}A$ describe properties of the area of this surface.

Second fundamental form

$$|(x + \mathrm{d}x, y + \mathrm{d}y) - (x, y)| = e^2\mathrm{d}x + 2f\mathrm{d}x\mathrm{d}y + g\mathrm{d}y^2 = \begin{pmatrix} e & f \\ f & g \end{pmatrix}$$

It is devoted to describing the shape of the surface and constructs the invariant quality system with the first fundamental form. Use these two fundamental forms, and we can decide two surfaces are identical or not.

1.2 Shape operator

From the second fundamental form, we got that *Shape operator*, a.k.a. *Weingarten map*, is a linear map to describe the property (curvatures) of the surface. Its eigenvalues are *principal curvature*, and their correspondent eigenvectors are their *principal directions*, respectively. We can use these things to calculate Gaussian curvature and mean curvature.

$$S(v) = -\mathrm{D}_v N$$
$$\langle S_x v, w\rangle = \langle \mathrm{d}f(v), w\rangle$$

Specifically,

$$S = (EG - F)^{-1}\begin{pmatrix} eG - fF & fG - gF \\ fE - eF & gE - fF \end{pmatrix}$$

Then we introduce that Gaussian curvature(intrinsic invariant) and mean curvature(extrinsic invariant) for surfaces.

Gaussian curvature

$$K = \frac{I}{II}$$

Mean curvature

$$K_m = \frac{eG + gE - 2fF}{EG - F^2}$$

2 Curvature in Discrete Differential Geometry

To calculate principal curvature on the discrete surface, i.e. eigenvalues of shape operator, we need to construct the shape operator for the discrete surface. The primary reference of this part is Taubin's paper[3].

Consider estimating principal curvatures and principal directions for triangulated surface with vertices V and faces F.

$$\begin{aligned}
&S = \{V, F\}\\
&V = \{v_i : 1 \leq i \leq n_V\}\\
&F = \{f_k : 1 \leq k \leq n_F, f_k = (i_1^k, i_2^k, i_3^k)\}
\end{aligned}$$

2.1 Notations of Vertices and Faces

Mapping v_i to share faces V^i where $v_j \in V^i$ is the neighborhood of v_i. And F_i is the set of faces contain v_i, such that f_k is incident to v_i.

For given vertices and faces, we define the norm vector of them as below. It is an average of vectors, with weights proportional to the surface areas of these faces

$$N_{v_i} = \frac{\sum_{f_k \in F^i} |f_k| N_{f_k}}{\| \sum_{f_k \in F^i} |f_k| N_{f_k} \|}.$$

2.2 Shape operator

For continuous surface, we have

$$M_p = \frac{1}{2\pi} \int_{-\pi}^{\pi} K_p(T_\theta) T_\theta T_\theta^\dagger \mathrm{d}\theta,$$

where $K_p(T_\theta) = \kappa_p^1 \cos^2\theta + \kappa_p^2 \sin^2\theta$ and $T_\theta = \cos\theta T_1 + \sin\theta T_2$. And for discrete surface, we have

$$\tilde{M_{v_i}} \approx \sum_{v_j \in V^i} w_{ij} \kappa_{ij} T_{ij} T_{ij}^t = \frac{1}{2\pi} \int_{-\pi}^{\pi} K_p(T_\theta) T_\theta T_\theta^\dagger \mathrm{d}\theta,$$

where unit normalized projection from v_j to v_i

$$T_{ij} = \frac{(I - N_{v_i} N_{v_i}^t)(v_i - v_j)}{\|(I - N_{v_i} N_{v_i}^t)(v_i - v_j)\|},$$

and directional curvature

$$\kappa_{v_i}(T_{ij}) = \frac{2 N_{v_i}^t (v_j - v_i)}{\|v_j - v_i\|^2}.$$

Since that

$$K_p(T) = u^t K_p u = \begin{pmatrix} n \\ t_1 \\ t_2 \end{pmatrix}^t \begin{pmatrix} 0 & 0 & 0 \\ 0 & \kappa_p^1 & 0 \\ 0 & 0 & \kappa_p^2 \end{pmatrix} \begin{pmatrix} n \\ t_1 \\ t_2 \end{pmatrix}.$$

We have $M_p = T_{12}^t \begin{pmatrix} m_p^{11} & m_p^{12} \\ m_p^{21} & m_p^{22} \end{pmatrix} T_{12}$ where $T_{12} = [T_1, T_2]$.

2.3 Simplify the shape operator by Household transform

Given $E_1 = (1,0,0)^t$ and $W_{v_i} = \frac{E_1 \pm N_{v_i}}{\|E_1 + N_{v_i}\|}$, so $W_{v_i} < 0$ means that $\|E_1 - N_{v_i}\| > \|E_1 + N_{v_i}\|$. Then we construct the Householder matrix $Q_{v_i} = I - 2w_{v_i} w_{v_i}^\dagger$ on the orthonormal basis $(\pm N_{v_i}, \tilde{T_1}, \tilde{T_2})^t$ of tangent space.

(1) Consider the eigenpair $N_{v_i} \sim 0$, we have

$$Q_{v_i}^t \tilde{M_{v_i}} Q_{v_i} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \tilde{m_{v_i}^{11}} & \tilde{m_{v_i}^{12}} \\ 0 & \tilde{m_{v_i}^{21}} & \tilde{m_{v_i}^{22}} \end{pmatrix}.$$

(2) Now we need to diagonalize the non-zero matrix mentioned above. We can solve it by rotation transform.

$$\begin{pmatrix} T_1 \\ T_2 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} \tilde{T_1} \\ \tilde{T_2} \end{pmatrix}$$

Now we reconstruct the matrix M_p by rotation angle θ,

$$M_p = \begin{pmatrix} m_p^{11} & m_p^{12} \\ m_p^{21} & m_p^{22} \end{pmatrix} = \int_{-\pi}^{\pi} \mathrm{d}\theta \frac{1}{2\pi} (\kappa_p^1 \cos^2\theta + \kappa_p^2 \sin^2\theta) \begin{pmatrix} \cos^2\theta & \cos\theta\sin\theta \\ \cos\theta\sin\theta & \sin^2\theta \end{pmatrix}.$$

Hence we got principal curvatures for the shape operator $K_p(T)$,

$$\begin{aligned} \kappa_p^1 &= 3m_p^{11} - m_p^{22} \\ \kappa_p^2 &= 3m_p^{22} - m_p^{11} \end{aligned}.$$

Finally, we got that all kinds of curvature which we need respectively,

- Principal curvature κ_p^1, κ_p^2,
- Gaussian curvature $\kappa_p^1 \kappa_p^2$,
- Mean curvature $\frac{1}{2}(\kappa_p^1 + \kappa_p^2)$.

where these matrix elements in M_p are

$$\begin{aligned} m_p^{12} &= T_1^t M_p T_2 = 0, \\ m_p^{11} &= T_1^t M_p T_1 = \frac{3}{8}\kappa_p^1 + \frac{1}{8}\kappa_p^2, \\ m_p^{22} &= T_2^t M_p T_2 = \frac{1}{8}\kappa_p^1 + \frac{3}{8}\kappa_p^2. \end{aligned}$$

2.4 Implementation

First, we implement this algorithm of calculating norm vectors on C++ and OpenGL 4[1], so the visualization of norm vectors for cow model is as follows.

[1] OpenGL 4's pipeline is different from order version, so we must use shader to render the object, such as the cow model in this report.

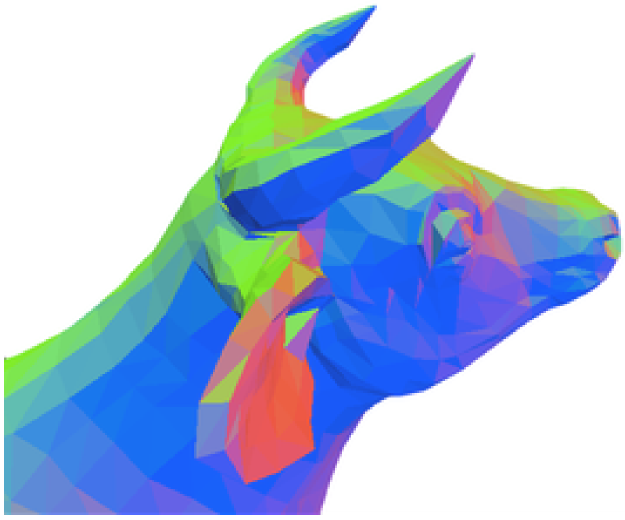

Figure 2: Norm vector for the model

Then, we implement the calculation of mean curvature by this method, and results are as follows. The brighter color of the vertex means larger mean curvature.

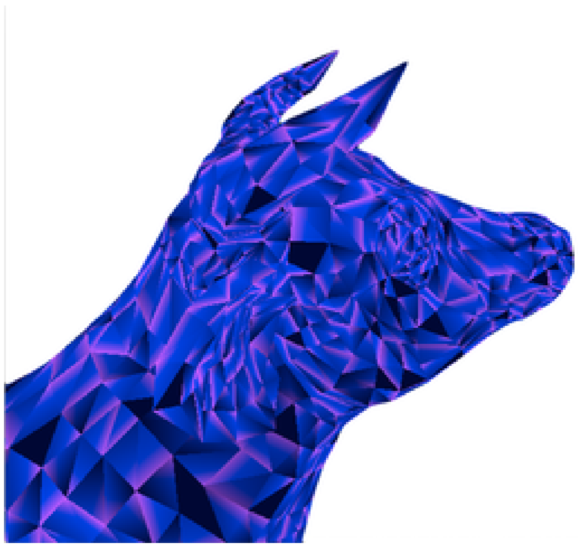

Figure 3: Mesh saliency for the model

3 Mesh Saliency

3.1 Surface curvature

Essentially, saliency[2] is a Gaussian-weighted center surround surface curvatures, i.e. only depend on the geometry of the model. For example, consider the sphere case, it is the canonical zero-saliency feature. Thus we want to use for describing the saliency by the curvature change.

As we mentioned about, from Taubin(1995)[3], we knew that how to calculate surface curvature (i.e. mean curvature). For any vertex v, we have a map l for mean curvature $l(v)$ on v such that $l : v \to l(v)$.

3.2 Gaussian filter

Consider the neighborhood $N(v,\sigma)$ described by the Euclidean distance $\{x| \|x-v\| < \sigma\}$ where x is a vertex on the mesh. We want to select these real salient vertices, so here we use the filter by Gaussian distribution to choose them.

Since that, we calculate the Gaussian-weighted average of the mean curvature

$$G(l(v),\sigma) = \frac{\sum_{x\in N(v,2\sigma)} l(x)\exp\{\frac{-\|x-v\|^2}{2\sigma^2}\}}{\sum_{x\in N(v,2\sigma)} \exp\{\frac{-\|x-v\|^2}{2\sigma^2}\}}.$$

3.3 Saliency computation

Now the saliency of vertex v is that

$$\psi(v) = |G(l(v),\sigma) - G(l(v),2\sigma)|.$$

And the multiple scales at a scale level i is that

$$\psi_i(v) = |G(l(v),\sigma_i) - G(l(v),2\sigma_i)|.$$

Here the σ_i is the standard deviation, in experience, $\sigma_i \in \{2\epsilon, 3\epsilon, 4\epsilon, \cdots, 6\epsilon\}$ where 0.3% l and l is the length of diagonal line of the cover box.

Then we consider the non-linear superssion to reduce the number of salient point, $\psi = \sum_i S(l_i)$. It is normalized, and $f_i = (M_i - \bar{m}_i)$ described by max saliency M_i and average saliency $\bar{m}_i$.

3.4 Implementation

From this algorithm, we also implement the calculation process of mesh saliency on C++ and OpenGL 4. The color of vertices is deeper, and it means that the mesh saliency on this vertex is larger. The results are as follows.

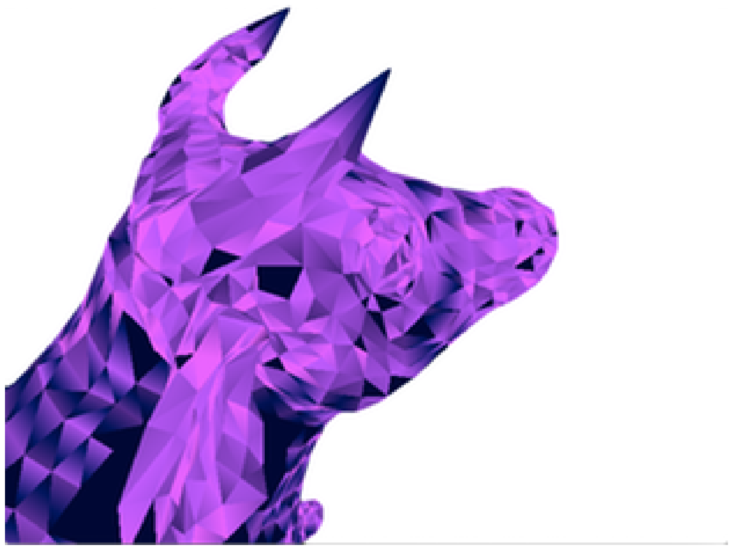

Figure 4: Mesh saliency for the model

4 Applications about Mesh Saliency

4.1 Salient simplification for quadrics-based simplification method

Now we introduce this mesh saliency on mesh simplification which based on Gerland & Heckbert[1]. The main idea of this method is to contract these minimum cost point iteratively, in the original paper it depends on the weight map W.

And we define a new weight map by mesh saliency, and use it to optimize the order of contracting vertices (and its correspondent weight). The saliency amplification operator A

$$\begin{aligned} W(v) &= A(\psi(v), \alpha, \lambda) \\ &= \begin{cases} \lambda\psi(v) & \text{if } \psi(v) \geq \alpha \\ \psi(v) & \text{if } \psi(v) < \alpha \end{cases} \end{aligned}$$

Here λ is the amplifying parameter and α is the threshold. We select $\lambda = 100$ and $\alpha = 30\%$. Hence the new simplification algorithm is that

(1) Initialzation

$$v : Q \leftarrow W(v)Q$$

(2) Merge the weight for each step

$$W(v) = W(v_i) + W(v_j)$$

4.2 Saliency viewpoint selection

In this part, we devoted to visualizing the most salient object features for automatically crafting best views, using gradient-descent based optimization method.

The objective function $U(\theta, \phi)$ where θ is the longitude and ϕ is the latitude. Technically, for the set of visible vertices $F(v)$ and mesh saliency ψ,

$$U(v) = \sum_{x \in F(v)} \psi(x)$$

$$v_m = \arg\max_v U(v)$$

Then select random view directions, and find the local maxima by applying gradient-descent method iteratively.

References

[1] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216. ACM Press/Addison-Wesley Publishing Co., 1997.

[2] Chang Ha Lee, Amitabh Varshney, and David W Jacobs. Mesh saliency. In *ACM transactions on graphics (TOG)*, volume 24, pages 659–666. ACM, 2005.

[3] Gabriel Taubin. Estimating the tensor of curvature of a surface from a polyhedral approximation. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 902–907. IEEE, 1995.

[4] Chen Weiheng. *Lectures in Differential Geometry (Chinese Edition)*. Beijing: Peking University Press, 2006.