

Graph-cut 과 주성분 분석을 이용한 Mesh 의 Edge 추출 알고리즘

한현덕, 김해광, 한종기

세종대학교

gusejr0425@sejong.ac.kr, hkkim@sejong.ac.kr, hjk@sejong.edu

Edge Extraction Algorithm for Mesh Data Based on Graph-cut Method and Principal Component Analysis

HyeonDeok Han, HaeKwang Kim, Jong-Ki Han

Sejong University

요 약

LiDAR 장비 및 SfM 과 MVS 방법을 이용하여 생성된 point cloud 와 mesh 에는 항상 노이즈가 포함되어 있다. 이러한 노이즈를 제거하기 위해선 노이즈와 edge 를 효과적으로 구분해낼 수 있어야 한다. 노이즈를 제거하기 위해 mesh 로부터 edge 를 먼저 구분해낸 후 edge 에 해당하는 영역과 평면에 해당하는 영역에 서로 다른 필터를 사용하는 많은 연구들이 있지만 강한 노이즈가 포함된 mesh 에서는 edge 를 잘 구분해내지 못하는 문제가 존재한다. 이러한 방법들은 mesh 로부터 edge 를 구분해내는 알고리즘의 성능이 노이즈를 제거하는 전체 알고리즘의 성능에 큰 영향을 주기 때문에 강한 노이즈에서도 edge 를 잘 구분해낼 수 있는 알고리즘이 필요하다. 본 논문에서는 PCA 와 graph-cut 을 이용하여 강한 노이즈가 포함된 mesh 에서 edge 영역을 추출하는 알고리즘을 제안한다.

1. 서론¹

3 차원 정보는 보통 point cloud 형태로 표현되며 point 의 3 차원 공간상의 좌표와 r, g, b 색상 정보를 포함한다. 이러한 3 차원 정보는 LiDAR 장비를 이용하여 얻을 수 있으며 SfM (Structure-from-Motion)과 MVS (Multi-view Stereo)와 같은 방법을 이용해 2D 이미지로부터 point cloud 를 생성할 수도 있다 [1, 2].

Point cloud 로 표현된 3 차원 정보는 point 사이에 빈 공간이 존재하여 완전한 물체로 표시되지 않는다. 보다 실제의 물체처럼 표현하기 위해 일반적으로 point cloud 로부터 삼각형 면들로 이루어진 mesh 형태를 만들어낸다.

LiDAR 장비 및 SfM 과 MVS 방법을 이용하여 생성된 point cloud 와 mesh 에는 노이즈가 항상 포함되어 있기 때문에 생성된 mesh 의 edge 정보를 보존하면서 노이즈를 제거하기 위한 여러 논문들이 존재한다 [3, 4, 5]. 해당 논문들은 edge 와 같은 특징이 있는 영역과 평평한 영역으로 mesh 를 구분한 후, 각각의 영역에 맞는 방법을 이용하여 노이즈를 제거한다.

본 논문은 mesh 의 노이즈를 제거하는데 필요한 과정인 노이즈가 포함된 mesh 로부터 edge 를 추출하는 알고리즘을 제안한다. 이를 위해 우선 PCA (Principal Component Analysis) 분석을 통해 각각의 면이 edge 에 해당하는 정도를 구하고, 이웃한 면은 서로 같은 영역에 해당하는 경우가 많은 것을 고려해 에너지함수를 설정한 후 graph-cut 방법을 통해 해당 에너지함수가 최소가 되도록 각각의 면을 edge 영역과 평면

¹ 연락처: 한종기

영역으로 labeling 한다.

본 논문의 구성은 다음과 같다. 2 장에서 Mesh denoising 에서 사용되는 기존 방법의 문제점을 설명한 후 해당 문제점을 해결하기 위한 방법을 3 장에서 제안한다. 4 장에서 실험결과를 보인 후 5 장에서 결론짓는다.

2. Mesh Denoising 에서의 활용 및 문제점

Mesh denoising 알고리즘은 노이즈가 포함되어 있는 mesh 로부터 보존되어야 할 edge 정보를 추출해내야 한다. 보통 edge 를 추출하는 알고리즘들은 edge 가 평평한 영역에 비해 고주파영역이라는 특성을 이용하지만 노이즈가 포함된 경우, 노이즈에서도 고주파 특성을 갖기 때문에 노이즈와 edge 를 구분하기 어렵다.

[3]에서는 tensor voting 알고리즘을 사용하여 면들과 point 들을 각각 feature 영역과 non-feature 영역으로 구분하는 방법을 사용했고 [4]는 [3]에서 사용한 tensor voting 알고리즘을 multi-scale 로 확장시켰다. 하지만 [3]과 [4]는 여전히 노이즈에 민감한 모습을 보이는 문제점이 존재한다.

[5]에서는 노이즈가 포함된 mesh 에서 feature 영역을 추출하기 어렵다고 판단하여 노이즈를 줄이는 전 처리 과정을 사용했다. 하지만 강한 노이즈에서는 전 처리 과정으로 충분히 노이즈를 제거할 수 없기 때문에 여전히 노이즈에 민감하다는 같은 문제가 존재한다.

3. 제안하는 알고리즘

제안하는 알고리즘의 목적은 강한 노이즈가 포함된 Mesh 에서도 효과적으로 edge 에 해당하는 면을 추출하는 것이다. 제안하는 알고리즘은 크게 2 단계로 구성된다. 첫 번째 단계는 Mesh 의 면인 삼각형의 중심 좌표들을 이용해 PCA 를 한 후, PCA 를 통해 얻어진 고유값을 이용하여 각각의 면이 edge 영역에 해당하는지에 대한 정도를 0 ~ 1 사이의 값으로 나타낸다 [6]. 보통의 경우 edge 에 해당하는 영역보다 평평한 영역이 더 많기 때문에 gamma 값을 이용하여 앞서 구한 값을 전체적으로 키워주면서 여전히 0 ~ 1 사이의 값을 갖도록 한다. 두 번째 단계는 이전 단계에서 구한 값과 이웃한 면들은 서로 같은 영역일 확률이 높다는 점을 이용하여 에너지 함수를 설정하고, 이 에너지 함수를 최소화하도록 각각의 면이 edge 인 영역과 평평한

영역으로 labeling 될 수 있도록 하기 위해 graph-cut 방법을 사용한다 [7].

3.1 Edge 영역에 해당하는 정도를 [0, 1]에 매핑

본 논문에선 edge 영역에 해당하는 정도를 측정하기 위한 방법으로 각각의 면의 중심 좌표들을 이용해 PCA 분석을 한 후, 가장 작은 고유값의 크기를 이용한다. 기존의 방법들은 보통 edge 를 구분하기 위해 법선벡터를 이용하는데 노이즈가 강하게 포함된 mesh 에서는 법선벡터를 활용하여 edge 를 구분할 수 없다. 노이즈가 강하게 포함된 mesh 에서는 평면 영역에 해당하는 법선벡터들이 edge 영역에 해당하는 법선벡터들 만큼 큰 변화를 일으킬 수 있기 때문이다. 따라서 본 논문에서는 법선벡터를 이용하는 대신 노이즈가 강하더라도 큰 폭의 변화가 없는 mesh 를 구성하는 면의 중심 좌표들을 이용한다.

Mesh 를 구성하는 면이 N 개 존재하고 i 번째 면을 f_i 라고 정의한다. f_i 에서 PCA 분석을 하기에 앞서 PCA 분석에 사용될 f_i 의 이웃 면들을 정의해야 한다. f_i 의 이웃 면들은 f_i 의 중심좌표로부터 중심좌표들 사이의 평균 거리의 k 배에 해당하는 거리 내에 중심좌표가 존재하는 면들로 정의하고 f_i 의 이웃 면들의 집합을 Ω_i 로 정의한다. 이때 f_i 와 이웃 면들의 중심좌표들 중 x 좌표와 y 좌표의 공분산은 식 (1)과 같다.

$$cov(x_i, y_i) = \sum_{j \in \Omega_i} (x_j - m_x)(y_j - m_y) \quad (1)$$

여기서 m_x 와 m_y 는 각각 Ω_i 에 포함되는 면들의 중심좌표 중 x 좌표의 평균과 y 좌표의 평균을 의미하고 x_i 와 y_i 는 각각 f_i 의 x 좌표와 y 좌표를 의미한다. 원래는 normalization 과정이 필요하지만 후에 진행되는 알고리즘의 특성상 normalization 과정은 결과에 영향을 미치지 않기 때문에 생략한다. 이를 바탕으로 f_i 와 이웃 면들의 중심좌표들을 이용한 공분산 행렬 C_i 는 식 (2)와 같다.

$$C_i = \begin{pmatrix} cov(x_i, x_i) & cov(x_i, y_i) & cov(x_i, z_i) \\ cov(x_i, y_i) & cov(y_i, y_i) & cov(y_i, z_i) \\ cov(x_i, z_i) & cov(y_i, z_i) & cov(z_i, z_i) \end{pmatrix} \quad (2)$$

PCA 는 C_i 에 대해 고유값분해를 하는 것으로 고유값분해 결과 3 개의 고유벡터 $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ 와 이에 대응되는 각각의 고유값 $\lambda_1, \lambda_2, \lambda_3$ ($\lambda_1 > \lambda_2 > \lambda_3$) 이 나온다. PCA 를 통해 얻어진 고유벡터 중 \mathbf{e}_1 은 입력으로 주어진 중심좌표들에 대해 분산이 가장 큰 방향을 의미하고, \mathbf{e}_2 는 \mathbf{e}_1 에 수직이면서 다음으로 가장 분산이 큰 방향을 의미하고, \mathbf{e}_3 는 $\mathbf{e}_1, \mathbf{e}_2$ 에 수직이면서 다음으로 가장 분산이 큰 방향을 의미한다. 이때 고유값들은 각각의 대응되는 고유벡터 방향으로의 분산을 의미한다. 따라서 λ_3 값이 작을수록

해당 면은 평면에 위치하고 있다고 볼 수 있다. 다만 주어진 mesh 모델에 따라서 λ_3 값의 범위는 다를 수 있기 때문에 식 (3)과 같이 $[0, 1]$ 사이의 값으로 mapping 해주는 작업이 필요하다.

$$\bar{\lambda}_3^i = \frac{\lambda_3^i - \min}{\max - \min} \quad (3)$$

보통 mesh 에서 edge 영역보다 평면 영역이 더 많기 때문에 $\bar{\lambda}_3^i$ 의 값이 작은 면이 더 많을 수밖에 없다. 이를 보완하기 위해 식 (4)와 같이 γ ($\gamma > 1$) 파라미터를 이용하여 전체적으로 값의 크기를 키우면서 여전히 $[0, 1]$ 사이의 값으로 mapping 되도록 한다.

$$\tilde{\lambda}_3^i = (\bar{\lambda}_3^i)^\gamma \quad (4)$$

그림 1 은 $\tilde{\lambda}_3^i \times 255$ 의 값을 r 색상의 값으로, $(1 - \tilde{\lambda}_3^i) \times 255$ 의 값을 b 색상의 값으로, g 색상의 값은 0 으로 준 결과를 보인다. 그림 1 의 (a)는 γ 의 값을 1로 설정했을 때의 값으로 식 (4)를 적용하지 않은 것과 같은 결과다. 그림 1 의 (b)는 γ 의 값을 1.8로 설정했을 때의 결과다. 그림 1에서 볼 수 있듯이 γ 의 값을 적절히 조절하면 edge 영역과 평면 영역을 잘 구분하여 $[0, 1]$ 사이의 값으로 mapping 될 수 있음을 알 수 있다.

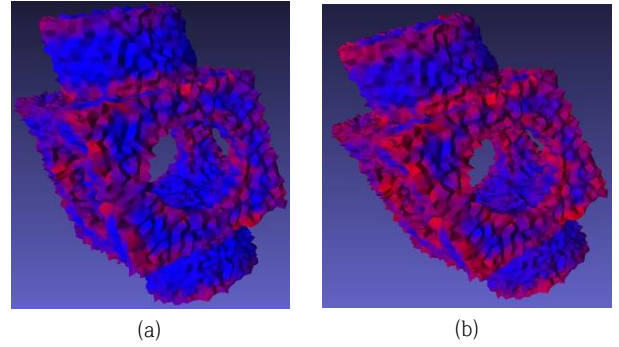


그림 1. 서로 다른 γ 에 따른 $\tilde{\lambda}_3^i$ 값의 시각화.

$$S_{ij}(l_i, l_j) = \begin{cases} \exp\left(-\frac{\|c_i - c_j\|_2^2}{\sigma^2}\right) & \text{if } l_i \neq l_j \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

여기서 σ 는 gaussian 함수의 폭을 결정하는데 본 논문에서는 mesh 의 면들의 중심 사이의 거리의 평균 값을 사용했다.

4. 실험결과

3.2 Graph-cut 을 이용한 edge 영역 추출

앞서 edge 인 영역에 해당하는지에 대한 정도를 $[0, 1]$ 사이의 값으로 정했다. 본 논문의 목적은 edge 영역과 edge 가 아닌 영역으로 각각의 면을 labeling 하는 데에 있고 이 목적을 위해 graph-cut 방법을 사용한다. 이웃한 면은 서로 같은 영역을 공유할 가능성이 높으므로 graph-cut 방법을 도입하기에 적절하다. 식 (5)는 graph-cut 을 통해 최소화할 에너지 함수를 나타낸다.

$$E(l) = \sum_{i=0}^{N-1} D_i(l_i) + w \cdot \sum_{(i,j) \in \mathcal{N}} S_{ij}(l_i, l_j) \quad (5)$$

l_i 는 f_i 가 edge 영역에 해당하는지 혹은 평면 영역에 해당하는지에 대한 label 값이고 D 는 data term 으로 앞서 구한 $\tilde{\lambda}_3^i$ 값을 이용한다. \mathcal{N} 은 서로 이웃한 면들의 집합으로 여기서 말하는 이웃은 1-ring neighbor 를 만족하는 경우를 의미한다. S 는 smoothness term 으로 이웃한 면들의 중심 좌표 c_i, c_j 사이의 거리에 따라 gaussian 함수를 weight 로 적용한 값으로 식 (6)과 같다. w 는 smoothness term 의 weight 값으로 본 논문의 실험결과는 모두 0.1의 값으로 고정한 결과다

본 논문은 실험을 위해 필요한 2 개의 파라미터가 존재한다. 첫 번째 파라미터는 PCA 에 사용될 이웃 면들을 결정하는데 사용될 파라미터인 k 다. 본 논문에서는 k 를 5 로 고정하여 사용했다. 다음으로 필요한 파라미터는 γ 값이다. 그림 2 를 보면 적절한 γ 값을 설정하면 강한 노이즈가 포함된 mesh 에서도 edge 영역을 잘 추출하는 것을 확인할 수 있다.

그림 2 에서 빨간색의 면은 edge 로 labeling 된 면을 의미하고 초록색의 면은 평평한 영역으로 labeling 된 면을 의미한다. 그림 2 의 (a)는 γ 값이 1 일 때의 결과로 식 (4)가 적용되지 않은 결과와 같다. 그림에서 볼 수 있듯이 edge 로 labeling 되어야 할 면들이 edge 로 구별되지 못하고 있다. 이는 mesh 에 평평한 영역이 edge 인 영역보다 많기 때문에 $\tilde{\lambda}_3^i$ 의 값이 대부분 작기 때문이다. 그림 2 의 (b)는 γ 값이 1.4 일 때의 결과로 그림 2 의 (a) 보다 edge 인 영역을 더 많이 찾아내지만 여전히 edge 로 labeling 되어야 할 면들이 edge 로 구별되지 못하고 있다. 그림 2 의 (c)는 γ 값이 1.8 일 때의 결과로 강한 노이즈가 포함된 mesh 에서 모든 edge 를 잘 찾아내는 것을 확인할 수 있다. 마지막으로 그림 2 의 (d)는 γ 값이 2.2 일 때의 결과로 그림 2 의 (c)와 마찬가지로 모든 edge 를 찾아내면서 edge 를 더 강하고 두껍게 찾는 것을 확인할 수 있다.

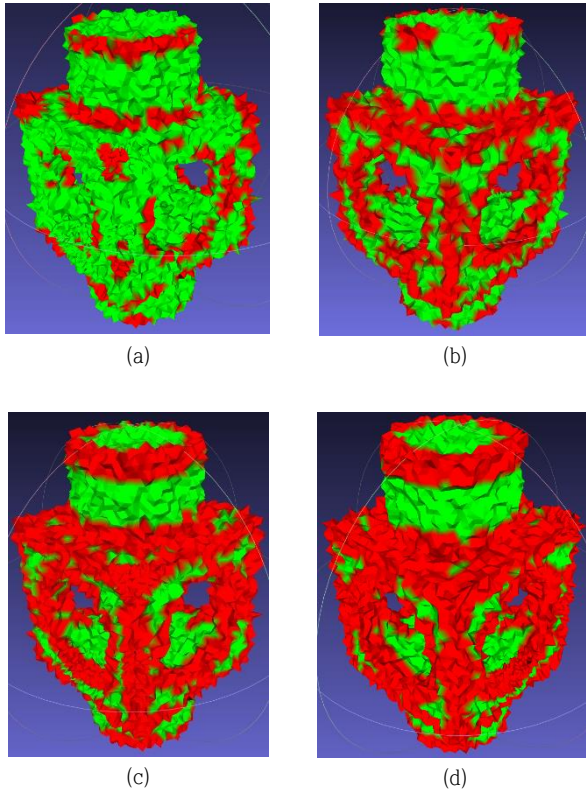


그림 2. 서로 다른 γ 에 따른 edge 추출 결과

5. 결론

본 논문은 강한 노이즈가 포함된 mesh 에서 edge 영역을 추출하기 위해 PCA 와 graph-cut 을 이용했다. 실험결과 γ 을 적절히 조절하여 강한 노이즈가 포함된 mesh 에서도 edge 영역을 효과적으로 추출할 수 있음을 확인했다. 특히 γ 값이 edge 와 평면을 잘 구별하는 값에서 어느정도 커지더라도 edge 영역을 더 강하고 두껍게 추출하는 효과를 보일 뿐이고 여전히 edge 와 평면을 잘 구분하는 모습을 보인다. 다만 γ 값을 수동으로 정해줘야 한다는 한계점 역시 존재하므로 후속 연구를 통해 개선이 필요하다.

ACKNOWLEDGMENT

This work was partly supported by the National Research Foundation of Korea (NRF) under Grant (No. 2019R1F1A1046236) and partly by Institute for Information & communications Technology Promotion (IITP) under Grant (2017-0-00486) funded by the Korea government (MSIT).

참 고 문 헌 (References)

- [1] P. Moulon, P. Monasse, and R. Marlet, "Adaptive structure from motion with a contrario model estimation," *In Asian Conf. Comput. Vision*, pp.257-270, Springer, 2013. Available: github.com/openMVG/openMVG/.
- [2] Shuhan Shen, "Accurate Multiple View 3D Reconstruction Using Patch-Based Stereo for Large-Scale Scenes," *IEEE transactions on image processing*, 22(5):1901–1914, 2013.
- [3] M. Wei, J. Yu, W.-M. Pang, J. Wang, J. Qin, L. Liu, and P.-A. Heng, "Bi-normal filtering for mesh denoising," *IEEE Trans. Vis. Comput. Graphics*, vol. 21, no. 1, pp. 43–55, Jan. 2015.
- [4] M. Wei, L. Liang, W.-M. Pang, J. Wang, W. Li, and H. Wu, "Tensor voting guided mesh denoising," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 931–945, Apr. 2017.
- [5] X. Lu, Z. Deng, and W. Chen, "A robust scheme for feature-preserving mesh denoising," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 3, pp. 1181–1194, Mar. 2016.
- [6] I. T. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," *Philos. Trans. A Math. Phys. Eng. Sci*, vol. 374, no. 2065, Apr. 2016, Art. no. 20150202.
- [7] Y. Boykov and O. Veksler, "Graph Cuts in Vision and Graphics: Theories and Applications," *The Handbook of Math. Models in Computer Vision*, Springer, 2006.