

일반논문 (Regular Paper)

방송공학회논문지 제25권 제3호, 2020년 5월 (JBE Vol. 25, No. 3, May 2020)

<https://doi.org/10.5909/JBE.2020.25.3.439>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

## 다시점 RGB-D 카메라를 이용한 실시간 3차원 체적 모델의 생성

김 경 진<sup>a)</sup>, 박 병 서<sup>a)</sup>, 김 동 옥<sup>a)</sup>, 권 순 철<sup>b)</sup>, 서 영 호<sup>a)†</sup>

### Real-time 3D Volumetric Model Generation using Multiview RGB-D Camera

Kyung-Jin Kim<sup>a)</sup>, Byung-Seo Park<sup>a)</sup>, Dong-Wook Kim<sup>a)</sup>, Soon-Chul Kwon<sup>b)</sup>, and Young-Ho Seo<sup>a)†</sup>

#### 요 약

본 논문에서는 다시점 RGB-D 카메라의 포인트 클라우드 정합을 위한 수정된 최적화 알고리즘을 제안한다. 일반적으로 컴퓨터 비전 분야에서는 카메라의 위치를 정밀하게 추정하는 것은 매우 중요하다. 기존의 연구에서 제안된 3D 모델 생성 방식들은 많은 카메라 대수나 고가의 3차원 Camera를 필요로 한다. 또한 2차원 이미지를 통해 카메라 외부 파라미터를 얻는 방식들은 큰 오차를 가지고 있다. 본 논문에서는 저가의 RGB-D 카메라를 8개 사용하여 전방위 자유시점을 제공할 수 있는 3차원 포인트 클라우드 및 매쉬 모델을 생성하기 위한 정합 기법을 제안하고자 한다. RGB영상과 함께 깊이지도 기반의 함수 최적화 방식을 이용하고, 초기 파라미터를 구하지 않으면서 고품질의 3차원 모델을 생성할 수 있는 좌표 변환 파라미터를 구하는 방식을 제안한다.

#### Abstract

In this paper, we propose a modified optimization algorithm for point cloud matching of multi-view RGB-D cameras. In general, in the computer vision field, it is very important to accurately estimate the position of the camera. The 3D model generation methods proposed in the previous research require a large number of cameras or expensive 3D cameras. Also, the methods of obtaining the external parameters of the camera through the 2D image have a large error. In this paper, we propose a matching technique for generating a 3D point cloud and mesh model that can provide omnidirectional free viewpoint using 8 low-cost RGB-D cameras. We propose a method that uses a depth map-based function optimization method with RGB images and obtains coordinate transformation parameters that can generate a high-quality 3D model without obtaining initial parameters.

Keyword : Registration, RGB-D Camera, Point Cloud, 3D, Camera Pose Estimation

a) 광운대학교 전자재료공학과(Kwangwoon university Electronic Materials Engineering)

b) 광운대학교 스마트시스템학과(Kwangwoon university Graduate School of Smart Convergence)

† Corresponding Author : 서영호(Young-Ho Seo)

E-mail: yhseo@kw.ac.kr

Tel: +82-2-940-8362

ORCID: <http://orcid.org/0000-0003-1046-395X>

※ 이 논문은 2019년도 한국콘텐츠진흥원의 문화기술연구개발사업 지원에 의한 연구임(R2019050033\_000000011210820867720101351). (This research was supported by Cultural Technology R&D Project through the Korea Creative Content Agency(R2019050033\_000000011210820867720101351)).

· Manuscript received March 26, 2020; Revised May 4, 2020; Accepted May 4, 2020.

## I. 서론

최근 3차원 그래픽스 기술과 실사 기술이 혼합된 가상현실(Virtual Reality, VR), 증강현실(Augmented Reality, AR), 혼합현실(Mixed Reality, MR), 그리고 확장현실(Extended Reality, XR) 기술 등이 동시에 발전하고 서로 융합하는 새로운 멀티미디어 시대가 열리고 있다. 이러한 환경에서는 사용자의 위치와 시점에 상관없이 360도의 자유시점형 체험이 기본적으로 제공되어야 하기 때문에 실사 데이터에 기반한 전방위 3차원 모델은 필수적인 것이 되어가고 있다. 마이크로소프트 연구팀이 2011년 KinectFusion<sup>[1]</sup>을 발표한 이후로 저가의 상용 RGB-D 카메라 여러 대를 사용하여 전방위 3차원 모델을 생성하는 연구가 활발히 진행되고 있다<sup>[2]</sup>. 다중 RGB-D 카메라를 이용한 3차원 모델 생성에 앞서 각 카메라에서 획득한 객체의 포인트 클라우드(point cloud)들을 하나의 좌표계로 통합하는 과정이 필요하다<sup>[2]</sup>. 이 과정을 포인트 클라우드 정합이라고 한다.

가장 잘 알려진 포인트 클라우드 정합 알고리즘으로는 ICP(Iterative Closest Point) 알고리즘이 있다<sup>[3]</sup>. ICP는 입력된 두 포인트 세트 간에 미리 정의된 겹치는 영역에 대해 가장 가까운 거리에 있는 포인트 쌍을 찾고, 반복적인 연산을 통해 이들의 거리를 최소화하는 좌표변환 파라미터를 구하는 방법이다. ICP와 비슷한 방식으로 포인트 세트의 반복적인 연산을 통해 정합하는 SoftAssign 알고리즘<sup>[4]</sup>과 이에 기반한 다양한 변형 알고리즘<sup>[5][6][7]</sup>들이 연구되었다<sup>[8]</sup>. 이 연구들은 파라미터의 초기값, 포인트 클라우드 사이의 겹침 영역의 크기 등에 많이 의존하며 국부 최소화(local minima)라는 단점을 갖는다<sup>[8]</sup>. 포인트 세트를 이용한 다른 정합 방식으로 특정 기하학 특성을 이용한 PCA(Principal Component Analysis) 정렬, 모달(modal) 그리고 스펙트럼 정합(spectral matching)과 같은 방법들이 연구되어 왔다<sup>[9][10]</sup>. 이 방법들 또한 포인트 세트가 어떻게 구성되어 있는가에 의존을 하는 단점이 있기 때문에<sup>[8]</sup> RGB-D 카메라로부터 획득된 깊이 값에 잡음이 많고 각 카메라 사이에 겹치는 영역이 적은 시스템에서는 적용하기는 어렵다.

포인트 클라우드 정합을 위해서는 각 카메라들의 좌표를 변환해야 한다. 포인트 클라우드 정합을 위한 좌표변환 행렬을 구하는 방식은 다양한 방법이 연구되어 왔다<sup>[11][12]</sup>. 많

이 사용되는 좌표변환 행렬 추출 기법으로 Zhang 알고리즘<sup>[11]</sup>이 있다. 이 알고리즘은 포인트 세트에 의존하지 않고, 특징점을 추출하기 쉬운 차르코보드 등을 이용한다. 하지만 이 방식은 핀홀 카메라 모델에 기반하여 내부 및 외부 파라미터를 추정하기 때문에 실제 카메라에 적용하게 되면 필연적으로 오차가 발생하게 된다<sup>[12]</sup>. 또한 깊이 카메라 기준이 아닌 RGB 카메라를 이용한 카메라 위치추정기반의 좌표변환 행렬을 구하는 방식이기 때문에 깊이 카메라의 좌표계를 기준으로 획득한 3차원 형상 정보를 직접 변환하는 데 사용할 수 없다<sup>[2]</sup>. ICP 알고리즘 등과 같이 포인트 클라우드 정합 결과가 초기 파라미터와 포인트 세트 구성에 의존적이라는 문제와 RGB 이미지를 이용한 카메라 자세 추정 방식에서 구한 좌표변환 행렬은 큰 오차를 갖는다는 문제점을 극복하기 위해 RGB 이미지에서 초기 파라미터를 계산하여 각각의 카메라에서 획득한 포인트 클라우드 중 같은 공간에 위치한 포인트를 이용하여 반복적인 연산을 통해 최적화된 좌표변환 파라미터를 찾는 알고리즘이 연구되었다<sup>[13]</sup>. 본 논문에서는 [13]에서 제안한 방식을 개선하여 초기 파라미터를 계산하지 않으면서 차르코 보드와 같은 특정 방식에 제한되지 않는 수정된 함수 최적화 기법을 제안한다. 이를 통해서 연산의 복잡도와 연산량을 줄이면서도 좋은 품질의 3D 모델을 생성할 수 있음을 보인다. 또한 특정 방식에 제한되지 않고 특징점이 추출되는 모든 영상에 대해 적용 가능하므로 범용성이 향상되었다.

본 논문은 다음과 같이 구성된다. 2장에서는 3차원 모델을 생성하기 위한 카메라 시스템과 실사 3차원 모델을 제작하는 과정을 소개하고 3장에서는 제안하는 포인트 클라우드 정합 알고리즘을 소개한다. 4장에서는 실험을 통해 측정된 정합 오차와 분석 결과를 보이고 5장에서 본 논문을 마무리한다.

## II. 3차원 모델의 생성

실사 기반의 3D 모델을 생성하기 위해 깊이 및 RGB 센서를 장착한 RGB-D 카메라를 사용하였다. 3D 모델을 생성하는 것이 목표이므로 8대의 RGB-D 카메라를 물체의 여

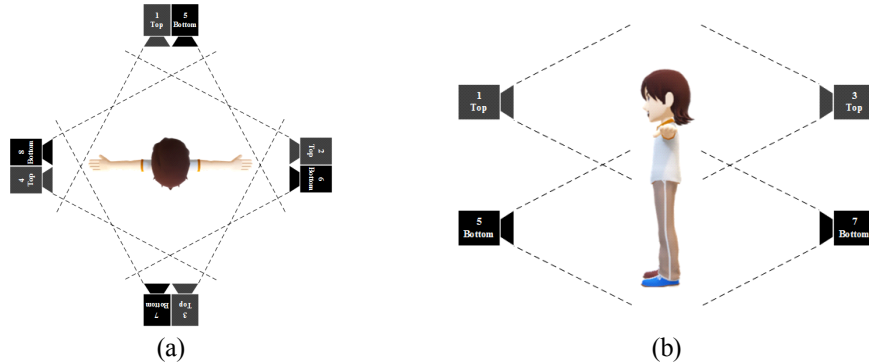


그림 1. 실사 3D 체적 촬영 시스템 (a) 수직, (b) 수평 방향에서의 촬영 각도 및 범위  
Fig. 1. Real 3D point cloud capturing system (a) vertical, (b) horizontal photographing angle and range

러 시점에 설치했다. 3차원 모델을 생성하기에 앞서 RGB-D 카메라를 통해 촬영되는 깊이 및 RGB 이미지를 이용하여 깊이 카메라의 좌표계를 따르는 포인트 클라우드를 각 카메라에서 획득한다. 8대의 RGB-D 카메라의 배치는 물체를 모든 높이에서 촬영하기 위해 위아래에 카메라가 설치된 스탠드 형태의 촬영 장비를 사용하여 구성했다. 그리고 물체를 모든 방향에서 촬영하기 위해 4세트의 스탠드를 앞, 뒤, 양옆 네 방향에 배치하였다. 그림 1은 본 논문에서 제안하는 카메라 시스템이다. 그림 1(a)은 수직 방향의 촬영 범위를 나타내고 그림 1(b)은 수평 방향의 촬영 범위를 나타낸다.

RGB-D 카메라를 사용하여 3차원 모델을 생성할 때 깊이 센서의 부정확성으로 인해 발생하는 깊이지도의 잡음성분을 최소화하는 작업이 필요하다. 잡음을 제거하기 위해 전처리 과정으로 깊이 카메라의 출력에 다양한 필터링을 수행하는 것이 일반적이다<sup>[13]</sup>. 그러나 좋은 품질을 얻기 위해서 필터링을 과도하게 적용하면 모션블러(motion blur)와 고스트(ghost) 현상이 생길 수 있다. 포인트 클라우드에서

모션블러와 고스트 현상은 시각적으로 큰 문제로 관찰되지 않으나 이것이 매쉬로 변환되면 시각적으로 큰 영향을 미친다. 또한 실시간으로 모델을 생성하는 경우에 프레임과 프레임 사이에 지연이 증가하기 때문에 필터링을 통해 원하는 만큼의 잡음 성분을 제거하기 어렵다. 이러한 이유로 촬영시 필터링은 최소한으로 적용하고, 촬영된 포인트 클라우드에 후처리로 샘플링 및 스무딩 작업을 추가함으로써 잡음이 감소된다<sup>[14]</sup>.

## 1. 실사 기반 3차원 메쉬 모델 시퀀스 생성

메쉬 모델 시퀀스를 생성하기 위해 먼저 각 카메라에서 깊이 및 RGB 이미지를 촬영하여 포인트 클라우드를 획득한다. 다음 각 카메라에서 획득한 포인트 클라우드를 정합하여 전방위의 포인트 클라우드를 생성한다. 정합된 모든 프레임의 포인트 클라우드는 샘플링 한 뒤 들로네(Delaunay) 삼각분할<sup>[15]</sup>과 같은 방식을 이용하여 삼각형 메쉬를 생성하여 저장한다. 그림 2는 다중 RGB-D 카메라를 이용해

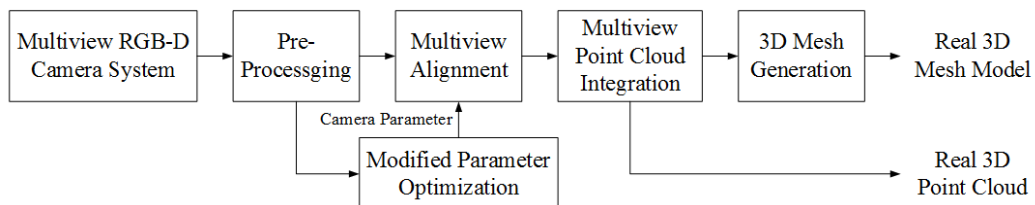


그림 2. 실사 기반 3차원 모델 생성 절차  
Fig. 2. Real 3D model generation procedure

실사 기반의 3차원 메쉬 모델 시퀀스를 얻기 위한 과정을 나타낸다.

### III. 제안한 정합 알고리즘

본 논문에서 제안한 포인트 클라우드 정합 방식은 깊이 및 RGB 이미지 모두를 이용하고, 최종 정합된 모델의 좌표계는 기준으로 설정한 카메라 좌표계를 따르게 된다. 8대의 RGB-D 카메라를 이용하여 전방위 3차원 모델을 생성하기 위해 먼저 각 카메라에서 차르코보드를 촬영한 RGB 이미지를 얻는다. 이 이미지를 이용하여 차르코보드 내부 코너의 픽셀 좌표를 얻는다. 그 다음 깊이 이미지를 이용하여 카메라마다 포인트 클라우드를 생성한다. 획득한 포인트 클라우드에서 저장해 둔 차르코보드의 내부 코너 픽셀 좌표의 3차원 좌표들만을 획득하여 반복적인 연산을 통해 이 좌표들 사이의 거리가 최소화가 되는 좌표변환 파라미터를 구한다.

#### 1. 포인트 클라우드 정합을 위한 좌표변환 파라미터 획득

아래의 내용은 카메라끼리 매칭점을 이용하여 포인트 클라우드 정합을 위한 좌표변환 파라미터를 획득하는 방법에 대해 설명한다. 매칭되는 포인트를 찾기 위해 QR코드와

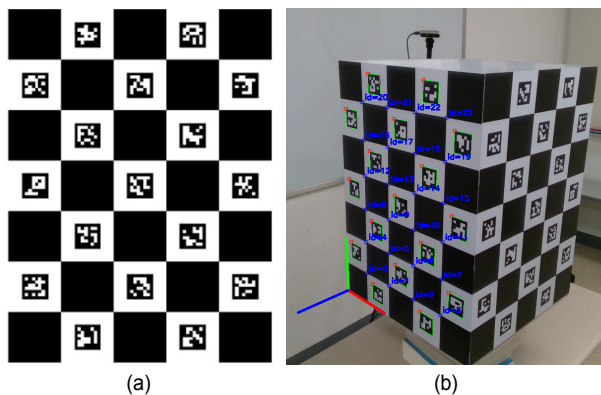


그림 3. (a) 차르코 보드, (b) 차르코 보드를 통해 구한 월드 좌표계  
Fig. 3. (a) Charuco board, (b) World coordinate system obtained through Charuco board

체스보드를 합쳐서 만든 차르코 보드(Charuco board)를 사용한다<sup>[16]</sup>. 일반 체스보드 패턴만을 이용해서 구할 수도 있지만 QR코드를 이용하면 일반 체스보드를 이용한 방식들보다 빠르게 보드 내부의 코너 좌표를 찾을 수 있기 때문에 체스보드가 아닌 차르코 보드를 사용했다<sup>[11][16]</sup>. 내부 코너를 찾는 것은 RGB 영상에서 진행했다. 그림 3(a)는 차르코 보드이고 그림 3(b)는 차르코 보드에서 내부코너를 추정하고 이것을 이용하여 구한 월드 좌표계를 표시한 이미지이다.

RGB 영상에서 검출한 차르코 보드 내부 코너에 대해 3차원 좌표를 구하기 위해, 깊이 영상과 RGB 영상간의 캘리브레이션(calibration)을 진행한다<sup>[17]</sup>. 그리고 깊이영상에서 내부 코너의 픽셀값에 해당 하는 위치의 깊이 값을 이용하여 3차원 좌표를 구한다.

본 논문에서는 빠른 동작을 위해서 차르코 보드를 사용하고 있지만 제안하는 방식은 반드시 차르코 보드를 사용해야 하는 것은 아니고, 영상들 간의 매칭할 수 있는 동일 좌표만 찾으면 되기 때문에 특징점을 추출하는 다른 알고리즘을 사용하여도 결과의 차이는 없다.

#### 2. 제안한 최적화 알고리즘

앞서 언급한 3차원 좌표를 8대의 모든 카메라에서 획득하고 매칭되는 포인트를 찾는다. 매칭되는 좌표끼리 같은 위치에 위치하도록 하는 좌표변환 파라미터를 구하는 방법을 제안한다. 이 파라미터는 최적화 알고리즘을 이용하여 오차가 최소가 되도록 계산한다. 최적화 문제를 풀기 위한 알고리즘으로는 경사 하강법(gradient descent) 방법, 관성 하강법(gradient descent with momentum), 네스테로프 하강법(Nesterov's gradient method)<sup>[18]</sup> 등이 있지만 구하고자 하는 파라미터의 수가 많지 않기 때문에 가장 계산비용이 적고 구현이 쉬운 경사 하강법을 사용했다.

좌표변환행렬에는 x, y, z 축 각각의 회전각과 평행이동값, 총 6개의 파라미터가 포함되어 있고 8대의 카메라 중 한 대의 카메라 좌표를 기준 좌표계로 하여 해당 좌표계로 변환되는 파라미터를 구한다.  $X_{ref}$ 는 기준 카메라 좌표를 나타내고  $X_i$ 는 나머지 카메라의 좌표를 나타낸다.  $R_{i \rightarrow ref}$ 와  $t_{i \rightarrow ref}$ 은 각각 식 기준 카메라 좌표로의 회전변환 행렬과

평행이동 행렬을 나타낸다. 초기  $R_{i \rightarrow ref}$ 은 단위행렬이고  $t_{i \rightarrow ref}$ 은 모두 0인 값으로 되어있다. 식 (1)은 초기 파라미터를 적용하게 되면 결과 값이  $X_i$ 가 되지만, 최적화를 진행하면서  $X_{ref}$ 에 수렴하게 된다.

$$X'_i = R_{i \rightarrow ref} X_i + t_{i \rightarrow ref} \quad (1)$$

최적화를 진행할 오차 함수는  $X_{ref}$ 와  $X'_i$ 의 제곱 유클리드 거리(Squared Euclidean Distance, SED)의 평균값이다. 식 (2)은 오차 함수를 나타낸다.

$$f_{Error} = \frac{1}{N} \sum_{j=0}^N \| X_{ref(j)} - X'_i(j) \|^2 \quad (2)$$

이 함수를 좌표변환 파라미터에 대해 미분하여 함수값이 최소가 되는 방향으로 파라미터를 갱신해 나가는 과정을 식 (3)과 같이 나타낼 수 있다.  $\alpha$ 는 학습률을 나타내는 상수로 본 실험에서는 0.01로 두었다.

$$P_{n+1} = P_n - \alpha \frac{\partial f_{Error}}{\partial P_n} \quad (3)$$

본 논문에서 제안하는 알고리즘에서 첫 번째 단계는 정합에 이용할 데이터를 얻기 위해 다중 RGB-D 카메라에서 깊이 및 RGB 이미지를 출력하는 것이다. RGB 이미지는 차르코 보드 내부 코너를 검출하는 “Charuco Board Corner Detection” 단계에서 사용된다. 깊이 영상은 보다 더 정확한 3차원 좌표를 얻기 위해 필터링을 통해 노이즈를 줄이는 단계인 “Noise Reduction”과정을 거친다. RGB-D 카메라에서 RGB 카메라와 깊이 카메라는 동일한 위치가 아닌 수평 방향으로 떨어져 있다. 따라서, 해당 코너좌표의 정확한 깊이 값을 가져오기 위해서 깊이 이미지와 RGB 이미지의 좌표를 일치시켜주기 위한 “RGB-D Calibration” 단계를 거쳐야 한다. 그리고 깊이 이미지를 이용하여 차르코 보드의 내부 코너들의 3차원 좌표를 획득하는 “3D Coordinate Calculation”을 진행한다. 최적화 단계인 “Iterative Operation for Minimum SED”는 식 (2)을 통해 현재 변환된 좌표

의 오차를 구하여 식 (3)을 이용하여 오차가 줄어드는 방향으로 반복적으로 연산하여 파라미터를 업데이트한다. 이 과정을 통해 제곱 유클리드 거리가 최소가 되는 최종 좌표 변환 파라미터를 얻는다. 그림 4는 본 논문에서 제안한 포인트 클라우드 정합 알고리즘의 흐름도이다.

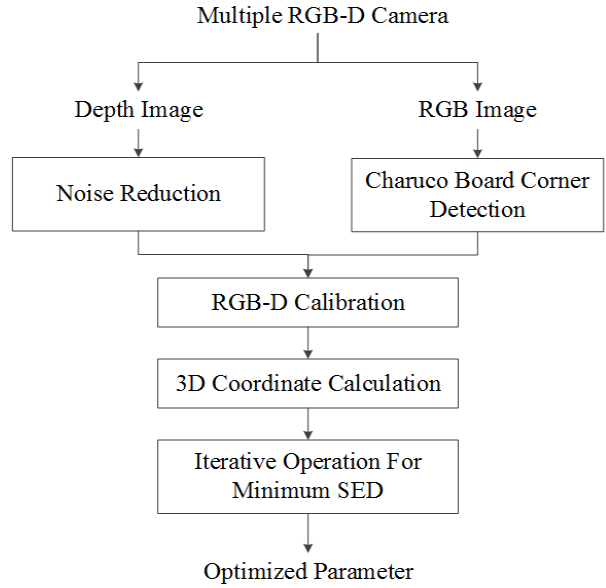


그림 4. 파라미터 최적화 흐름도  
Fig. 4. Parameter optimization Flowchart

## IV. 실험결과

본 실험에서는 8대의 Microsoft Kinect Azure 카메라를 사용하여 실험을 진행했다. 카메라 배치는 그림 1에서 설명한 촬영 시스템을 따른다. 8대 중 4대는 물체의 하단 부분을 촬영할 수 있도록 지면에서 0.7m되는 높이에 설치되었고 나머지 4대는 물체의 상단 부분을 촬영할 수 있도록 지면에서 1.7m되는 높이에 설치되었다. 깊이 값에 임계값을 설정하여 0.1m에서 2.5m내의 물체에 대한 포인트 클라우드를 획득할 수 있도록 하였다. 그림 5는 실제 구성한 카메라 시스템을 촬영한 것이다.

8대의 RGB-D 카메라는 각 카메라의 깊이 카메라 좌표계를 따르는 포인트 클라우드를 평균적으로 초당 30 프레임



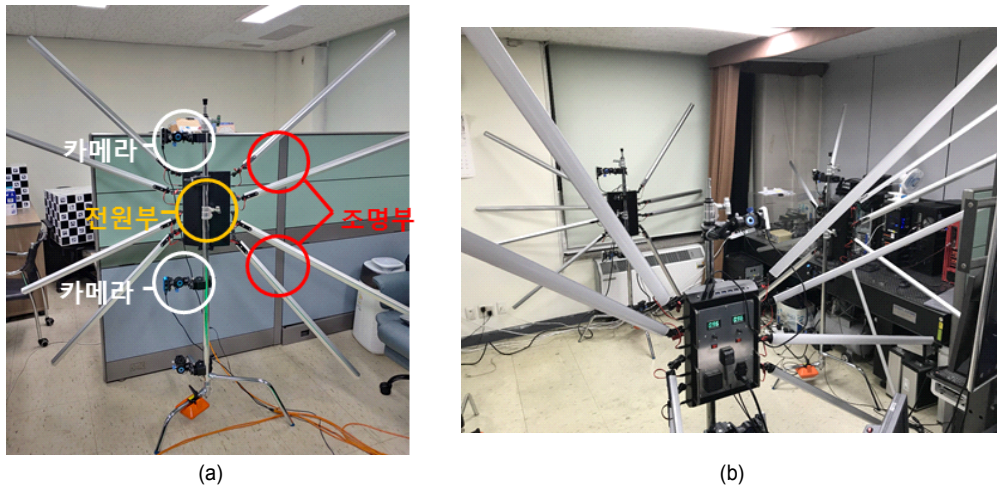


그림 5. 카메라 시스템 (a) 카메라 시스템 구성 (b) 전체 카메라 시스템  
Fig. 5. Camera system (a) camera system parts (b) Full camera system

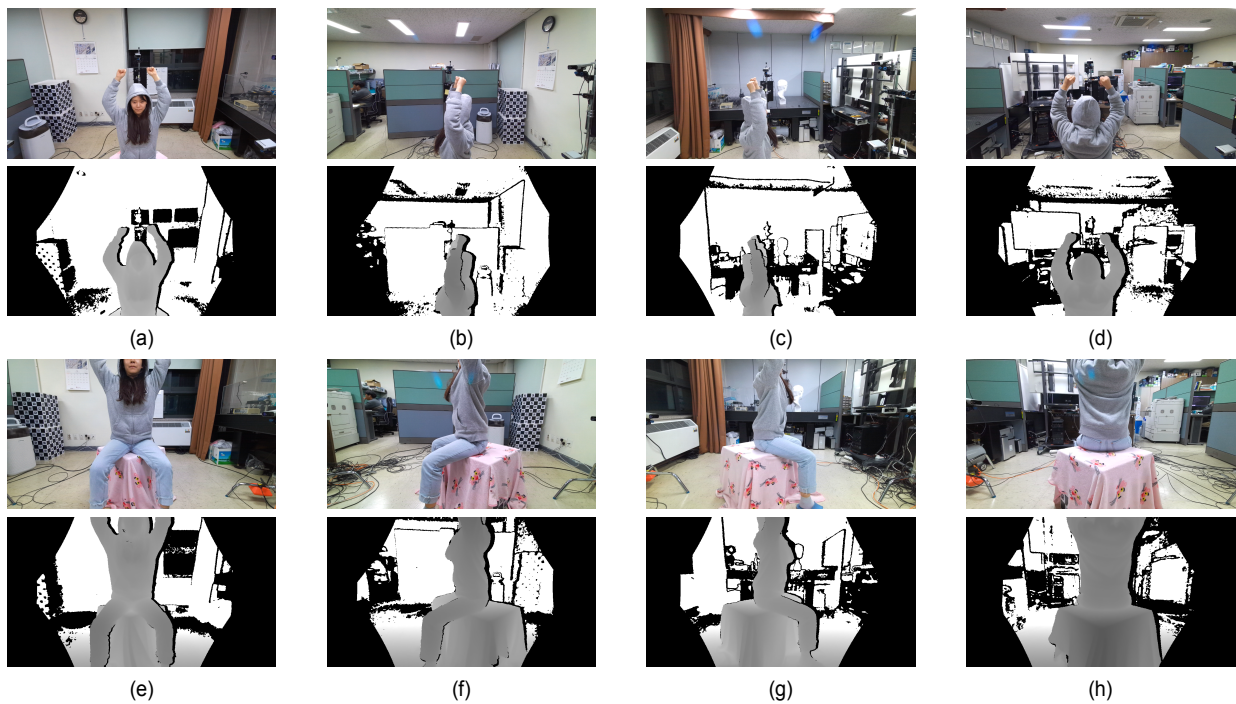


그림 6. 각 카메라에서 획득한 포인트 클라우드, 위쪽에 배치된 카메라 중 (a) 앞쪽, (b) 왼쪽, (c) 오른쪽, (d) 뒤쪽, 아래쪽에 배치된 카메라 중 (e) 앞쪽, (f) 왼쪽, (g) 오른쪽, (h) 뒤쪽  
Fig. 6. Point cloud acquired from each camera, (a) front, (b) left, (c) right, (d) rear of the cameras placed above, (e) front, (f) left, (g) right, (h) rear of the placed cameras below.

이상 실시간으로 출력할 수 있다. 그림 6은 정합 전 각 카메라에서 획득한 포인트 클라우드이다. 그림 6(a), (b),

(c), (d)는 위에 배치된 카메라 중 각각 앞쪽, 왼쪽, 오른쪽, 뒤쪽의 카메라로 촬영한 포인트 클라우드이고 그림 6(e),

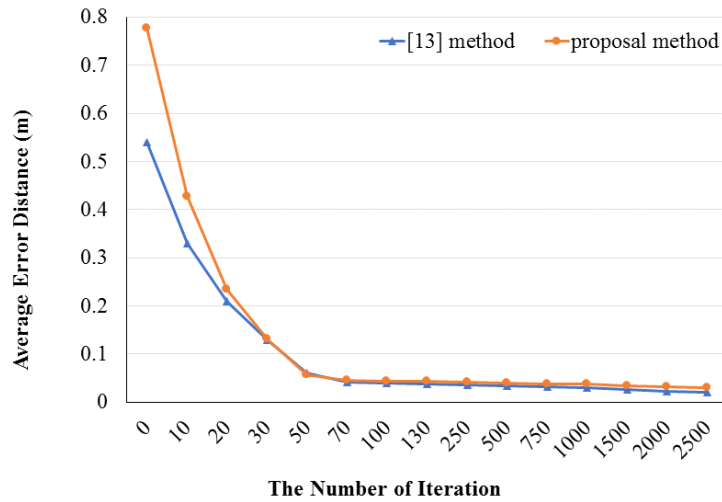


그림 7. 반복횟수에 따른 평균 오차 거리에 대한 그래프  
Fig. 7. Graph of average error distance according to iteration

(f), (g), (h)는 아래 배치된 카메라로 촬영한 포인트 클라우드로이다.

파라미터의 오차는 차르코 보드의 내부코너 좌표의 제곱 유클리드 거리를 통해 판단하였다. 연산은 전의 오차와 0.0000000001의 차이를 가질 때까지 반복했다. 카메라마다 다르긴 했지만, 평균적으로 연산을 100번 가량 반복할 때까지는 빠르게 값이 줄어들다가 그 이후로는 점점 0에 수렴되는 것을 실험을 통해 확인하였다. 반복횟수는 50,000회에서 200,000회까지 카메라마다 다른 값을 가졌다. 제안한 방식도 마찬가지로 0.0000000001의 차이를 가질 때까지 반복연산을 수행했고, 반복횟수 또한 200,000회 이하의 모두 다른 값을 가졌다. 그림 7은 2500번까지의 반복횟수에 따른 평균 오차 거리에 대한 그래프를 나타낸 것이고, [13]

방식과 제안하는 방식에 대해서 비교하였다.

Camera1을 기준 카메라로 설정했기 때문에 최종 모델 좌표계는 Camera1의 깊이 카메라의 좌표계를 따른다. 최적화 전의 파라미터로 정합했을 때 평균적으로 59cm의 오차를 가지고 있었지만, 최적화 알고리즘을 적용하게 되면 평균 3mm의 오차를 가진다. 표 1은 최적화 전의 정합 오차값, [13]에서 제안하는 방식의 평균 정합 오차값 그리고 본 논문에서 제안하는 방식의 오차값을 비교한 표이다. [13]의 결과와 비교할 때 오차의 평균값이 0.08mm 높아진 결과를 나타냈으나 매우 작은 값이라서 시각적으로는 거의 영향을 주지 않는다. 또한 전체적으로 제안하는 방식이 오차의 분포가 더 균일한 특성을 나타내었다.

그림 8은 최적화 알고리즘 적용 전후의 다시점 RGB-D

표 1. 포인트 클라우드의 평균 정합 오차  
Table 1. Average registration error of point cloud

	Parameter		Camera							Avr.
			2	3	4	5	6	7	8	
[13]	Initial (cm)		69.9	56.5	58	52.6	53.3	58.8	61.9	58.7
	Optimization (mm)	Value	3.6	1.3	4.8	2.8	1.2	2.2	4.1	2.9
		Value-Avr.	0.7	1.6	1.9	0.1	1.7	0.7	1.2	7.9
Proposed	Optimization (mm)	Value	3.87	2.14	3.10	3.4	2.23	3.24	2.87	2.98
		Value-Avr.	0.89	0.84	0.12	0.42	0.75	0.26	0.11	3.39

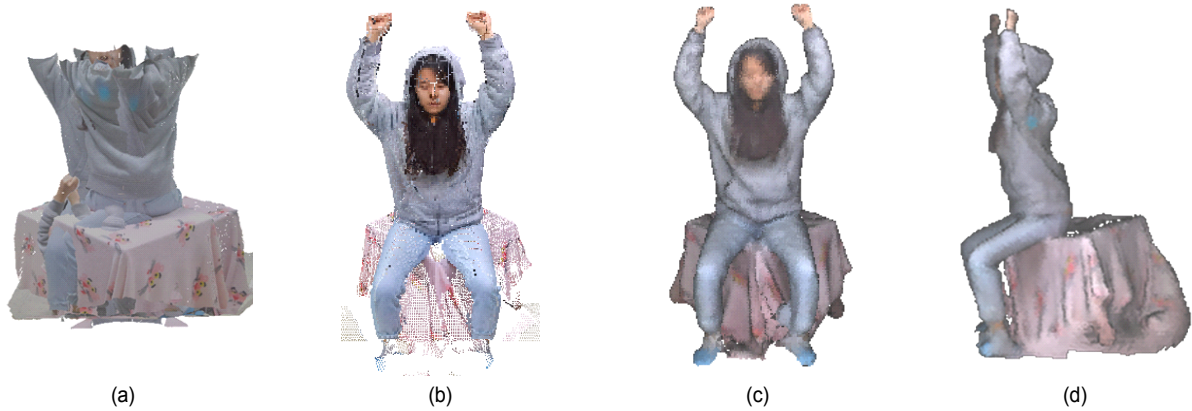


그림 8. 제안한 최적화 알고리즘 적용 전후의 전방위 포인트 클라우드 및 메쉬 3D 모델 (a) 정렬되기 이전의 포인트 클라우드 모델, (b) 최적화된 파라미터를 이용하여 정렬한 포인트 클라우드, (c) 메쉬 모델로 변환한 결과(시점1) (d) 메쉬 모델로 변환한 결과(시점2)

Fig. 8. The omnidirectional point cloud before and after the proposed optimization (a) the point cloud model before alignment (b) the point cloud after alignment using the optimized parameter, (c) the converted mesh model (viewpoint 1), (d) the converted mesh model (viewpoint 2)

카메라에서 획득한 포인트 클라우드를 정합한 결과에 대한 그림이다. 그림 8은 본 실험에서 구한 좌표변환 파라미터를 이용하여 정합한 결과이다. 그림 8(a)는 정합 전의 포인트 클라우드이고, (b)는 제안하는 알고리즘으로 구한 파라미터로 정합한 포인트 클라우드이다. (c)와 (d)는 다른 시점에서의 메쉬 모델이다.

간이 적었음에도 국부 최소화 문제를 극복할 수 있었다. 그리고 특징점을 사용함으로써 연산과정을 줄이고 범용성이 향상되었지만 정합 결과에 대한 오차는 크게 달라지지 않는 것을 확인할 수 있었다.

## V. 결 론

본 논문에서는 다시점 RGB-D 카메라를 이용해 전방위 3차원 모델을 생성하기 위해 필수적인 포인트 클라우드 정합 알고리즘을 제안하였다. 차르코 보드를 이용한 초기 파라미터를 구하는 방식을 제거하고 특징점을 이용하여 최적화 하는 방식을 사용했다. 이 방식을 사용함으로써 특징점이 추출되는 물체만 있다면 포인트 클라우드 정합을 진행할 수 있기 때문에 범용성을 높일 수 있었다. [13]의 알고리즘을 이용한 정합 결과는 평균 2.9mm의 오차를 가졌지만, 본 논문에서 제안하는 방식에서 약 2.98mm 정도의 오차를 가졌다. 카메라 사이의 오차의 분포가 고르게 나타나서 오차가 한 곳에 집중되지 않고 골고루 분산됨으로써 시각적으로 더 좋은 결과를 나타냈다. 그리고 RGB 이미지에서 특징점을 추출하여 서로 다른 시점의 카메라에 공간상의 같은 위치를 알려주었기 때문에 카메라 사이에 겹치는 공

## 참 고 문 헌 (References)

- [1] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, A. Fitzgibbon, "KinectFusion: Real-Time Dynamic 3D Surface Reconstruction and Interaction Using a Moving Depth Camera" ISMAR, Vol. 11, No. 2011, pp. 127-136, October 2011.
- [2] S. Choi, S. Park, "Convenient View Calibration of Multiple RGB-D Cameras Using a Spherical Object." KIPS Transactions on Software and Data Engineering, Vol.3 No.8, pp.309-314, 2014.
- [3] C. Gregory, W. Sang, K. David, "ICP Registration Using Invariant Features.", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, Vol. 24, No. 1, pp. 90-102, January 2002.
- [4] S. Gold, A. Rangarajan, C. Lu, S. Pappu, E. Mjolsness, "New algorithms for 2d and 3d point matching: Pose estimation and correspondence." In Pattern Recognition, Journal of KIISE, Vol. 31, pp. 1019 - 1031, August 1998.
- [5] S. Granger, X. Pennec. "Multi-scale em-icp: A fast and robust approach for surface registration." In ECCV, pp. 418 - 432, June 2002.
- [6] P. David, D. DeMenthon, R. Duraiswami, H. Samet. "Simultaneous pose and correspondence determination using line features." In CVPR, pp. 424 - 431, June 2003.
- [7] W. Zhao, D. Nister, S. Hus. "Alignment of continuous video onto 3d point clouds." PAMI, Vol. 27, pp.1305 - 1318, August 2005.



- [8] Li, Hongdong, R. Hartley. "The 3D-3D Registration Problem Revisited." 2007 IEEE 11th International Conference on Computer Vision, pp. 1-8, 2007
- [9] S. Sclaroff, A. Pentland. "Modal matching for correspondence and recognition." In PAMI, June 1995.
- [10] M. Leordeanu, M. Hebert. "A spectral technique for correspondence problems using pairwise constraints." In ICCV, Vol. 2, pp. 1482 - 1489, 2005
- [11] Z. Zhang, "A flexible new technique for camera calibration," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.22, No.11, pp.1330-1334, 2000.
- [12] J. Lee, "Camera calibration and Compensation distortion", Korea robotics society review, Vol. 10 No, 1, pp. 23-29, February 2013
- [13] K.-J Kim, B.-S. Park, D.-W. Kim, Y.-H. Seo, "Point Cloud Registration Algorithm Based on RGB-D Camera for Shooting Volumetric Objects.", JBE Vol. 24, No. 5, pp. 765-774, September 2019.
- [14] B. Lin, M. Su, P. Cheng, P. Tseng, S. Chen "Temporal and Spatial Denoising of Depth Maps", Sensors 2015 Vol. 15, No. 8, pp. 18506-18525, August 2015.
- [15] J. Digne, C. Franchis, "The Bilateral Filter for Point Clouds", Image Processing on Line Vol. 2017, No. 7, pp.278-287, March 2018.
- [16] T. Fang, L. A. Piegl. "Delaunay triangulation in three dimensions." IEEE Computer Graphics and Applications, Vol. 15, No. 5, pp. 62-69, September 1995
- [17] J. Kim, S. Yoo, K. Min. "Microsoft Kinect-based Indoor Building Information Model Acquisition." Computational Structural Engineering Institute of Korea 31.4 (2018): 207-214.
- [18] S. Lee, "Convergence Rate of Optimization Algorithms for a Non-strictly Convex Function", Institute of Control Robotics and Systems, pp. 349-350, May 2019.

---

저 자 소 개

---



**김 경 진**

- 2019 2월 : 광운대학교 전자재료공학과 졸업(공학사)
- 2019 3월 ~ 현재 : 광운대학교 전자재료공학과(석사과정)
- ORCID : <https://orcid.org/0000-0002-0307-3454>
- 주관심분야 : 포인트 클라우드, 디지털 홀로그래피, 2D/3D 영상 처리 및 압축



**박 병 서**

- 2019 2월 : 광운대학교 경영학과 졸업
- 2019 3월 ~ 현재 : 광운대학교 전자재료공학과(석박통합과정)
- ORCID : <https://orcid.org/0000-0003-3396-8264>
- 주관심분야 : 포인트 클라우드, 딥러닝, 2D/3D 영상 처리



**권 순 철**

- 2012년 08월: 광운대학교 정보디스플레이 공학박사
- 2012년 09월 - 2013년 02월: 광운대학교 초빙교수
- 2013년 - 현재 : 광운대학교 부교수
- ORCID: <https://orcid.org/0000-0001-6595-6415>
- 주관심분야: 3D그래픽스, 3D영상처리

---

저 자 소 개

---



김 동 욱

- 1983년 2월 : 한양대학교 전자공학과 졸업(공학사)
- 1985년 2월 : 한양대학교 공학석사
- 1991년 9월 : Georgia공과대학 전기공학과(공학박사)
- 1992년 3월 ~ 현재 : 광운대학교 전자재료공학과 정교수
- ORCID : <https://orcid.org/0000-0001-6106-9894>
- 주관심분야 : 3D 영상처리, 디지털 홀로그램, 디지털 VLSI Testability, VLSI CAD, DSP설계, Wireless Communication



서 영 호

- 1999년 2월 : 광운대학교 전자재료공학과 졸업(공학사)
- 2001년 2월 : 광운대학교 일반대학원 졸업(공학석사)
- 2004년 8월 : 광운대학교 일반대학원 졸업(공학박사)
- 2005년 9월 ~ 2008년 2월 : 한성대학교 조교수
- 2008년 3월 ~ 현재 : 광운대학교 전자재료공학과 교수
- ORCID : <https://orcid.org/0000-0003-1046-395X>
- 주관심분야 : 실감미디어, 2D/3D 영상 신호처리, 디지털 홀로그램