

# 3D Mesh Texturing

□ 이승용, 한종기 / 세종대학교

## 요약

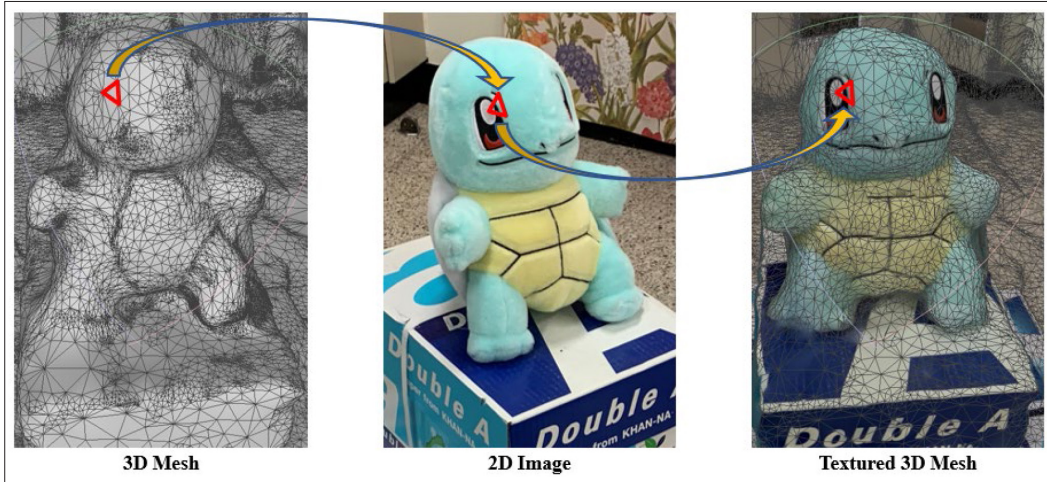
2D 영상들을 이용하여 3D immersive scene을 복원하는 3D reconstruction 기술에서 texturing은 최종 과정이자, 3D 데이터에 현실감을 부여하는 중요한 과정이다. Texturing 방법 중 하나인 projection-based method는 다른 방법들에 비해 상대적으로 노이즈와 카메라 파라미터의 부정확도에 강인하다. 본 기고문에서는 projection-based method에 대해 설명하고 이 과정 중 발생하는 문제점과 이를 해결하기 위한 후속 연구들에 대해 소개하고 개선된 결과를 확인한다. 마지막으로 향후 연구 방향과 보다 현실감 있는 3D 영상 생성을 위해 이루어야 할 목표를 제시한다.

## 1. 서론

2차원 영상을 이용한 3차원 영상을 복원하는 3D reconstruction 기술의 발전으로 인해 전문적인 장비가 없이도, 물체나 환경을 카메라로 촬영한 다수의 영상만을 이용해 3차원 영상을 얻는 것이 가능해졌다. 2D 영상의 특징점을 찾고 다른 영상들과 매칭 관계를 파악하여 카메라의 위치와 방향을 추정하는 SfM(Structure from Motion)[1]과 여러 장의 영상과 카메라 파라미터들을 입력으로 물체의 표면을 추정하는 MVS(Multi View Stereo)

[2]를 이용하여 3D mesh 구조를 형성할 수 있다. 하지만 <그림 1>의 왼쪽과 같이 3D mesh 자체는 색상 정보를 포함하고 있지 않아 현실감이 느껴지지 않는다. Immersive 3D 영상을 생성하기 위해선 <그림 1>과 같이 2D 영상에서 mesh의 face에 맞는 texture를 추정하는 과정이 필요하고 이 과정을 texturing이라고 한다.

3D mesh에 적절한 texture를 추정하여 적용하는 것은 immersive 3D scene을 생성하는 마지막 과정인 만큼 최종적인 결과물의 품질에 큰 영향을 미친다. 잘못 추정된 texture는 3D 영상의 몰입도를 떨어뜨리고 부자연스러움



<그림 1> Texturing이란

이 부각된다. 이에 texturing의 완성도를 높이려는 많은 연구들이 진행되어 왔고, 그 방법에 따라 크게 blending-based method, parameterization-based method, projection-based method로 나누어진다[3]. 본 기고문에서는 위의 3가지 방법에 대한 간단한 소개를 시작으로 다른 두 방법에 비해 상대적으로 노이즈에 강인하고 고스트 현상이나 블러링 현상이 덜 발생하는 projection-based method에 대한 설명과 그 과정에서 발생하는 문제점, 그리고 그 문제점을 개선한 연구들을 소개한다.

## II. 3D Texture Reconstruction

3D mesh는 일반적으로 색상 정보를 포함하지 않고 모두 단일 색상으로 표현된다. 이러한 3D mesh 구조에 현실감을 더하기 위해 mesh에 적절한 texture를 부여하는 방법은 크게 3가지 방법으로 분류할 수 있다[3].

첫 번째 방법은 blending-based method[4,5]이다. 이 방법은 3D mesh 형성에 사용된 영상들을 카메라 파라미터를 이용하여 mesh에 투영하고 투영된 영상들의 가중 평균을 구하여 mesh의 texture로 이용하는 방법이다. 이

러한 방법은 작은 물체나 작은 스케일의 실내 공간에서는 적합하지만 추정된 카메라 파라미터와 depth의 정확도에 크게 의존하기 때문에 높은 정확도를 요구한다. 이 때문에 노이즈에 민감하고 카메라 파라미터나 depth에 오차가 존재할 시, texture에 블러링이 발생하거나 고스트 현상이 나타날 수 있고 3D model의 크기가 결과에 영향을 끼친다는 단점이 있다.

두 번째 방법은 parameterization-based method[6,7]이다. 이 방법은 3D mesh를 segmentation하고 각 segment마다 이미지 texture를 배치한다. 각 mesh의 면적, 법선의 각도 등을 고려한 일정한 법칙에 따라 segmentation을 수행한다. 각 segment마다 2D 이미지를 매핑하여 사용하기 때문에 첫 번째 방법보다 정확도가 높지만 여전히 카메라 파라미터의 정확도에 의존성이 높은 편이고, 같은 segment에 속한 mesh가 해당 2D 이미지에서는 보이지 않거나, 품질이 떨어지는 상황이 발생하기 때문에 부분적으로 부자연스러운 texture가 배치될 수 있다는 단점이 있다.

세 번째 방법은 projection-based method[8,9]이다. 이 방법은 각 mesh의 face마다 하나의 2D 이미지를 할당하여 texture를 배치하는 방법이다. 각 face마다 하나

의 이미지를 선택하므로 각 이미지마다 조명이나 밝기에 의해 완성된 texture의 일관성이 떨어진다는 단점이 있다. 이를 해결하기 위해 [8]은 Markov Random Field 에너지 함수를 이용해 이와 같은 문제를 위한 해결책을 제시하였다. Projection-based method는 앞서 설명한 두 방법에 비해 블러링이나 고스트 현상이 적고, 카메라 파라미터와 depth의 정확도의 영향을 덜 받는다는 장점이 있다. 본 기고문에서는 [9]에서 제시한 projection-based method의 전처리, 뷰 선택(view selection), 색상 조정(color adjustment) 과정을 소개한다.

### III. Projection-based method

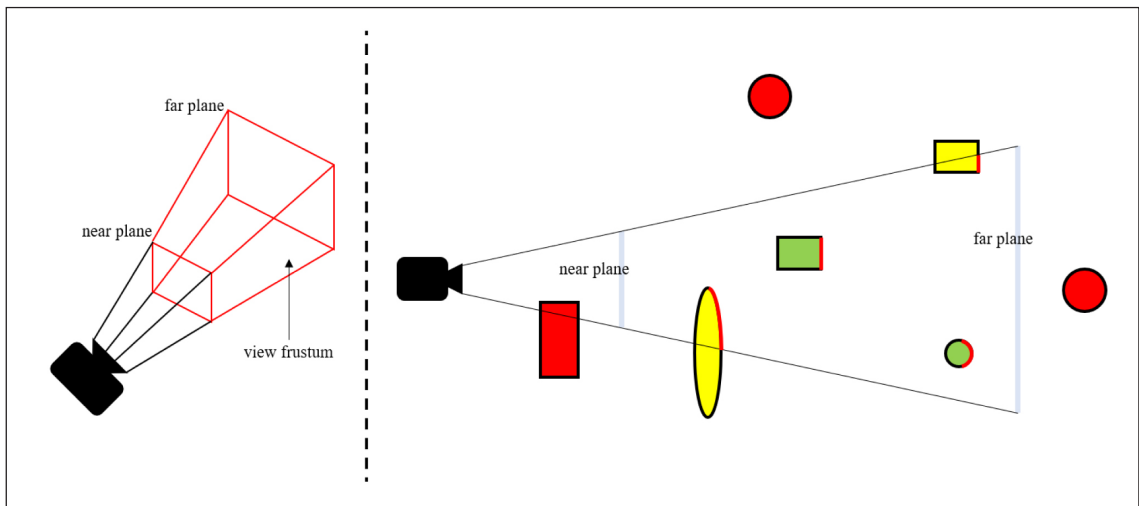
[9]는 projection-based method를 크게 3가지 과정으로 나눈다. 전처리 과정에서는 각 카메라마다 visible한 mesh의 face를 찾고 저장한다. View selection 과정에서는 각 face에서 어떤 카메라의 이미지를 사용할지 정하고 해당 이미지에서 texture patch를 추출한다. 색상 조정 과정에서는 다른 카메라의 이미지에서 texture patch들의 색상 값을 보정하여 보다 일체감 있는 texture를 생

성한다.

#### 1. 전처리 과정

전처리 과정은 각 카메라에서 visible한 face 정보를 추출하는 과정으로 view frustum culling과 back face culling을 통해 visible 정보를 얻는다.

View frustum은 near plane과 far plane으로 이루어지고 해당 영역 안에 있는 face들을 visible하다고 판단한다. <그림 2>에서 빨간색 물체들은 view frustum 밖에 있기 때문에 빨간색 물체를 이루는 face들은 카메라에서 non-visible하고, 노란색 물체를 이루는 mesh의 face 중 view frustum안에 있는 face들과 초록색 물체를 이루는 face들은 visible하다. 또한 <그림 2>에서 노란색과 초록색 물체 중 빨간색으로 표기된 부분을 이루는 face들은 카메라의 이미지상에서 물체의 뒷면에 해당하는 영역이기 때문에 보이지 않는다. 따라서 카메라를 등지고 있는 face들을 non-visible하다고 판단하고 이 과정을 back face culling이라고 한다. 3D mesh를 만드는데 사용된 모든 카메라에 대해서 위 작업을 수행하고 나면 각 face마다 어떤 카메라가 해당 face를 촬영했는지 알 수 있다.



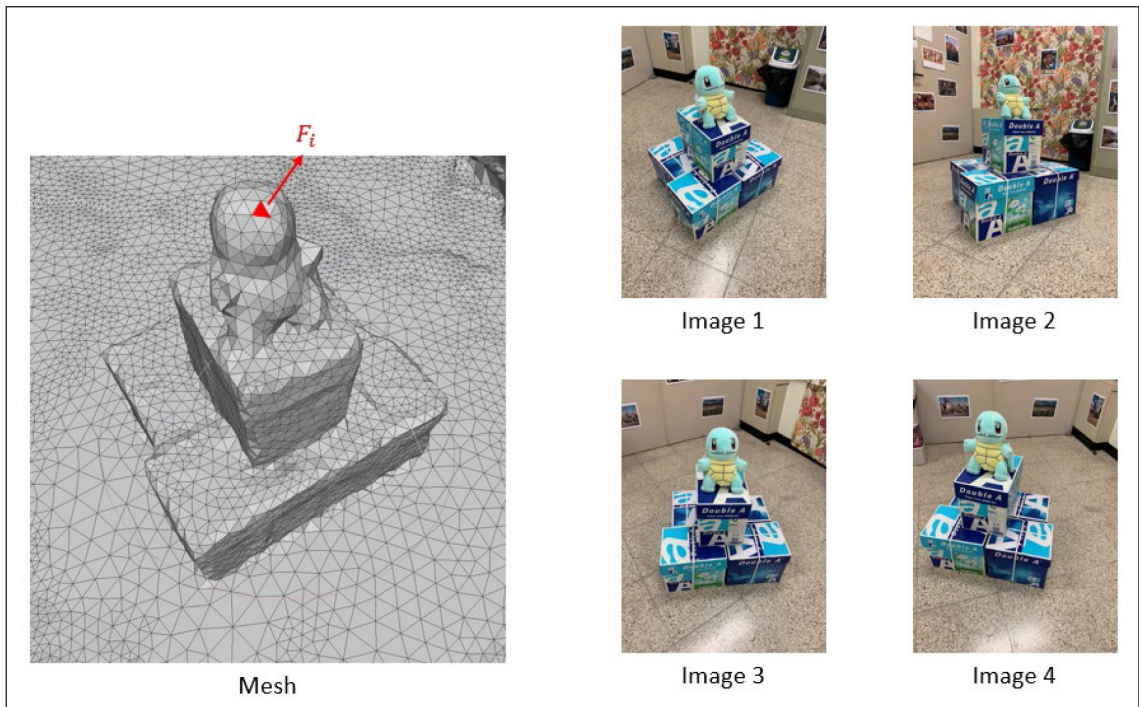
<그림 2> View Frustum Culling & Back Face Culling

## 2. View Selection

$$E(l) = \sum_{F_i \in \text{Faces}} E_{\text{data}}(F_i, l_i) + \sum_{(F_i, F_j) \in \text{Edges}} E_{\text{smooth}}(F_i, F_j, l_i, l_j) \quad (1)$$

Mesh 형성은 다수의 이미지를 이용하여 이루어지기 때문에 한 face의 texture 정보를 담고 있는 이미지 또한 다수이다. 따라서 이러한 다수의 이미지 중 가장 최

적의 texture를 가지고 있는 이미지를 선정하는 과정이 중요하고 이 과정을 view selection이라고 한다. 이 때 사용하는 함수는 MRF(Markov Random Field) 에너지 함수로 식(1)과 같다. MRF 에너지 함수는 data term과 smoothness term으로 이루어져 있다. View selection은 MRF 에너지 함수의 값을 최소화하는 방향으로 진행된다. 이때 최소값을 추정하는 방법은 graph-cut과 alpha



<그림 3> View Selection



<그림 4> Data term에 영향을 주는 case들



expansion[10]을 사용한다.

$$E_{data} = - \int_{P(F_i, l_i)} \left\| \nabla (I_{l_i}(p)) \right\|_2 dp \quad (2)$$

식(1)의 Data term은 식(2)와 같고, 식(2)에서  $P(F_i, l_i)$ 는  $i$ 번째 face  $F_i$ 를 view  $l_i$ 로 projection 시켰을 때 내부에 해당하는 픽셀들이다. 따라서 data term은 이미지로 projection된 삼각형 내부의 픽셀들에 대해, Sobel operator로 구한 gradient의 magnitude 합을 음수로 표현된다. 다시 말해, gradient의 magnitude의 합이 클수록 에너지 함수는 작아진다. Face를 이미지로 projection 시킨 면적이 비슷한 경우에 <그림 4>의 blurred view와 같은 흐릿한 이미지보다 clear view와 같은 선명한(초점이 잘 맞는) 이미지의 변화량이 크게 나타날 것이다. 만약 비슷한 선명도를 가지고 있는 이미지들이 있다면 <그림 4>의 near view와 같이 가까이서 찍은 이미지에서 projection시킨 면적이 더 크므로 gradient 절대값 합이 크게 나타난다. 정리하면 data term은 해당 face를 촬영한 이미지들 중에서 더 선명하고 가까이에서 찍은 이미지에 낮은 에너지를 부여하는 방식으로 만들어졌다.

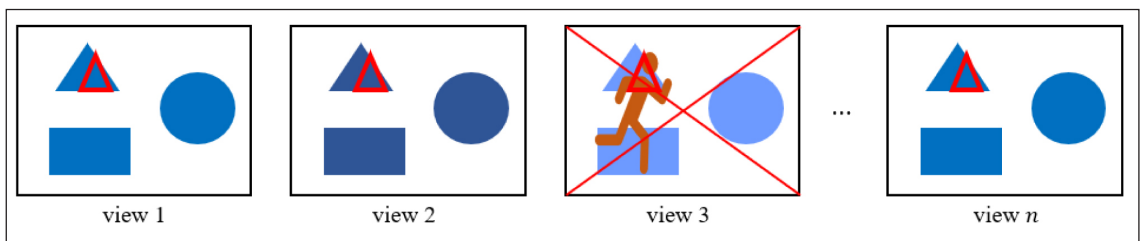
$$\exp\left(-\frac{1}{2}(c_i - \mu)^T \sum^{-1} (c_i - \mu)\right) \quad (3)$$

이때, data term이 gradient를 이용하기 때문에 여러 장의 이미지 중에 움직이는 물체가 있을 경우 gradient가 증

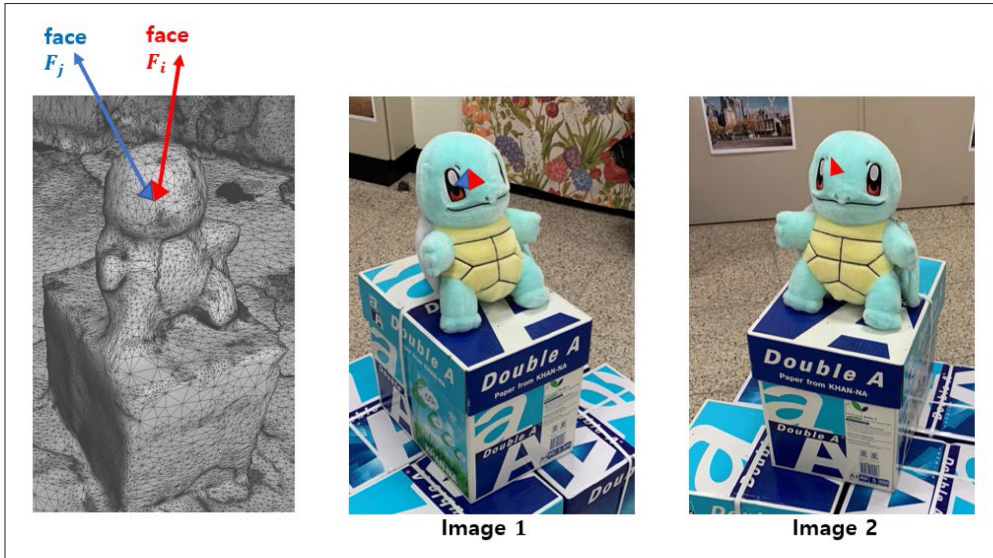
가하여 data term에 의해 선택될 확률이 높아진다. 따라서 이러한 outlier view를 제거하는 것이 중요하고 이 과정을 outlier view removal 또는 photo consistency check라고 한다. <그림 5>에서 빨간색 삼각형이 face를 이미지에 projection시킨 결과일 때 view 3은 움직이는 물체가 이미지에 포함되어 좋은 texture가 아님에도 삼각형 내의 픽셀들의 gradient가 높아 data term의 에너지가 낮게 측정될 확률이 높다. 이러한 outlier view 제거를 위해, view 1부터  $n$ 까지 모든 view에서 삼각형 내의 픽셀 값들의 평균  $c_1, c_2, \dots, c_n$ 을 구한다.  $c_1, c_2, \dots, c_n$ 의 평균  $\mu$ 와 공분산행렬  $\Sigma$ 을 이용하여 식(3)를 모델링한다. 각 view마다 식(4)를 계산한 값이 임계값보다 작으면 outlier로 판단한다. 위 과정을 모든 view가 inlier로 판단되거나 max iteration만큼 수행할 때까지 반복한다. 이때, 충분한 표본 확보를 위해  $n$ 은 4 이상이어야 한다.

Smoothness term은 식(4)와 같고 이웃한 face와 같은 view를 선택하여 texture를 사용할 때 더 낮은 에너지를 갖는다. <그림 6>에서  $F_j$ 가 이미지 1에서의 texture를 이용한다고 했을 때  $F_i$ 가  $F_j$ 와 마찬가지로 이미지 1의 texture를 이용하면  $E_{smooth}$ 는 0, 그렇지 않고 다른 이미지의 texture를 이용하면  $E_{smooth}$ 는 1의 값을 가진다. 이는 이웃한 face들이 같은 이미지의 texture를 사용했을 때보다 일체감이 있고 seam이 나타날 확률이 적기 때문이다.

$$E_{smooth}(F_i, F_j, l_i, l_j) = \begin{cases} 0 & l_i = l_j \\ 1 & l_i \neq l_j \end{cases} \quad (4)$$



<그림 5> Outlier view removal



<그림 6> Smoothness term

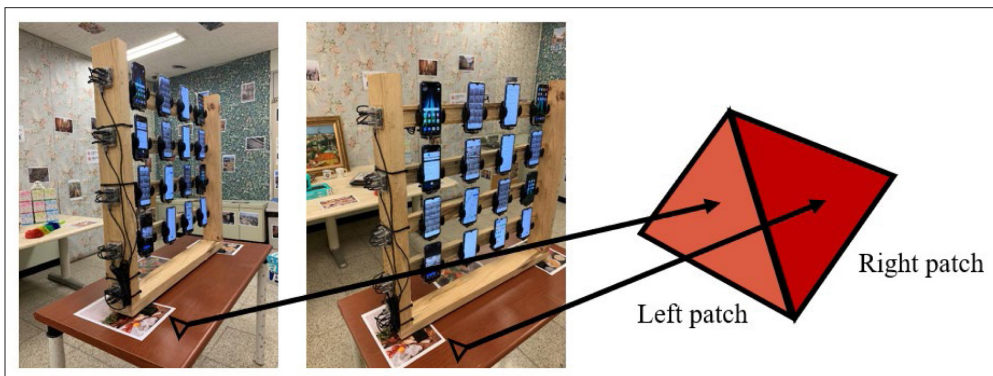
### 3. 색상 조정

앞선 과정들로 인해 얻은 texture들은 여러 장의 이미지에서 얻은 texture이다. 각 이미지들은 카메라의 위치, 각도 또는 카메라의 설정값 등 다양한 요인에 의해, 같은 물체를 촬영했다더라도 이미지마다 밝기와 색상 차이가 존재

한다. 때문에 face들 간의 많은 색상 불연속성이 존재하고 이는 완성된 textured mesh의 일관성을 떨어뜨린다. 이러한 색상 불연속성을 일으키는 seam을 최소화하기 위해 색상 조정 과정이 필요하다.

<그림 7>과 같이 다른 이미지에서 texture를 가져왔을 때 인접한 face에서 컬러값에 의한 seam이 발생한다. 이

$$\underset{g}{\operatorname{argmin}} \sum_{\substack{v \text{ (vertex} \\ \text{on the seam)}}} \left( f_{v_{\text{left}}} + g_{v_{\text{left}}} - (f_{v_{\text{right}}} + g_{v_{\text{right}}}) \right)^2 + \frac{1}{\lambda} \sum_{\substack{v_i, v_j \text{ are} \\ \text{adjacent and} \\ \text{in the same patch}}} (g_{v_i} - g_{v_j})^2 \quad (5)$$



<그림 7> 다른 이미지에서 texture를 가져오며 발생하는 seam

를 보정하기 위한 수식은 식(5)와 같다. 식(5)의 첫 번째 항은 원래 밝기값  $f_{V_{left}}$ 와  $f_{V_{right}}$ 에 각각 가중치  $g_{V_{left}}$ ,  $g_{V_{right}}$ 를 더했을 때 최대한 차이가 적도록 하는 항이고, 두 번째 항은 같은 patch 안에 존재하는 이웃하는 vertex의 가중치의 차이가 적도록 하여 같은 patch 안에서 밝기값의 변화가 점진적이게 하는 항이다. 이러한 조건을 만족하는 가중치들을 찾아 R,G,B 각 채널의 밝기를 조정하여 seam을 줄인다. 이러한 과정을 거쳐도 모든 불연속성이 제거되지는 않기 때문에 Poisson blending[11]을 추가적으로 수행하여 최대한 자연스러운 texture map을 형성한다.

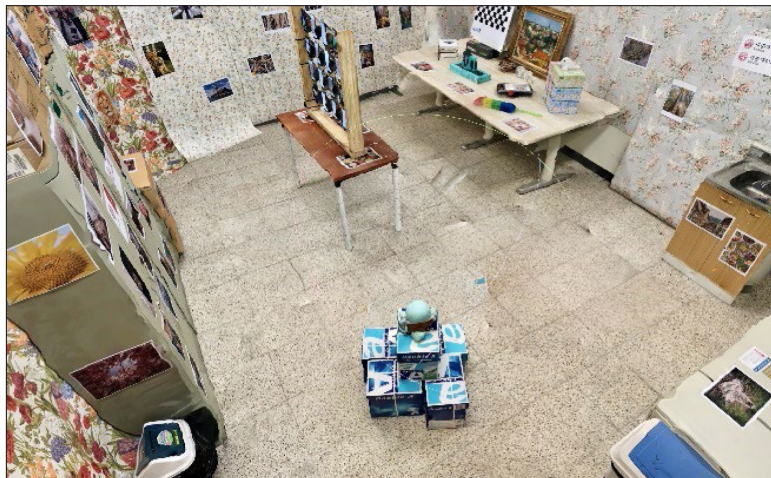
#### IV. 문제점 및 개선 방안

III장에서 설명한 내용에서 outlier 제거와 최적의 texture를 선정하기 위한 방법들이 있었지만, <그림 8>에서 볼 수 있듯이, 일부 영역에서 잘못된 texture를 사용했음을 확인할 수 있다. 이에 대한 원인 중 하나는 III장의 view selection 과정에서 카메라 파라미터의 부정확성과 같은 노이즈에 강인하지 못하기 때문이다. 이번 장에서는 이와 같은 문제를 해결한 연구들[12,13]에 대해서 소개한다.

### 1. Depth 정보를 이용한 Texturing의 View Selection 알고리즘

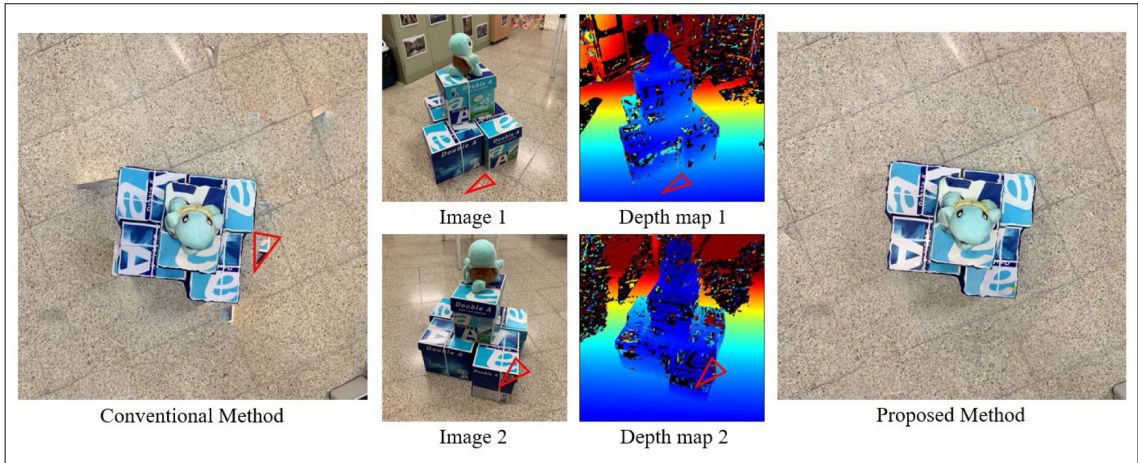
[12]는 view selection의 outlier view removal 과정에서 view의 color 정보만 이용하는 경우 outlier를 판단하지 못하는 경우가 발생하고 이를 해결하기 위해 depth 정보를 활용하는 방법을 제시한다. [9]의 outlier view removal은 color 정보만 이용하기 때문에 view가 충분히 존재해야 outlier를 판단할 수 있고, 움직이는 물체로 인한 outlier는 제거할 수 있지만 카메라 파라미터의 부정확성으로 인해 발생하는 outlier는 제거하기 어렵다는 단점이 있다. 일반적으로 mesh에서 각각의 face는 크기가 작기 때문에 물체의 edge를 포함하지 않는다. 따라서 각 face를 depth map에 projection시켰을 때 depth의 변화량이 일정하게 나타날 확률이 높다. 이에 대한 내용을 바탕으로 식(6)을 outlier 제거에 이용한다. Depth map에 face를 projection시켰을 때, patch 내에 depth의 2차 미분값이 임계값보다 큰 픽셀이 존재하면 outlier로 판단한다.

$$\max_{p \in P(F_i, l_i)} \left| \nabla^2 (D_i(p)) \right| > \tau \quad (6)$$



<그림 8> [9]의 방법을 이용한 texturing 결과





<그림 9> [9]의 방법과 Depth 정보를 이용한 view selection 방법의 결과 비교

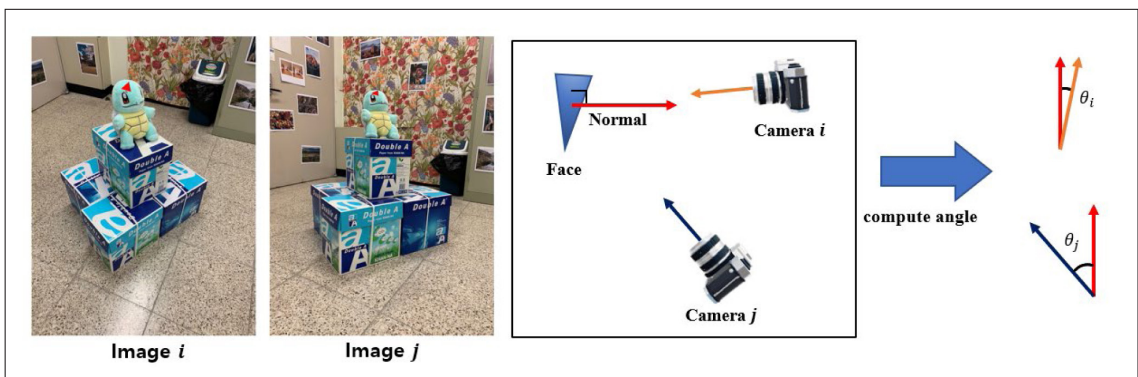
(그림 9)의 왼쪽 그림은 [9]의 방법을 이용한 결과이다. 이와 같이 color값만 이용하였을 때, outlier view를 제거하지 못해 잘못된 texture를 선정하는 경우가 발생한다. 이때 depth map의 변화량을 이용한다면 image 2의 texture는 식(6)에 의해 outlier로 선정되어 제거되고 image 1은 inlier로 판단되어 올바른 texture가 선정되어 오른쪽 그림과 같이 개선된 결과를 얻을 수 있을 것이다.

## 2. 기하정보를 고려한 Texturing의 View Selection

[13]은 식(1)의 data term이 이미지의 gradient만을 고

려하기 때문에, 원래 face와 다른 texture라도 gradient가 높으면 에너지가 낮아져 texture로 선정되는 경우가 있고, 이를 해결하기 위해 face의 normal과 카메라의 방향을 추가적으로 고려하는 방법을 제시한다.

(그림 10)은 빨간색 face를 포함한 view가 이미지  $i$ 와 이미지  $j$ 일 때, face의 normal과 카메라가 이루는 각도  $\theta$ 가 더 작은 view가 더 좋은 texture를 가지고 있을 확률이 높음을 보인다. 이를 근거로 식(2)에 face와 카메라 간의 기하정보를 포함한 가중치  $w_\theta$ 를 추가하여 식(7)을 제안하였다.  $w_\theta$ 는 식(8)과 같고,  $\theta$ 가  $60^\circ$ 보다 크다면 해당 view가 좋은 texture를 가지고 있을 가능성이 적기 때문에  $\theta$ 가  $60^\circ$ 이하일 때는 각도가 작을수록 에너지가 작아지고,



<그림 10> Face의 normal과 카메라의 각도 정보





<그림 11> [9]의 방법과 기하정보를 이용한 방법의 결과 비교

60° 보다 클 때는 에너지 함수에 penalty를 부여하여 에너지를 증가시킨다.

$$E_{data} = - \int_{P(F_i, l_i)} \left\| \nabla (I_i(p)) \right\|_2 dp + w_\theta \quad (7)$$

$$w_\theta = \begin{cases} -\lambda * (90^\circ - \theta_i) & \theta_i \leq 60^\circ \\ \lambda * \theta_i & \theta_i > 60^\circ \end{cases} \quad (8)$$

<그림 11>은 [9]와 [13]의 결과를 비교한 그림으로 [9]의 결과인 왼쪽의 그림에서 나타나던 잘못 texturing된 영역들이 [13]의 결과인 오른쪽 그림에선 사라지면서 보다 자연스러운 3D 영상을 얻을 수 있음을 확인할 수 있다.

## V. 결론

3D mesh의 texture를 추정하는 texturing 과정은 3D 영상을 만드는 최종 과정인 만큼 texturing이 잘못된 경

우 영상의 몰입감이 떨어지며 결과물의 완성도에 큰 영향을 끼친다. 이미지 기반 texturing에는 여러가지 방법과 연구들이 있지만 본 기고문에서는 projection-based method[9]를 소개하였다. 이 방법은 다른 방법에 비해 노이즈나 카메라 파라미터 오차에 대해 강인한 편이지만 여전히 view selection 과정에서 카메라 파라미터의 부정확성과 같은 이유로 잘못된 texture를 선정하는 결과를 보인다. 이를 해결하기 위해 depth 정보를 이용하는 연구[12]와 face와 카메라의 기하정보를 이용하는 연구[13]에 대해 소개하고 결과를 비교하여 문제점이 개선된 것을 확인하였다. 하지만 여전히 2D 영상에서 texture를 추출하는 방식은 2D 영상이 충분히 확보되어야 좋은 성능을 보이고, 빛 반사나 영상의 흔들림 등으로 인해 실제 세계와는 몰입도의 격차가 존재한다. 따라서 향후 연구들은 이러한 문제들을 개선하여, view가 충분하지 않거나, 2D 영상의 품질이 떨어져도 실제 세상과 같은 자연스러운 texture를 생성하는 연구들이 진행될 것으로 예상된다.

## 참 고 문 헌

- [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/](https://github.com/openMVG/openMVG/).
- [2] Cernea, D. OpenMVS: Open Multiple View Stereovision, Available: <https://github.com/cdcseacave/openMVS/>
- [3] Li, S.; Xiao, X.; Guo, B.; Zhang, L. A Novel OpenMVS-Based Texture Reconstruction Method Based on the Fully Automatic Plane Segmentation for 3D Mesh Models, Remote Sens, 2020, 12, 3908.
- [4] Callieri, M.; Cignoni, P.; Corsini, M.; Scopigno, R. Masked photo blending: Mapping dense photographic data set on high-resolution sampled 3D models. Comput. Graph, 2008, 32, 464-473.
- [5] Hoegner, L.; Stilla, U. Automatic 3D reconstruction and texture extraction for 3D building models from thermal infrared image sequences. Quant. InfraRed Thermogr, 2016.
- [6] Inzerillo, L.; Di Paola, F.; Alogna, Y. High quality texture mapping process aimed at the optimization of 3d structured light models. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 2019, 389-396.
- [7] Liu, L.; Ye, C.; Ni, R.; Fu, X.-M. Progressive parameterizations. ACMTrans. Graph. TOG 2018, 37, 1-12.
- [8] Lempitsky, V.; Ivanov, D. Seamless mosaicing of image-based texture maps. In Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, USA, 17-22 June 2007; pp. 1-6.
- [9] Waechter, M.; Moehrl, N.; Goesele, M. Let there be color! Large-scale texturing of 3D reconstructions. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6-12 September 2014; pp. 836-850.
- [10] Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI 23 (2001)
- [11] Pérez, P.; Gangnet, M.; Blake, A. Poisson image editing. ACM Trans. Graph, 2003, 22, 313-318.
- [12] 한현덕, 한종기. (2022). Depth 정보를 이용한 Texturing의 View Selection 알고리즘. 한국방송미디어공학회 학술발표대회 논문집, 442-445.
- [13] 이승용, 한종기. (2023). 기하정보를 고려한 Texturing의 View Selection. Workshop on Image Processing and Image Understanding

## 저 자 소개



### 이 승 용

- 2023년 : 세종대학교 전자정보통신공학과 공학사
- 2023년 9월 ~ 현재 : 세종대학교 정보통신공학과 석사과정
- ORCID : <https://orcid.org/0009-0005-7843-1083>
- 주관심분야 : 영상 신호처리, VR



### 한 종 기

- 1992년 : KAIST 전기및전자공학과 공학사
- 1994년 : KAIST 전기및전자공학과 공학석사
- 1999년 : KAIST 전기및전자공학과 공학박사
- 1999년 3월 ~ 2001년 8월 : 삼성전자 DM연구소 책임연구원
- 2001년 9월 ~ 현재 : 세종대학교 전자정보통신공학과 교수
- 2008년 9월 ~ 2009년 8월 : University California San Diego (UCSD) Visiting Scholar
- ORCID : <https://orcid.org/0000-0002-5036-7199>
- 주관심분야 : VR, 3D 영상신호 구현, 비디오 코덱, 영상 신호처리, 정보 압축, 방송 시스템