

# 지오메트리 특징 강화를 위한 메시 필터링에 세일리언스 통합

이수빈\*<sup>○</sup>, 김종현\*\*

\*강남대학교 소프트웨어응용학부,

\*\*인하대학교 소프트웨어융합대학(디자인테크놀로지학과)

e-mail: [jonghyunkim@inha.ac.kr](mailto:jonghyunkim@inha.ac.kr)

## Incorporating Saliency into Mesh Filtering for Enhancing Geometry Features

SuBin Lee\*<sup>○</sup>, Jong-Hyun Kim\*\*

\*School of Software Application, Kangnam University,

\*\*College of Software and Convergence (Dept. of Design Technology), Inha University

### 요 약

본 논문에서는 메시 세일리언스(Mesh saliency)와 법선벡터를 이용해 3D 삼각형 메시의 형태를 안정적으로 과장하고 GPU 기반으로 새롭게 설계하는 프레임워크를 제안한다. 우리는 High-boost 메시 필터링 알고리즘에서의 계단현상(Aliasing) 문제를 메시 세일리언스를 통해 제거하고, GPU 기반에서 설계해 고속화한다.

▶ Keyword : High-boost mesh filtering, Aliasing, Mesh saliency, Mesh exaggeration, GPU optimization

## I. Introduction

High-boost mesh filtering은 이미지 처리 과정에서 사용되는 샤프닝(Sharpening) 알고리즘을 3D 메시에 적용할 때 삼각형의 뒤틀림 문제를 방지하는 방법이다. 메시의 해상도를 증가시키지 않고 표면 텍스처를 향상시켜 시각적으로 품질을 높인다. 그러나 메시의 불규칙성과 특정 영역에서 계단현상은 여전히 한계로 남아있다. 라플라시안 평활화와 같은 후처리로 이 문제를 억제할 수 있지만, 근본적인 문제를 해결하기 위해서는 High-boost mesh filtering 자체에서 이를 방지하는 기능이 추가적으로 필요하다.

High-boost mesh filtering에서 표면 텍스처 강화를 위한 새로운 법선 벡터를 계산할 때 사용하는 면적 가중 평균(Area-weighted average)은 계단현상을 유발하는 주요 원인 중 하나이다. 본 논문에서는 메시 세일리언스를 기반으로 계산된 세일리언스 방향

벡터를 High-boost mesh filtering의 새로운 법선 벡터 계산 과정에 통합함으로써, 지역적 중요도가 높은 영역의 곡률을 유지하고 메시의 특징을 보존하며 계단현상 문제를 완화시킬 수 있는 방법을 제안한다.

## II. Preliminaries

### 1. High-boost mesh filtering

High-boost mesh filtering 알고리즘은 Fig. 1에서 제시된 법선 벡터 기반의 오차 최소화 과정을 통해 각 정점의 새로운 최적 위치를 계산한다.

#### 1.1 boosted normal vector

법선 기반 오차 최소화에 사용되는 데이터는 삼각형의 법선 벡터  $n$ 과 법선 벡터를 강화한 boosted normal  $m$ 이다. 강화된 법선 벡터  $m(T)$ 는 다음과 같이 정의된다.

$$m(T) = \frac{(1+\alpha)n(T) - \alpha n^{(k)}(T)}{\|(1+\alpha)n(T) - \alpha n^{(k)}(T)\|} \quad (1)$$

$$n^{(k)}(T) = \frac{\sum_{i \in N(T)} A(U_i) n(U_i)}{\sum A(U_i)} \quad (2)$$

여기서  $n(T)$ 는 이웃 삼각형의 법선 벡터를  $k$ 번 반복 평활화하여 생성된 smoothed normal  $n^{(k)}(T)$ 로 가중치  $\alpha$ 에 따라 반영된다. Smoothed normal  $n^{(k)}(T)$ 는 면적 가중 평균으로 계산된다. 면적이 큰 삼각형일수록 평균을 계산할 때 더 큰 영향을 미친다.

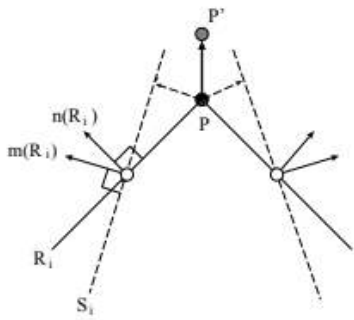


Fig. 1. Normal-based error minimization.

## 2. Mesh saliency

메쉬 세일리언시는 인간이 실제로 받아들이는 시각적 중요도를 고려하기 위해 가우시안 가중치를 이용한 곡률 평균으로 세일리언시를 계산하여 3D 메쉬에서 지역적 중요도를 나타내는 방법이다.

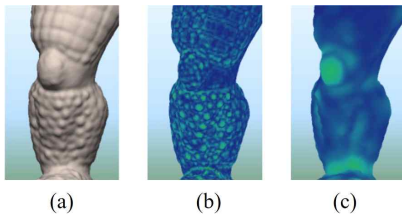


Fig. 2. Sample output from [2]. (a) part of the right leg of the Armadillo model, (b) magnitude of mean curvatures, (c) mesh saliency values.

단순한 곡률 기반 계산은 지역적 중요도를 잘 나타내지 못한다. Fig. 2a는 Armadillo의 오른쪽 다리는 돌기들의 곡률이 크더라도 같은 패턴이 반복되어 실제로는 시각적으로 단조롭다고 느껴지는, 중요도가 높지 않은 부분이다. 곡률 기반 계산은 Fig. 2b와 같이 곡률이 있는 돌기 모두 인식하지만, 메쉬 세일리언시를 계산한 결과인 Fig. 2c는 지역적 중요도가 높은 Armadillo의 무릎을 인식하는 결과를 도출했다.

이처럼 메쉬 세일리언시는 곡률이 크지만 지역적 중요도가 높지 않은 영역을 구분할 수 있다. 메쉬 세일리언시를 계산하는 과정은 다음과 같다. 먼저, 메쉬의 곡률을 계산하고 각 정점에 대해 평균 곡률  $C(v)$ 를 계산한다 (수식 3 참조).

$$S_i(v) = |G(C(v), \sigma_i) - G(C(v), 2\sigma_i)| \quad (3)$$

가중 평균 곡률을 계산한 후 다양한 크기나 범위에 서의 특징을 강조하기 위해 다중 스케일에서 세일리언시를 계산하고 비선형적으로 정규화한 후 통합한다.  $\sigma_i$ 는  $i$ 번째 스케일에서의 가우시안 필터의 표준 편차이다.  $M_i$ 는 스케일  $i$ 에서의 최대 세일리언시,  $\overline{m_i}$  지역 최댓값들의 평균을 나타낸다 (수식 4 참조).

$$S(S_i) = S_i \times (M_i - \overline{m_i})^2, \quad S = \sum_i S(S_i) \quad (4)$$

본 논문에서는  $\sigma_i \in \{2\epsilon, 3\epsilon, 4\epsilon, 5\epsilon, 6\epsilon\}$ 로 5개의 스케일과 메쉬 경계의 경계박스 대각선 길이의 0.5%를  $\epsilon$ 로 설정하여 계산한 메쉬 세일리언시를 사용한다.

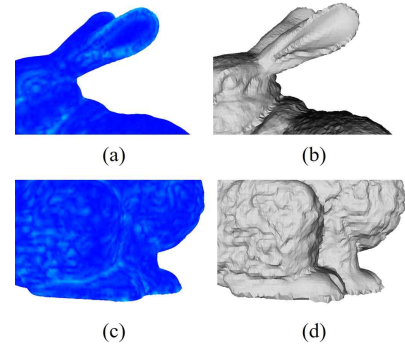


Fig. 3. Bunny model (a, c) saliency value of left ear and right foot, (b, d) high-boost mesh filtering on Bunny.

## III. The Proposed Scheme

면적 가중 평균 연산은 면적만을 가중치로 고려해 평균을 내는 이산적인 샘플링으로 인해 연속적인 곡면을 완벽하게 표현하지 못한다. 복잡한 곡면이나 날카로운 모서리가 있는 영역과 같이 곡률의 변화가 급격한 부분에서 이 문제가 두드러지며 계단현상 문제가 발생할 가능성이 높아진다.

Fig. 3a와 3c는 Bunny 모델에서 곡률의 변화가 급격한 부분에 해당하며 high-boost mesh filtering을 적용하자 계단현상이 발생한다. 메쉬 세일리언시를 이용하면 Bunny의 해당 영역이 지역적 중요도가 높은 영역임을 파악하고 주변 영역과 쉽게 구분할 수

있다. 본 논문에서는 메쉬 세일리언시가 높은 영역에 대해 세일리언시를 기반으로 한 새로운 필터링 방향을 계산한다. 이를 smoothed normal을 계산하는 면적 가중 평균 연산에 통합하여 계단현상 문제를 완화하고자 한다.

### 3.1 Saliency direction

먼저, 각 정점에서 Saliency 값의 변화량 (Gradient)이 최대인 방향을 계산한다.  $N_v$ 는 정점  $v$ 의 이웃 정점들의 집합,  $p$ 는 정점의 위치,  $s$ 는 정점의 세일리언시이다. 계산된 기울기를 각 정점의 세일리언시 방향 벡터로 설정한다. 이는 주변에서 세일리언시가 큰 쪽을 향하도록 계산된다 (수식 5 참조).

$$g_v = \sum_{i \in N_v} \frac{p_i - p_v}{|p_i - p_v|} (s_i - s_v) \quad (5)$$

각 삼각형을 구성하는 세 정점의 세일리언시 방향 벡터의 평균을 삼각형의 세일리언시 방향 벡터로 설정한다. 삼각형이 존재하는 평면에 대해 법선 벡터  $n$ 을 기준으로 세일리언시 방향 벡터를 반사하여 최종적으로 삼각형의 방향성 세일리언시 방향  $D_f$ 를 결정한다 (수식 6 참조).

$$D_f = -D_f + 2(D_f \cdot n)n \quad (6)$$

최종적으로 계산한 Saliency Direction는 Fig. 4와 같이 곡률이 급격한 부분에서 스무딩을 해도 자신의 곡률을 유지하도록 기여한다.

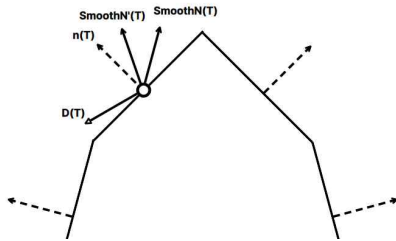


Fig. 4. Saliency direction's role in maintaining mesh curvature.

### 3.2 Saliency Direction Incorporating

세일리언지 방향이 법선 벡터의 보조 역할을 하도록 면적 가중 평균 연산에 다음과 같이 통합한다 (수식 7 참조).

$$\text{if } s_T > s_\lambda, \quad (7)$$

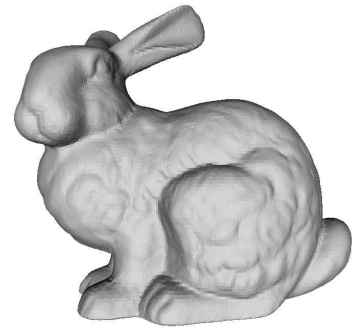
$$n^{(k)}(T) = \frac{\sum_{i \in N(T)} A(U_i)n(U_i) + s'_T A(U_T)D_T}{\sum A(U_i) + s' A(U_T)}$$

삼각형의 세일리언시가 설정된 임계값  $s'$ 을 초과

할 때, 수식 2에 수식 7과 같은 방식으로 세일리언시 방향을 추가해 삼각형의 특성을 반영한다.  $s'$ 는  $\lambda$ 배 분위수 세일리언시에 대해  $\alpha$ 배 증폭된 세일리언시이다. 증폭된 세일리언시를 가중치로 적용해 세일리언시가 큰 영역에서 세일리언시 방향이 더 큰 영향력을 행사하도록 한다. 세일리언시는 0에서 1 사이의 값으로 정규화 되어있기 때문에 증폭된 세일리언시는 0과  $\alpha$ 사이의 값을 가진다.

### 3.3 Result

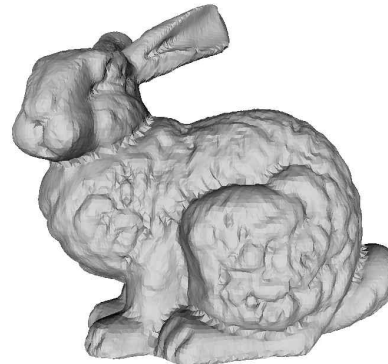
Fig. 5c는 세일리언시 방향을 통합한 스무딩을 적용한 high-boost mesh filtering의 결과로, 표면 텍스처가 개선된 모습을 보여준다. 세일리언시가 70번째 백분위수 이상인 영역의 값을 1.5배 증폭하였다.



(a)



(b)



(c)

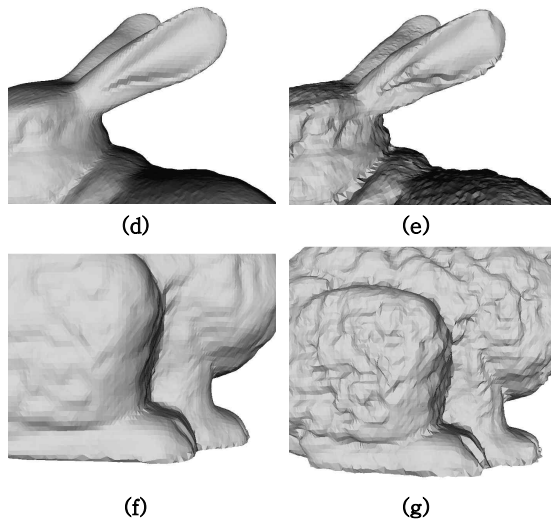


Fig. 5. (a, d, f) original Bunny model, (b) high-boost mesh filtering result, (c, e, g) high-boost mesh filtering with saliency direction result

높은 세일리언스에서 원본의 특성을 보존하려는 접근 덕분에, Fig. 3b와 3d에서 눈에 띄게 확인할 수 있었던 계단현상이 Fig. 5에서 개선된 모습을 보인다. Fig. 6b에서는 무더졌던 돌기 끝부분의 곡률이 유지되며, 6c와 같이 샤프하게 강화된다. 마지막으로, Fig. 6d에서는 지역적 중요도가 높으나 기존 방법에서는 중요하게 여겨지지 않았던 부분이 부각된다.

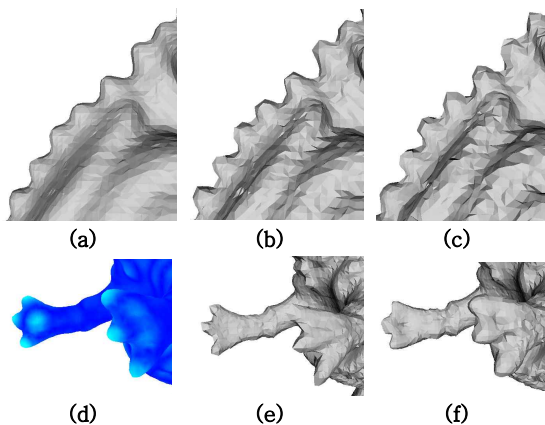


Fig. 6. Dorsal spines of dino (b, e) High-boost Mesh Filtering Result (c, f) High-boost Mesh Filtering with Saliency Direction Result, (d) Saliency Value of dino's hands

## IV. Conclusions

본 논문에서는 high-boost mesh filtering의 새로운 개선 방안으로 메쉬 세일리언스를 기반으로 한 세일리언스 방향 벡터를 제안하고 이를 법선 벡터 계산 과정에 통합하여 계단현상 문제를 완화하는 기법을

제시하였다. 본 기법은 지역적 중요도가 높은 영역의 곡률을 효과적으로 유지하면서 메쉬의 주요 특징을 보존할 수 있음을 확인하였다.

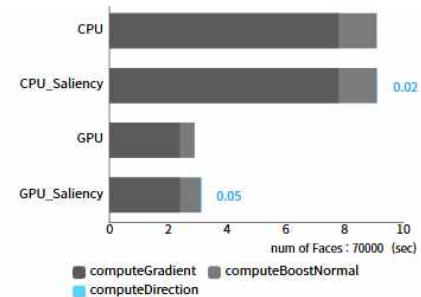


Fig.7 Comparison of High-boost Mesh filtering Elapsed times on CPU and GPU

세일리언스 방향은 초기에 한 번만 계산되며 삼각형의 개수가 약 70,000개일 때 약 0.02초로 매우 짧은 시간이 소요된다. GPU의 경우, 세일리언스가 높은 영역을 구분하는 임계값을 설정하기 위한 백분위수 계산 과정에서 정렬로 인해 CPU보다 약 0.03초가 더 소요된다. 그러나 Fig. 7에서 확인할 수 있듯이 연산 시간의 가장 큰 부분을 차지하는 경사 하강법을 이용한 새로운 좌표 계산 함수에서 CPU와 GPU는 약 3배 이상의 속도 차이를 보이기 때문에 세일리언스 방향을 통합하며 증가한 시간이 전체적인 연산 시간에 부담을 주지 않는다. GPU에서 임계값 설정을 위한 정렬 과정을 제외한다면 약 0.004초로 매우 짧은 시간이 소요되기 때문에 정렬을 사용하지 않는 유효적인 임계값 설정 방식을 도입하면 연산 시간은 더욱 단축될 것이다.

## ACKNOWLEDGEMENT

이 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.RS-2022-00155915, 인공지능융합혁신 인재양성(인하대학교))

## References

- [1] Yagou, H., Belyaev, A., & Wei, D. (2003). High-Boost Mesh Filtering for 3-D Shape Enhancement. *Journal of Three Dimensional Images*, 17(1), 170-175.
- [2] Lee, C. H., Varshney, A., & Jacobs, D. W. (2005). Mesh saliency. *ACM Transactions on Graphics*, 24(3), 659-666.