한국컴퓨터그래픽스학회
Korea Computer Graphics Society Vol. 24, No. 3, P.61~71

RGB-D 영상으로 복원한 점 집합을 위한 고화질 텍스쳐 추출

서웅[O] 박상욱 임인성[*]

서강대학교 컴퓨터공학과

{wng0620, psu9808, ihm[*]}@sogang.ac.kr

High-quality Texture Extraction for Point Clouds Reconstructed from RGB-D Images

Woong Seo[O] Sang Uk Park Insung Ihm[*]

Department of Computer Science and Engineering, Sogang University

요 약

RGB-D 카메라 촬영 영상에 대한 카메라 포즈 추정을 통하여 복원한 3차원 전역 공간의 점 집합으로부터 삼각형 메쉬를 생성할 때, 일반적으로 메쉬의 크기가 커질수록 3차원 모델의 품질 또한 향상된다. 하지만 어떤 한계를 넘어서 삼각형 메쉬의 해상도를 높일 경우, 메모리 요구량의 과도한 증가나 실시간 렌더링 성능저하 문제뿐만 아니라 RGB-D 센서의 정밀도 한계로 인한 접 집합 데이터의 노이즈에 민감해지는 문제가 발생한다. 본 논문에서는 실시간 응용에 적합한 3차원 모델 생성을 위하여 비교적 적은 크기의 삼각형 메쉬에 대하여 3차원 점 집합의 촬영 색상으로부터 고화질의 텍스쳐를 생성하는 기법을 제안한다. 특히 카메라 포즈 추정을 통하여 생성한 3차원 점 집합 공간과 2차원 텍스쳐 공간 간의 매핑 관계를 활용한 간단한 방법을 통하여 RGB-D 카메라 촬영 영상으로부터 복원한 3차원 모델에 대하여 효과적으로 텍스쳐를 생성할 수 있음을 보인다.

Abstract

When triangular meshes are generated from the point clouds in global space reconstructed through camera pose estimation against captured RGB-D streams, the quality of the resulting meshes improves as more triangles are hired. However, for 3D reconstructed models beyond some size threshold, they become to suffer from the ugly-looking artefacts due to the insufficient precision of RGB-D sensors as well as significant burdens in memory requirement and rendering cost. In this paper, for the generation of 3D models appropriate for real-time applications, we propose an effective technique that extracts high-quality textures for moderate-sized meshes from the captured colors associated with the reconstructed point sets. In particular, we show that via a simple method based on the mapping between the 3D global space resulting from the camera pose estimation and the 2D texture space, textures can be generated effectively for the 3D models reconstructed from captured RGB-D image streams.

키워드: RGB-D 촬영 영상, 카메라 포즈 추정, 3차원 점 집합, 삼각형 메쉬, 텍스쳐 생성.

Keywords: RGB-D image stream, camera pose estimation, 3D point set, triangular mesh, texture generation.

*corresponding author:Insung Ihm/Sogang University(ihm@sogang.ac.kr)

Received : 2018.06.23./ Review completed : 1st 2018.06.29. / Accepted : 2018.07.04.
DOI : 10.15701/kcgs.2018.24.3.61
ISSN : 1975-7883(Print)/2383-529X(Online)

1. 서론

마이크로소프트사의 키넥트 센서(Microsoft Kinect Sensor)와 같은 저가의 RGB-D 카메라가 보급되면서 RGB 이미지만이 아닌 깊이 정보를 활용해서 실제 세상을 인식하거나 사람이나 사물을 3차원 모델로 복원하는 문제가 중요해지고 있으며 이를 해결하기 위한 다양한 연구들이 수행되었다. 최근에는 구글사의 탱고(Google Tango) 프로젝트를 통해 RGB-D 카메라가 탑재된 스마트폰을 상용화하여 출시하였으며 또한 스트럭쳐 센서(Structure sensor)와 같이 기존의 RGB 카메라만 탑재되어 있는 모바일 기기에 추가적으로 깊이 센서를 장착하여 RGB-D 데이터를 생성할 수 있는 장치도 상용화되어 다양한 분야에서 활용되고 있다. 이렇게 RGB-D 카메라 하드웨어가 소형화 및 경량화되어 모바일 기기에 탑재될 정도로 개선되었지만 정밀도 측면에서는 하드웨어적인 한계로 인하여 고성능의 3차원 스캐너에 비해서 매우 부족한 상황이다. 따라서 저가의 RGB-D 카메라를 활용하여 3차원 복원을 수행하기 위해서는 촬영 노이즈나 정밀도 문제가 극복되어야 한다.

RGB-D 카메라를 통해 촬영되는 이미지들은 3차원 모델 복원에 앞서 각 이미지의 픽셀들을 3차원 정점으로 표현하고 이를 하나의 좌표계로 정합하여 점 집합(point cloud) 형태로 만들어야 한다. 이때 정합 방법에 따라서 오차가 발생하게 되고 이는 최종적으로 카메라 촬영 노이즈와 함께 점 집합에 오류를 발생시키며 이후 3차원 모델 복원 과정에서 결과 품질에 문제를 발생시킨다. 정합된 점 집합으로부터 삼각형 메쉬(triangular mesh)를 생성할 때 일반적으로 삼각형 꼭지점에서 주변 점 데이터를 통해 색상을 결정하고 이를 쉐이딩(shading)에 사용한다. 따라서 실제 물체의 평면 부분에 다양한 색상의 그림이나 무늬가 있는 경우 3차원 복원시 삼각형 메쉬의 해상도에 따라 복원된 표면의 품질이 달라진다. 실시간 렌더링 분야에서는 3차원 모델 표면의 질감 표현을 위하여 텍스쳐 매핑 기술이 오랫동안 사용되었으며 3차원 복원시에도 복원된 모델의 표면 질감의 향상을 위해서 메쉬의 해상도를 증가시키는 것이 아니라 RGB-D 촬영 데이터로부터 텍스쳐를 자동으로 생성할 수 있다면 삼각형을 불필요하게 많이 사용하지 않아도 고품질의 3차원 모델을 생성할 수 있다.

한편, 최근 스마트폰과 같은 모바일 기기의 성능이 획기적으로 발전하였다. 하지만 프로세서의 성능이 향상됨과 동시에 모바일 기기에 탑재되는 디스플레이의 해상도 또한 향상되어 최근에는 4K 해상도에 가까운 고해상도의 디스플레이가 탑재되고 있고 배터리나 발열 문제로 인하여 프로세서의 성능 또한 제한적으로 사용이 가능하기 때문에 모바일 기기에서의 렌더링시 아직도 성능 제약이 심한편이라고 할 수 있다. 따라서 모바일 기기에서 그래픽 응용 프로그램을 제작할 때 PC 환경에 비하여 3D 모델의 품질이나 렌더링 효과를 제한하여 실시간 렌더링 성능을 확보하려고 노력하고 있는 상황이다. 이에 본 논문에서는 모바일 환경에서도 실시간 렌더링에 쉽게 활용이 가능하도록 3차원 복원시 저해상도의 삼각형 메쉬로 변환 후 고해상도의 텍스쳐를 자동으로 생성하여 RGB-D 데이터로부터 고품질의 3차원 모델을 생성할 수 있는 방법을 제안한다.

2. 관련연구

Newcombe 등[1]은 마이크로소프트 키넥트 센서를 활용하여 실시간으로 촬영되는 RGB-D 이미지 데이터를 통해 카메라의 좌표계를 추적하여 각 이미지의 픽셀이 의미하는 점 데이터를 하나의 좌표계로 빠르게 정합하는 방법을 제안하였다. 이를 통해 실시간 성능으로 매 프레임을 정합하였으며 점 집합 데이터를 가시화하기 위하여 삼각형 메쉬를 생성하지 않고 GPGPU를 활용하여 레이 캐스팅을 하였다.

Pfister 등[2]은 점 집합 데이터를 삼각형 메쉬로 변환하지 않고 surfel을 사용하여 점 집합 상태에서 가시화 하는 방법을 제안하였다. 이를 통해 데이터 변환 과정 없이 효과적으로 가시화가 가능하나 다양한 그래픽스 응용 분야에서는 점 집합이 삼각형 메쉬로 변환이 되어야 활용성이 높아지므로 데이터 활용 측면에서 한계가 있다.

Kazhdan 등[3]은 점 집합으로부터 삼각형 메쉬를 생성하는 방법을 제안하였는데 촬영 노이즈나 정합 에러, 음영지역 등으로 점 집합에 구멍이 있어도 이를 메우며 효과적으로 삼각형 메쉬를 생성한다. 따라서 저가의 RGB-D 카메라로 생성된 점 집합 데이터에 대해서 효과적으로 삼각형 메쉬 생성이 가능하다. 본 논문에서는 삼각형 메쉬 생성시 해당 방법을 사용하였다.

텍스쳐가 없는 3차원 모델에 텍스쳐를 생성하는 문제는 모델링 분야에서 중요한 문제이며 Lévy 등[4]이 제안하는 방법과 같이 모델을 패치 형태로 나누어 3차원 모델 표면에서의 지역성을 텍스쳐 좌표에 반영하는 형태로 텍스쳐를 생성하여 3차원 모델의 표면을 2차원 텍스쳐 좌표계로 매핑시켰다. 이를 통해 디자이너가 모델의 질감을 표현하는 텍스쳐를 효과적으로 제작할 수 있도록 하였다. 본 논문에서는 3차원 모델이 어떻게 텍스쳐 좌표계에 매핑되는지와 관계없이 점 집합으로부터 각 텍셀(texel)들의 색상을 자동으로 추출하여 고화질의 텍스쳐를 생성하였다.

정밀도가 높은 레이저 스캐너로부터 산출한 점 집합과 이에 연동이 된 여러 장의 고화질 사진 영상으로부터 텍스쳐를 생성하는 연구(관련 기존 기법은 Arikan 등[5]의 논문 참조)들은 주로 해당 삼각형 영역에 대하여 어떤 사진으로부터 색상을 가져와 효과적으로 혼합하는데 주안점을 둔다.

하지만 일반적으로 범용 RGB-D 카메라, 특히 깊이 센서의 성능 한계로 인하여 이를 통하여 복원한 점 집합의 점의 좌표와 속성으로 붙어 있는 색깔에는 적지 않은 노이즈와 오류가 개입되어 있고, 따라서 텍스쳐를 생성하는데 있어 주의를 필요로 한다. 일반적으로 컴퓨터 비전 분야에서는 카메라의 포즈를 실시간으로 그리고 정밀하게 추정하는 문제에 더 관심이 있는데, 성공적인 실시간 SLAM 방법 중의 하나인 Whelan 등의 방법[6]에서와 같이 촬영한 RGB 데이터를 색상 볼륨에 점진적으로 저장할 경우, 반복적인 선형 보간으로 인하여 과도하게 부드러운 텍스쳐를 얻게 된다.

Ma 등[7]은 RGB-D 매핑을 통하여 생성한 고밀도 점 집합을 4각형 형태의 평면적인 영역으로 분할 한 후, 이를 적은 개수의 삼각형으로 곡면을 복원하면서 텍스쳐 공간의 단순한 샘플링을 통하여 텍셀의 색깔을 생성하였다. Liu 등[8]은 Kinect 센서에 고화질 카메라를 장착하여 고화질의 색상 정보를 얻었는데, Ma 등[7]의 방법과 유사하게 단순한 매핑 방법을 사용하여 텍스쳐를 산출하였다.

한편 Zhou와 Koltun[9]은 RGB-D 카메라로부터 산출한 카메라 포즈와 기하 모델의 부정확성을 고화질의 촬영 영상에 대한 비선형 최적화 기법을 적용하여 향상된 색상 매핑이 가능토록 하였는데, 이는 고화질의 텍스쳐 생성에 큰 도움이 될 것이다.

Handa 등[10]은 RGB-D 데이터를 사용하는 주행 궤적 추정 알고리즘을 검증하기 위한 방법을 제안하였으며 실제 촬영된 RGB-D 데이터가 아닌 가상의 실내공간을 렌더링하여 RGB-D 데이터를 생성하고 이를 통해 알고리즘간 성능을 비교하도록 하였다. 본 논문에서도 해당 데이터에 대한 실험을 추가하여 제안하는 방법을 검증하였다.

3. 3차원 복원

실제 세상의 물체를 3차원으로 복원하기 위해서 가장 우선적으로 중요한 것은 고품질의 점 집합을 생성하는 것이다. RGB-D 이미지를 촬영하는 하드웨어에 따라 차이점은 있지만 일반적으로 RGB-D 카메라로 촬영되는 이미지는 노이즈가 있으며 특히 깊이 이미지에 대한 노이즈가 심하게 발생한다. 따라서 다양한 이미지 필터링을 활용하여 전체 이미지에서 노이즈를 억제하고 신뢰할만한 픽셀들을 탐지하여 이를 기준점으로 각 이미지들을 하나의 좌표계로 정합하게 된다. 정합 과정도 크게 프레임-대-프레임(frame-to-frame)과 프레임-대-모델(frame-to-model) 방법 나눌 수 있으며 어떤 방법을 선택하느냐에 따라 완성된 점 집합의 품질이나 특성이 달라진다. 본 논문에서는 점 집합을 생성하는 과정에 대해서는 다루지 않는다. 하지만 실험의 다양성을 위하여 노이즈 없이 가상으로 렌더링된 이미지를 프레임-대-모델 방법을 사용하여 고품질로 정합된 점 집합 데이터와 실제 촬영된 RGB-D 데이터를 프레임-대-프레임 방법으로 정합하여 노이즈나 오류가 상당히 존재하는 점 집합 데이터를 모두 실험에 활용하였다.

3차원 복원 과정의 예시로 Figure 1은 점 집합을 삼각형 메쉬로 변환한 결과이다. 먼저 ICL-NUIM 데이터셋[10] 중 Living Room 데이터에 대해서 Whelan 등[11]이 제안한 정합 기법으로 점 집합을 생성한 후 Screened Poisson Surface Reconstruction 기법[3]으로 삼각형 메쉬를 생성하였다. (a)는

(a) Point cloud (880,794 points)

(b) Triangular mesh (117,129 vertices, 234,128 faces)

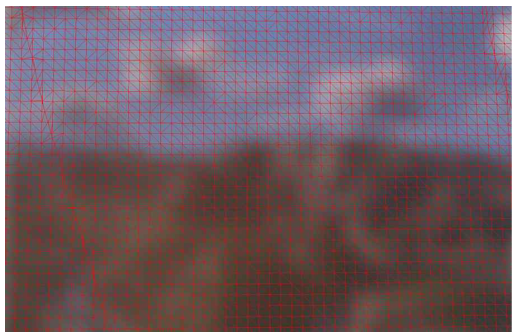
(c) Triangular mesh (1,005,482 vertices, 2,010,844 faces)

Figure 1. Two triangular meshes created from the same point set

점 집합 데이터 원본이며 실제 RGB-D 카메라로 촬영된 데이터가 아닌 실험을 위해 가상으로 노이즈 없이 생성된 가상의 데이터이다. (b)와 (c)는 삼각형 메쉬로 변환한 결과로 인자를 조절하여 메쉬의 해상도를 다르게 생성하였다. (c)의 경우 (b)에 비해 삼각형이 약 10배정도 많이 생성이 되었기 때문에 더욱 정밀하게 복원이 된 것처럼 보이지만 변환된 모델 (c)의 벽 부분을 보면 노이즈가 없이 생성된 점 집합임에도 삼각형이 잘게 쪼개지면서 노이즈가 생성된 것처럼 표면이 거칠어져 오히려 모델의 품질이 떨어지는 것을 확인할 수 있다. 반면에 생성된 모델에서 그림 액자 부분을 보면 삼각형 메쉬의 해상도가 높은 쪽인 (c)에서 그림의 품질이 (b)에 비해 상당히 좋아진 것을 확인할 수 있다. 이에 대한 원인은 Figure 2에서 확인할 수 있다. Figure 2는 해당 그림 액자를 확대한 것으로 삼각형의 수가 약 10배로 증가함에 따라 wireframe을 보면 삼각형들의 크기가 거의 픽셀 크기 수준으로 조밀하게 생성된 것을 확인할 수 있다. 따라서 점 집합으로부터 삼각형 메쉬 생성시 모델의 색상 정보가 삼각형 꼭지점의 속성으로 저장되기 때문에 당연히 메쉬의 해상도가 높은 쪽에서 그림의 품질이 좋아진다.

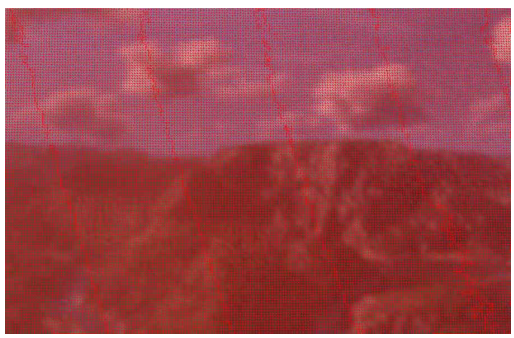
제시된 예의 경우 실제 촬영된 데이터가 아니라 가상으로 만들어진 3차원 실내 공간을 렌더링을 통해서 RGB-D 데이터를 생성하였으므로 색상이나 깊이 정보에 대한 촬영 노이즈 및 오차가 없는 데이터를 기준으로 점 집합이 생성되었다. 즉 모델의 벽면과 같은 평면은 점 집합 상태에서도 매우 균일한 표면을 가지고 있음에도 너무 조밀하게 삼각형 메쉬가 생성되면 오히려 오류가 발생한다. 따라서 실제 촬영된 RGB-D 데이터, 더 나아가 키넥트 센서나 모바일 기기에 탑재되는 저가의 카메라로 촬영된 RGB-D 데이터의 경우 실제 물체는 매끄러운 표면임에도 점 집합 상태에서는 매우 거칠게 생성되는 경우가 많다. 또한 각 RGB-D 이미지들의 정합 방법에 따라서 프레임간 단차 등 오차도 발생하므로 최종 점 집합에서는 다양한 오류가 포함되어있을 수 있다. 본 논문에서 실험한 실제 촬영 데이터의 경우에도 삼각형 메쉬 생성시 해상도를 증가시키면 실제는 매끄러운 표면임에도 생성된 메쉬는 매우 거칠고 불규칙하게 표면이 생성되는 것을 확인하였다. 게다가 메쉬 해상도 증가에 따른 메모리 사용량 증가 및 삼각형 수 증가하면서 실시간 렌더링에서의 성능 저하 또한 문제가 된다. 이와 같이 3차원 복원시 메쉬의 해상도는 항상 높은게 좋은 것이 아니라 점 집합의 오차나 구멍 등 정합된 최종 점 집합의 전체적인 품질이나 생성된 모델의 응용 범위에 따라서 삼각형 메쉬 생성시 적절한 해상도가 있다.

(a) Low resolution mesh (117,129 vertices, 234,128 faces)



(b) Low resolution mesh shown with wireframe

(c) High resolution mesh (1,005,482 vertices, 2,010,844 faces)



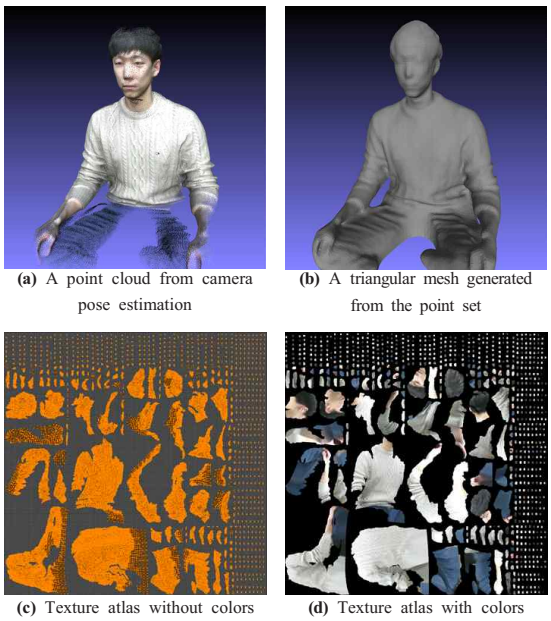
(d) High resolution mesh shown with wireframe

Figure 2. Texture quality comparison between two different meshes

4. 본 논문 제안 텍스쳐 생성 방법

4.1 텍셀 색상 결정

전체 과정은 점 집합 데이터와 임의의 텍스쳐가 매핑되어 있는 삼각형 메쉬가 입력으로 들어와서 텍스쳐 이미지가 생성되어 출력되는 것으로 요약할 수 있다. 즉, Figure 3에서 (a), (b), (c)가 입력에 해당하고 최종적으로 (d)가 만들어지는 과정이다. 본 논문에서 제안하는 텍스쳐 생성 방법은 앞서 언급한대로 RGB-D 이미지로부터 점 집합을 생성하는 방법이나 점 집합에서 삼각형 메쉬를 생성하는 방법, 삼각형 메쉬의 표면을 임의의 비어있는 텍스쳐 이미지와 매핑하는 방법은 어떠한 방법을 적용하더라도 관계가 없이 동작한다.



(a) A point cloud from camera pose estimation

(b) A triangular mesh generated from the point set

(c) Texture atlas without colors

(d) Texture atlas with colors

Figure 3. Texture generation from a reconstructed point set

입력과 출력을 수식으로 표현하면 다음과 같다. 점 집합을 구성하는 각 점들은 위치정보, 법선 벡터, 색상 정보를 가지고 있고 삼각형 메쉬는 각 삼각형마다 위치정보, 법선 벡터, 텍스쳐 좌표 정보를 가지고 있다.

입력:

점 집합 : $P = \{(p_i, n_i, c_i) \mid i = 1, 2, \dots, n_p\}$

삼각형 메쉬 : $M = \{T_i \mid i = 1, 2, \dots, n_t\}, where$

$$T_i = (v_{i0}, v_{i1}, v_{i2})$$

$$v_{ij} = (p_{ij}, n_{ij}, tc_{ij})$$

출력:

텍스쳐 이미지 : I

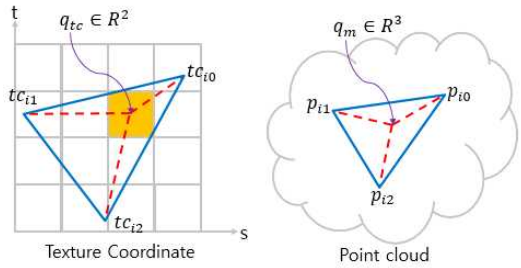


Figure 4. Texel color generation using barycentric coordinate system

텍스터의 색상을 결정하기 위해서는 먼저 모든 삼각형에 대해서 2차원 텍스쳐 좌표계상의 삼각형과 교차하는 모든 텍셀들을 찾는다. 이를 위해 Figure 4와 같이 텍스쳐 좌표계상의 삼각형의 세 꼭지점($tc_{i0}, tc_{i1}, tc_{i2}$)과 해당 삼각형이 포함되는 사각형 경계(bounding box)에 포함되는 내부 모든 텍셀의 중점(q_{tc})들에 대해 아래 수식과 같이 삼각형의 표면적(surface area)을 활용하여 무게중심 좌표(barycentric coordinate)를 계산하고 이때 $\alpha+\beta+\gamma=1$을 만족하는 텍셀을 찾는다. 이러한 방법으로 삼각형 내부와 교차하는 텍셀들을 찾는것과 동시에 각 텍셀 중점에서 계산된 α, β, γ를 활용하여 각 텍셀 중점을 3차원으로 다시 사상시킨다.

$$\alpha = \frac{SurfaceArea(q_{tc}, tc_{i1}, tc_{i2})}{SurfaceArea(tc_{i0}, tc_{i1}, tc_{i2})}$$

$$\beta = \frac{SurfaceArea(q_{tc}, tc_{i2}, tc_{i0})}{SurfaceArea(tc_{i0}, tc_{i1}, tc_{i2})}$$

$$\gamma = \frac{SurfaceArea(q_{tc}, tc_{i0}, tc_{i1})}{SurfaceArea(tc_{i0}, tc_{i1}, tc_{i2})}$$

위에서 기술한 바와 같이 텍셀 중점 q_{tc}에서 계산된 α, β, γ와 3차원 공간 상 삼각형의 세 꼭지점(p_{i0}, p_{i1}, p_{i2})을 활용하여 다음 수식을 통해 2차원 공간상의 텍셀 중점(q_{tc})을 3차원 공간상 점(q_m)으로 변환이 가능하다. 이때 q_m에서의 법선 벡터 또한 동일한 방법으로 계산이 가능하다.

$$q_m = \alpha p_{i0} + \beta p_{i1} + \gamma p_{i2}$$
$$n_m = \alpha n_{i0} + \beta n_{i1} + \gamma n_{i2}$$

이와 같이 텍스쳐 좌표계에서 삼각형의 넓이를 통해 무게중심 좌표를 계산할 때 삼각형의 넓이가 매우 작아 계산과정에서 수치적으로 불안정해지는 경우가 있다. 따라서 삼각형의 크기가 매우 작은 경우에는 위에서 기술한대로 무게중심 좌표를 통해 3차원 좌표를 생성해내지 않고 단순히 삼각형의 중점을 계산하여 주변 점들을 탐색하였다. 이때 어차피 삼각형의 크기가 작기 때문에 중점을 사용하여도 문제가

발생하지 않는다.

텍셀 중점 q_{tc}를 3차원 상의 점 q_m으로 변환하고 해당 지점에서 점 집합을 탐색하여 q_m과 거리가 가장 가까운 순서대로 점들을 찾아 탐색된 점들의 색상 평균값을 통해 텍셀의 색상을 결정한다. 이때 전체 점 집합 탐색시 빠른 성능을 위하여 전처리로 점 집합을 공간 가속 자료구조인 kd-tree를 구축하여 사용하였다. 텍셀의 색상을 추출하기 위해 점 q_m과 거리가 가까운 점들을 탐색할 때 최대 몇 개의 점을 찾아서 색상을 누적시키느냐에 따라 텍스쳐의 품질이 달라진다. 탐색하는 점의 수가 많아질수록 텍스쳐는 부드러워지고 적어질수록 선명한 텍스쳐를 생성할 수 있다. 따라서 텍셀 색상 추출과정에서 인자값으로 탐색할 점의 수를 조절하여 원하는 텍스쳐의 품질을 조절할 수 있다. 앞서 기술한 바와 같이 가상의 데이터가 아니라 실제 RGB-D 카메라를 통해 생성되는 점 집합은 오차나 오류가 많기 때문에 점 집합에 노이즈가 많은 경우 텍스쳐를 부드럽게 만드는 것이 좋은 경우도 있으므로 입력으로 주어지는 점 집합의 품질에 따라서 생성되는 텍스쳐의 품질 또한 조절해야한다.

4.1 텍스쳐 후처리

삼각형 메쉬를 생성하고 메쉬의 표면을 2차원 텍스쳐 좌표계로 매핑할 때 매핑 방법에 따라 균일한 표면임에도 삼각형들의 텍스쳐 좌표가 여러 지역으로 나뉘는 경우가 발생한다. 이후 삼각형과 교차하는 텍셀들을 계산하여 색상을 결정할 때 텍스쳐 좌표는 실수 공간이지만 실제로는 이산적인 메모리 공간에 매핑이 되기 때문에 쉐이딩시 삼각형의 세변에서 에일리어싱(aliasing) 현상이 발생할 수 있다. Figure 5는 이러한 문제점이 실제 렌더링에서 영향을 끼치는 상황을 보여준다. 텍스쳐 좌표계에서의 에일리어싱 현상이나 무게중심 좌표를 통해 삼각형 세 변에서 교차되는 텍셀을 결정할 때 발생하는 수치적 오류 및 미세한 오차로 인하여 텍스쳐에서 아무 색상도 없는 부분이 물체에 매핑되어 삼각형 세 변 근처에서 검은색 선이 보이게 된다.

Figure 5. Artefacts in the meshes and textures

이러한 문제점을 해결하기 위하여 전체 삼각형에 대해 교차하는 텍셀들의 색상을 모두 추출한 후에 색상이 결정된 텍셀에서 빈 공간 방향으로 색상을 확장시키는 형태로 필터링을 수행하였다. Figure 6은 색상 확장시 적용한 필터링을 나타낸 것으로 현재 텍셀에서 주변 8개의 텍셀을 확인하여 미리 계산된 색상이 있는지 확인한다. 이후 미리 계산된 텍셀들의 색상을 평균내는 방법으로 색상을 확장시켰다. Figure 6 (a)의 경우 미리 계산된 텍셀이 주변에 4개가 있으므로 4개의 텍셀 값을 모두 더한 후에 4로 나누며 마찬가지로 Figure 6 (b)는 미리 계산된 텍셀이 주면에 2개가 있으므로 동일한 방법으로 현재 텍셀의 색상 값을 결정한다.

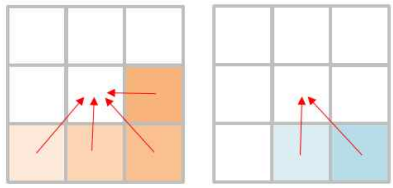


Figure 6. Texel filtering

Figure 7은 필터링을 통해 확장되는 텍셀을 붉은색으로 표시한 결과이다. 텍스쳐 공간에서 삼각형의 세 변에 인접한 텍셀은 지역성을 가지고 있기 때문에 색상을 확장하여도 다른 삼각형과 겹치지만 않는다면 문제가 없다. 에일리어싱을 제거하려면 텍스쳐의 해상도에 따라 두께의 변화가 있어야 하지만 대체적으로 3픽셀 정도 두께면 대부분의 에일리어싱이 사라지는 것을 확인하였다. 본 논문에서는 텍스쳐 공간에서 삼각형들 주변으로 5픽셀 두께로 확장시켰다. 이 방법을 적용하는 경우 Figure 5에서와 같이 삼각형이 끝나는 지점에서 발생하는 검은색 선이 사라지게 된다.

Figure 7. Texture post-processing

5 실험결과

5.1 실험 환경

본 논문에서는 두 가지 형태의 실험 데이터를 사용하였다. 먼저 오류 및 오차가 없는 데이터에 대해서 실험하기 위하여 ICL-NUIM 데이터셋[10]의 Living room과 Office room 데이터를 Whelan 등[11]이 제안한 정합 기법으로 점 집합을 생성하였다. 다음으로 마이크로소프트사의 키넥트 센서를 사용하여 실제 사람과 차량의 앞 그릴 부분을 촬영한 RGB-D 데이터를 An 등[12]의 방법인 프레임-대-프레임 기반 정합 방법을 사용하여 점 집합을 생성하여 실험하였다. 두 종류의 점 집합 실험 데이터에 대해서 삼각형 메쉬 생성은 Kazhdan 등[3]이 제안한 Screened Poisson Surface Reconstruction 기법으로 메쉬를 생성하였다. Table 1은 실험 데이터에서 점 집합을 구성하는 전체 점의 수와 삼각형 메쉬를 구성하는 정점 및 삼각형의 수를 나타낸다. 텍스쳐 생성시에 텍셀당 탐색하는 주변 점의 수는 10개로 설정하였으며 텍스쳐의 해상도는 모두 균일하게 2048x2048 해상도로 생성하였다.

Table 1. Four test datasets

	Point Cloud	Triangular Mesh	
		Vertex no.	Face no.
Living room	880,794	117,129	234,128
Office room	1,019,965	98,552	197,092
Man	2,125,409	37,357	74,584
Car-grille	2,361,274	37,717	75,232

5.2 실험 결과

Figure 8, 9, 10, 11은 실험 결과를 보여준다. 각 그림에서 (a)는 점 집합 원본 데이터이며 (b)는 점 집합을 삼각형 메쉬로 변환한 결과이다. 여기에 추가로 삼각형 메쉬의 밀도를 확인할 수 있도록 모든 삼각형들을 wireframe으로 표시하였다. (c)는 3차원 복원시 가장 기본적인 방법이라고 볼수 있는 삼각형 메쉬 데이터 생성시 각 삼각형 꼭지점에서 속성으로 색상 정보를 넣어 쉐이딩을 한 결과이다. 각 실험결과의 (b)를 통해 가상의 데이터와 실제 촬영 데이터 모두 충분히 많은 삼각형이 생성된 것을 확인할 수 있으며 그럼에도 불구하고 Living room과 Office room의 경우 그림 액자 부분을 보면 삼각형 메쉬의 꼭지점에만 색상 정보가 있기 때문에 그림이 많이 흐려진 것을 확인할 수 있다. 하지만 (d)와 같이 본 논문에서 제안하는 텍스쳐 생성 기법을 통해 동일한 삼각형 메쉬 데이터에 대해서 그림이 선명하게 렌더링 되는 것을 확인할 수 있다. 실제 촬영된 RGB-D 이미지로 생성된 Man과 Car-grille 데이터(Figure 10, 11)도 옷의 질감이나 차량의 번호판을 보면 본 논문에서 제안하는 기법을 통해 렌더링시 품질이 향상된 것을 확인할 수 있다.

Table 2는 각 실험 데이터에 대해서 텍스쳐 생성시간을 측정한 결과이다. Intel Core i7-8700K(3.7GHz) CPU가 탑재된 PC에서 1 쓰레드로 수행하였을 때 결과이며 해상도가 1024x1024에서 2048x2048로 증가할 때 픽셀수가 4배로 증가하므로 수행시간 또한 비례하여 증가하는 것을 확인할 수 있다. 전체적인 수행시간은 점 집합을 구성하는 점의 수와 삼각형 메쉬를 구성하는 삼각형의 수, 전체 텍스쳐에서 모든 삼각형이 차지하는 텍셀의 수가 많아질수록 길어진다. 제안하는 텍스쳐 생성 방법이 삼각형 단위나 픽셀 단위에서 독립적으로 수행이 가능하기 때문에 향후 CPU 멀티쓰레드나 GPGPU를 사용한 가속을 통하여 상당한 성능 향상이 가능할 것으로 예상된다.

Table 2. Experimental result on four test datasets

	Time(s)	
	1024^2	2048^2
Living room	216.72	840.38
Office room	176.41	699.37
Man	51.89	204.07
Car-grille	482.56	1816.46

6. 결론

본 논문에서는 3차원 복원을 통한 삼각형 메쉬 생성시 모바일 기기 등 낮은 성능의 컴퓨팅 환경에서도 모델의 실시간 렌더링에서의 활용을 위하여 비교적 적은 수의 삼각형으로 구성된 메쉬에 고화질의 텍스쳐 매핑을 통하여 고품질의 모델을 생성할 수 있음을 확인하였다. 제안한 텍스쳐 추출 기법은 각 텍셀의 색상을 결정할 때 3차원 공간상에서 탐색할 주변 점의 수를 변경하여 텍스쳐 품질 조절이 가능하기 때문에 점 집합 데이터의 품질에 따라 원하는대로 품질을 조정할 수 있다. 특히 실험에 사용한 Man 데이터와 같이 실제 사람을 촬영하여 생성한 데이터의 경우 사람은 호흡 등으로 인하여 촬영 시간동안 몸을 완전히 정지하는 것이 불가능하므로 사물을 촬영하는 것과 비교해서 노이즈나 오차가가 상대적으로 더 많이 발생할 수밖에 없다. 이런 경우 점 집합 정합 과정에서도 문제가 발생할 수 있기 때문에 고해상도의 메쉬로 변환하면 오히려 노이즈나 정합시 발생하는 단차 등이 그대로 드러나게 되므로 삼각형 메쉬는 저해상도로 부드럽게 생성하고 텍스쳐의 품질 조절하여 복원 모델의 완성도를 높일 수 있다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW 중심대학지원사업의 연구결과로 수행되었음 (2015-0-00910).

References

[1] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," *Proc. the 2011 IEEE International Symposium on Mixed and Augmented Reality (ISMAR),* pp. 127-136, October 2011.

[2] H. Pfister, M. Zwicker, J. van Baar, M. Gross, "Surfels: Surface elements as rendering primitives," *In ACM Transactions on Graphics (Proc. ACM SIGGRAPH 2000),* pp. 335-342, 2000.

[3] M. Kazhdan and H. Hoppe, "Screened Poisson surface reconstruction," *ACM Transactions on Graphics (TOG),* Vol. 32, No. 3, pp. 1-13, June 2013.

[4] B. Lévy, S. Petitjean, N. Ray, and J. Maillot. "Least squares conformal maps for automatic texture atlas generation." *ACM Transactions on Graphics (Proc. ACM SIGGRAPH 2002),* Vol. 21, No. 3, pp. 362-371, 2002.

[5] M. Arikan, R. Preiner, C. Scheiblauer, S. Jeschke, and M. Wimmer, "Large-scale point-cloud visualization through localized textured surface reconstruction," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 20, No. 9, pp. 1280-1292, 2014.

[6] T. Whelan, M. Kaess, H. Johannsson, M. Fallon, J. Leonard, and J. McDonald, "Real-time large-scale desne RGB-D SLAM with volumetric fusion," *The Intl. J. of Robotics Research*, Vol. 34, No. 4-5, pp. 598-626, 2015.

[7] L. Ma, T. Whelan, E. Bondarev, P. H. N. de With, and J. McDonald, "Planar simplification and texturing of dense point cloud maps," *Proc. 2013 European Conf. on Mobile Robots*, pp. 25-27, 2013.

[8] S. Liu, W. Li, P. Ogunbona, and Y. Chow, "Creating simplified 3D Models with high quality textures," *Proc. 2015 Intl. Conf. on Digital Image Computing; Techniques and Applications*, 2015.

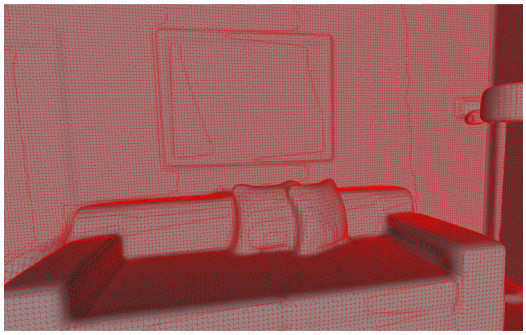
[9] Q. Zhou and V. Koltun, "Color map optimization for 3D reconstruction with consumer depth cameras," *ACM Transactions on Graphics*, Vol. 33, Issue 4, Article No. 155, 2014.

[10] A. Handa, T. Whelan, J. McDonald and A. J. Davison, "A Benchmark for RGB-D Visual Odometry, 3D Reconstruction and SLAM," *Proc. IEEE Int. Conf. Robot. Autom.,* pp. 1524-1531, 2014.

[11] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker and A. J. Davison, "ElasticFusion: Dense SLAM Without A Pose Graph," *Proc. robotics Science and systems (RSS),* 2015.

[12] J. An, J. Lee, J. Jeong, and I. Ihm, "Tracking an RGB-D camera on mobile devices using an improved frame-to-frame pose estimation method," *Proc. IEEE Winter Conf. on Applications of Computer Vision 2018*, pp. 1142-1150, 2014.

(a) Point cloud

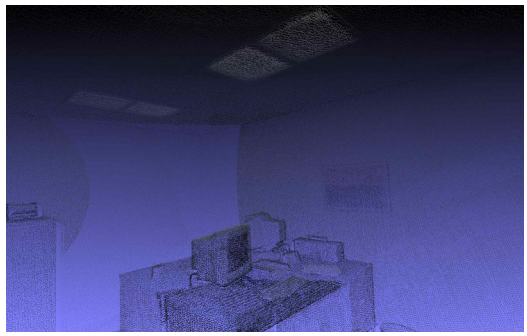


(b) Triangular mesh shown with wireframe

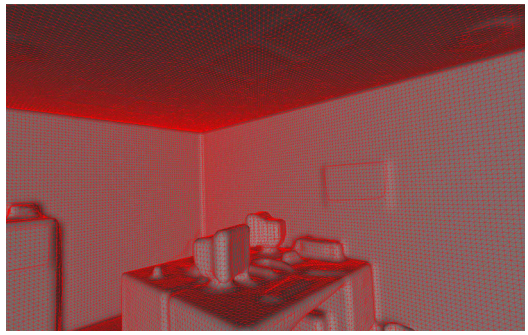
(c) Naive method

(d) Our method

Figure 8. Experimental result (Living room)



(a) Point Cloud

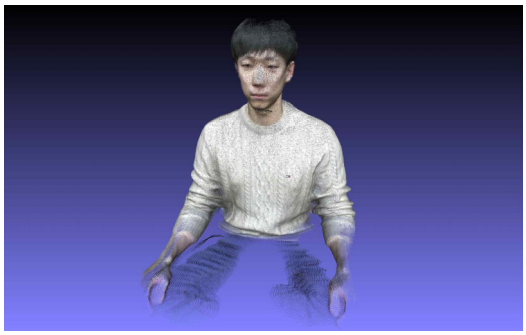


(b) Triangular mesh shown with wireframe

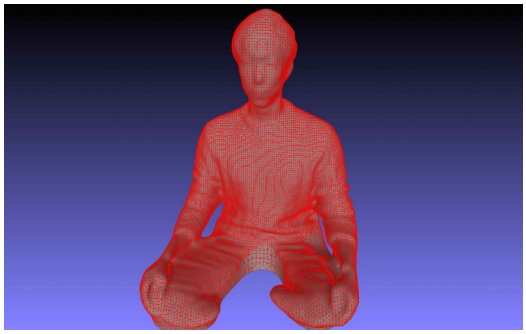
(c) Naive method

(d) Our method

Figure 9. Experimental result (Office room)



(a) Point Cloud



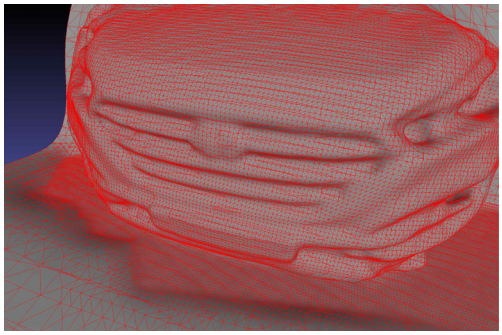
(b) Triangular mesh shown with wireframe

(c) Naive method

(d) Our method

Figure 10. Experimental result (Man)

(a) Point Cloud



(b) Triangular mesh shown with wireframe

(c) Naive method

(d) Our method

Figure 11. Experimental result (Car-grille)

〈저자소개〉

서 웅

- 2012년 한국외국어대학교 전자공학과 학사
- 2014년 서강대학교 컴퓨터공학과 석사
- 2014년~현재 서강대학교 컴퓨터공학과 박사과정
- 관심분야 : 컴퓨터 그래픽스, 모바일 렌더링, GPGPU

박 상 욱

- 2014년~현재 서강대학교 컴퓨터공학과 학사과정
- 관심분야 : 컴퓨터 그래픽스

임 인 성

- 1985년 서울대학교 자연과학대학 계산통계학과 학사
- 1987년 Rutgers 대학교 컴퓨터과학과 석사
- 1991년 Purdue 대학교 컴퓨터과학과 박사
- 1993년~현재 서강대학교 공과대학 컴퓨터공학과 교수
- 관심분야 : 컴퓨터 그래픽스, 고성능 계산, 과학적 가시화