

일반논문 (Regular Paper)

방송공학회논문지 제25권 제5호, 2020년 9월 (JBE Vol. 25, No. 5, September 2020)

<https://doi.org/10.5909/JBE.2020.25.5.798>

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

Multi-view Stereo에서 Dense Point Cloud를 위한 Fusing 알고리즘

한현덕^{a)}, 한종기^{a)†}

Fusing Algorithm for Dense Point Cloud in Multi-view Stereo

Hyeon-Deok Han^{a)} and Jong-Ki Han^{a)†}

요약

디지털 카메라와 휴대폰 카메라의 발달로 인해 이미지를 기반으로 3차원 물체를 복원하는 기술이 크게 발전했다. 하지만 Structure-from-Motion(SfM)과 Multi-view Stereo(MVS)를 이용한 결과인 dense point cloud에는 여전히 듬성한 영역이 존재한다. 이는 깊이 정보를 추정하는데 있는 어려움과, 깊이 지도를 point cloud로 fusing할 때 이웃 영상과의 깊이 정보가 불일치할 경우 깊이 정보를 삭제하고 point를 생성하지 않았기 때문이다. 본 논문에선 평면을 모델링하여 삭제된 깊이 정보에 새로운 깊이 정보를 부여하고 point를 생성하여 기존 결과보다 dense한 point cloud를 생성하는 알고리즘을 제안한다. 실험 결과를 통해 제안하는 알고리즘이 효과적으로 기존의 방법보다 dense한 point cloud를 생성함을 확인할 수 있다.

Abstract

As technologies using digital camera have been developed, 3D images can be constructed from the pictures captured by using multiple cameras. The 3D image data is represented in a form of point cloud which consists of 3D coordinate of the data and the related attributes. Various techniques have been proposed to construct the point cloud data. Among them, Structure-from-Motion (SfM) and Multi-view Stereo (MVS) are examples of the image-based technologies in this field. Based on the conventional research, the point cloud data generated from SfM and MVS may be sparse because the depth information may be incorrect and some data have been removed. In this paper, we propose an efficient algorithm to enhance the point cloud so that the density of the generated point cloud increases. Simulation results show that the proposed algorithm outperforms the conventional algorithms objectively and subjectively.

Keyword : Point Cloud, Depth Information, Multi-view Stereo

a) 세종대학교 전자정보통신공학과(Sejong University, Dept. of Electrical Engineering)

† Corresponding Author : 한종기(Jong-Ki Han)

E-mail: hjk@sejong.edu

Tel: +82-2-3408-3739

ORCID: <https://orcid.org/0000-0002-5036-7199>

※ This work was partly supported by the National Research Foundation of Korea (NRF) under Grant NRF-2018R1A2A2A05023117 and partly by Institute for Information & communications Technology Promotion (IITP) under Grant 2017-0-00486 funded by the Korea government (MSIT).

· Manuscript received August 28, 2020 Revised September 17, 2020; Accepted September 17, 2020.

I. 서 론

디지털 카메라와 휴대폰 카메라의 발달로 인해 누구나 손쉽게 고화질의 영상을 얻을 수 있게 되어 고화질의 영상을 이용하여 3차원 물체를 복원하는 기술이 최근에 크게 발전했다. 3차원 물체를 복원하는 기술을 간단히 소개하면 다음과 같다. 우선 이미지들의 특징점을 [1][2]와 같은 방법을 이용하여 추출하고 특징점 매칭쌍을 구해 카메라 내부 파라미터와 외부 파라미터를 구한다. 이와 같은 과정을 거쳐 sparse point cloud를 생성할 수 있고 이를 Structure-from-Motion(SfM)^{[3][4]}이라고 한다.

Sparse point cloud를 dense하게 만들기 위해 [5][6][7]과 같은 Multi-view Stereo(MVS) 기술들이 존재한다. Sparse point cloud를 dense하게 만드는 것은 point cloud의 quality를 높일 뿐 아니라 [8][9]와 같은 mesh reconstruction 기술에서 약간의 노이즈가 포함되더라도 dense한 point cloud를 요구한다는 점에서 상당히 중요하다.

MVS에서 sparse point cloud를 dense하게 만드는 과정은 크게 두 개의 과정으로 볼 수 있다. 첫 번째 과정은 SfM에서 구한 카메라 내부 파라미터와 외부 파라미터를 이용해 각각의 영상마다 깊이 정보를 추정하여 깊이 지도를 생성한다. 두 번째 과정은 앞서 추정한 깊이 지도를 이용하여 각 영상의 픽셀마다 추정된 깊이 정보 값을 이웃 영상의 픽셀에서 추정된 깊이 정보 값과 비교한다. 이를 이용해 월드 좌표계에 알맞은 위치를 구하고 point를 생성하는 과정이 깊이 지도를 fusing하는 과정이다.

MVS를 이용하면 dense point cloud를 얻을 수 있지만 여전히 sparse한 부분이 존재할 수 있다. Dense point cloud에서 sparse한 부분이 존재하게 되는 이유는 영상에 특징이 없을 경우 깊이 정보를 추정할 때 정확한 깊이 정보를 구하기 힘들기 때문이다. 깊이 정보를 추정하는 단계에서 영상에 특징이 없는 부분은 이웃 영상과의 시차를 정확히 구하기가 힘들어 부정확한 깊이 정보를 추정하게 된다. 부정확한 깊이 정보는 깊이 지도를 fusing하는 과정에서 부정확한 깊이 정보를 갖는 해당 픽셀을 3차원 공간으로 projection한 후 이웃 이미지로 reprojection했을 때 만나는 픽셀의 깊이 정보와 불일치하기 때문에 깊이 정보가 삭제되고 point cloud로 point를 생성하지 못하게 된다. 즉, 깊이 지도를

fusing하는 단계에서 한 영상의 픽셀의 깊이 정보뿐 아니라 이와 상응하는 이웃 영상의 픽셀의 깊이 정보 역시 잘 구해지는 경우에만 point를 생성할 수 있고 반대의 경우 깊이 정보가 제거되고 point 역시 생성할 수 없어 sparse한 부분이 존재하게 된다.

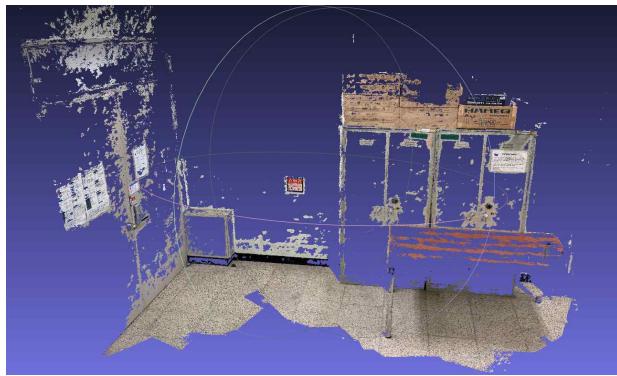
본 논문에선 생성된 point cloud를 이용하여 평면을 모델링하여 깊이 지도를 fusing하는 단계에서 현재 영상의 깊이 정보와 이웃 영상의 깊이 정보가 불일치하여 깊이 정보가 제거된 픽셀에 새로 올바른 깊이 정보를 구해 point를 생성하도록 하는 알고리즘을 제안한다. 본 논문의 구성은 다음과 같다. II장에서 기존의 방법들과 문제점을 소개하고 III장에서 제안하는 알고리즘을 설명한다. IV장에서 실험결과를 기존의 방법들과 비교하여 보이고 V장에서 향후 연구방향과 함께 결론짓는다.

II. 기존 알고리즘 및 문제점

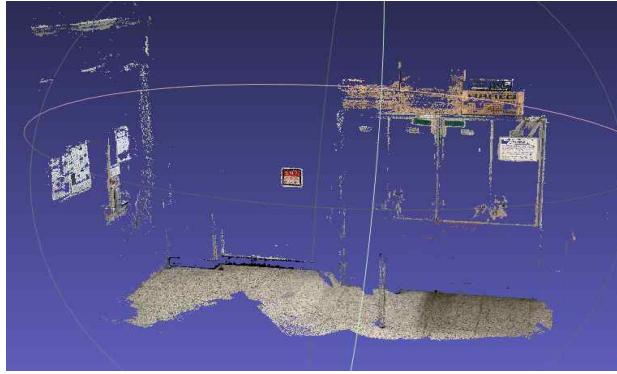
Point cloud의 완성도와 quality를 높이기 위한 기존의 연구는 크게 두 분류로 나눌 수 있다. 하나는 깊이 정보를 추정하는 방법^{[10][11]}에 대한 연구이고 또 다른 하나는 깊이 지도를 fusing하는 방법^{[6][7]}에 대한 연구다.

[10]은 촬영하는 물체의 표면이 카메라의 이미지 평면과 평행하지 않은 경우를 고려하여 깊이 정보를 구하는 방법을 제안했다. 즉, 생성하려는 point의 법선 벡터를 고려하여 깊이 정보를 추정하는 방법이다. [11]에서는 [12]에서 제안한 superpixel로 영상을 분할한 후 평면을 모델링하여 깊이 정보가 존재하지 않는 픽셀에 깊이 정보를 채워주는 방법을 제안했다.

[10][11]과 같이 깊이 정보를 잘 구하려는 노력에도 불구하고 깊이 지도를 fusing하는 과정에서 깊이 정보, 법선 벡터와 같은 정보들이 이웃 이미지의 깊이 정보, 법선 벡터와 일치하지 않을 경우 기존의 방법^{[6][7]}은 깊이 정보와 법선 벡터 정보를 제거하기 때문에 point를 생성하지 못해 그림 1과 같이 최종 point cloud에서 sparse한 영역이 발생하게 된다. [6]은 중복되는 point는 1개만 생성하고 occluded point를 판별하여 제거하는 방법을 제안했다. 그림 2에서 C_r 은 참조 이미지의 카메라를 나타내고 C_1, C_2, C_3, C_4 는 참



(a) Point cloud resulted from [6]



(b) Point cloud resulted from [7]

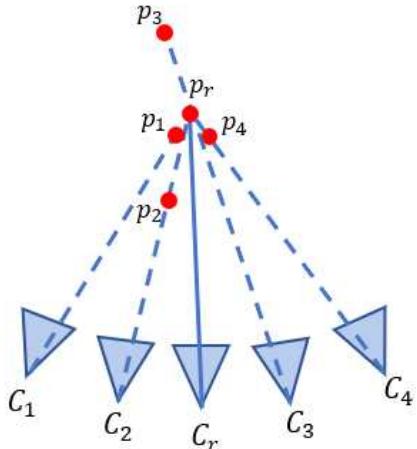
그림 1. [6][7]을 이용하여 생성한 point cloud

Fig. 1. Point cloud resulted from [6] and [7]

조 이미지의 이웃 이미지들의 카메라를 나타낸다. [6]은 참조 카메라의 깊이 지도에 존재하는 각각의 픽셀마다 해당 픽셀의 point p_r 과 이에 해당하는 이웃 카메라의 픽셀의 points p_1, p_2, p_3, p_4 의 위치를 고려하여 각 픽셀의 깊이 정보의 유효성을 판단한다. 그림 2에서 p_3 의 경우 참조 픽셀의 point p_r 보다 더 깊은 깊이 정보를 가지는데 [6]에서는 이를 p_3 은 p_r 에 의해 가려졌다고 판단하여 카메라 C_3 에서 p_3 에 해당하는 픽셀의 깊이 정보를 제거한다. 그림 2에서 p_1, p_4 의 경우 p_r 과의 거리가 임계값 τ 보다 작으므로 중복되는 point로 판단하여 카메라 C_1, C_4 에서 p_1, p_4 에 해당하는 픽셀의 깊이 정보를 제거한다. p_2 와 같이 위의 두 가지 경우에 모두 해당하지 않는 경우에는 깊이 정보를 제거하지 않고 넘어간다. [6]의 깊이 지도를 point cloud로 fusing하는 방법은 중복되는 point를 생성하지 않고 occluded

point를 고려하여 point cloud의 정확성을 높이지만 occlude 여부의 판단에 따라 제거된 깊이 정보에 별 다른 조치를 취하지 않는다. [7]은 point cloud의 quality를 높이기 위해 각 픽셀의 깊이 정보, 법선 벡터, 픽셀의 3차원 좌표를 이웃 이미지의 일치하는 픽셀과 비교하여 충분히 일치하지 않을 경우 해당 픽셀의 point를 만드는 것을 포기한다. [6]에서와 마찬가지로 [7]은 point를 만들어내지 못한 픽셀에 대해 아무런 조치를 취하지 않아 완성된 point cloud에 sparse한 영역이 존재한다.

본 논문에선 [6][7]과 같은 기존의 방법들이 깊이 지도를 fusing하여 point cloud를 만드는 단계에서 각각의 조건에 따라 point를 만드는 것을 포기하고 깊이 정보를 삭제한 픽셀들에게 올바른 깊이 정보를 부여하여 point로 생성될 수 있도록 하는 알고리즘을 제안한다.

그림 2. [6]에서 제안하는 fusing 알고리즘에서 참조 카메라 C_r , point p_r 과 이와 이웃하는 카메라 C_1, C_2, C_3, C_4 와 point p_1, p_2, p_3, p_4 Fig. 2. Reference camera C_r and p_r with its neighbor cameras C_1, C_2, C_3, C_4 and points p_1, p_2, p_3, p_4 in [6]

III. 제안하는 알고리즘

본 논문은 생성된 point cloud를 이용해 평면을 모델링하고 이를 기반으로 point를 생성하지 못하고 깊이 정보가 제거된 픽셀에 깊이 정보를 부여하여 point를 생성하도록 하는 알고리즘을 제안한다. 제안하는 알고리즘은 크게 두 단

계로 구성된다. 첫 번째로 각 이미지를 Canny edge detection [13]을 이용하여 영상의 edge를 추출하고, 이를 이용해 각 영상마다 평면을 모델링하는데 사용할 영역을 골라내는 단계다. 두 번째 단계에선 평면을 모델링할 영역에서 point가 존재하는 픽셀들을 이용하여 평면을 모델링하고 깊이 정보가 없는 pixel에 올바른 깊이 정보를 부여하여 point를 생성해 내는 단계로 구성된다.

1. 각 영상에서 평면 모델링에 사용할 영역 추출

Point cloud에서 평면을 모델링 할 때 사용할 point를 point cloud 상에서 고르는 것은 거의 불가능하고 가능하더라도 많은 계산을 요구한다. 따라서 본 논문에선 각 영상에서 평면을 모델링 할 때 사용할 영역을 추출하고 추출한 영역 중 깊이 정보, 범선 벡터와 같은 정보가 유효하여 point를 생성한 픽셀들을 이용하여 평면을 모델링한다.

영상을 분할하는 방법은 여러 방법이 존재하지만 본 논문에선 Canny edge detection을 사용한다. Edge detection을 사용하는 이유는 edge가 존재하는 영역에서 edge가 없는 영역보다 이웃 영상과의 시차를 구하기 쉽고 보다 정확한 깊이 정보가 구해져 point가 잘 생성되기 때문이다. 즉, edge를 따라서 영상을 분할한 후 edge로 둘러싸인 영역을 추출하면 추출된 영역의 가장자리의 픽셀들은 대부분 point가 존재하고 이를 이용해 평면을 모델링 한 후 추출한 영역에서 point가 없는 픽셀에 새로 올바른 깊이 정보를 부여할

수 있기 때문에 edge detection을 사용하여 영상을 분할한다. 영상에서 edge로 둘러싸인 영역을 추출하는 방법으로는 BFS(Breadth First Search) [14]를 사용했고 이 과정의 결과를 그림 3이 나타낸다. 그림 3의 (a)는 원본 영상이고 (b)는 원본 영상에서 Canny edge detection을 이용해 edge를 추출한 영상이다. (c)와 (d)는 BFS를 이용하여 (b)에서 edge로 둘러싸인 영역을 추출한 결과다. BFS는 맹목적 탐색방법의 하나로 보통 큐 혹은 스택이 사용되며 방법은 다음과 같다. 우선 시작 노드를 방문한 후 큐에 저장하고 해당 노드에 방문했다는 표시를 남긴다. 다음으로 큐에서 노드를 꺼낸 후 꺼낸 노드와 인접한 모든 노드를 방문하고 방문한 노드엔 모두 방문했다는 표시를 한 후 큐에 저장한다. 큐에 존재하는 노드 중 하나를 골라 꺼내고 이에 인접한 노드 중 방문하지 않은 노드를 방문하고 역시 방문한 노드엔 표시를 하며 위와 같은 과정을 큐에 남은 노드가 없을 때까지 반복한다. 본 논문에선 edge로 둘러싸인 영역을 추출하기 위해 ‘0’에 해당하는 픽셀을 방문해야 할 노드, ‘1’에 해당하는 노드를 방문한 노드로 설정했고 변경된 값은 다음 영역을 찾을 때도 유지하여 다음 영역을 추출할 때 포함되지 않도록 했다. 그림 3에서 (c)의 경우 평면을 모델링하기 위한 영역으로 바람직하지만 (d)의 경우 바람직하지 못한 영역이 추출됐음을 확인할 수 있다. (d)의 경우 Canny edge detection과 같은 edge detection 알고리즘이 영상을 분할하는 목적이 아니라 edge를 찾기 위한 목적을 지니는 알고리즘이므로 에지가 중간에 조금씩 끊어질 수 있



(a) Original image



(b) Edges resulted by Canny detector



(c) Example of plane region resulted from BFS



(d) Example of non-plane region resulted from BFS

그림 3. Canny edge detection [13]과 BFS [14]로 추출한 영역 분류의 예시
 Fig. 3. Examples of regions classified by Canny edge detection [13] and BFS [14]

기 때문에 발생하는 현상이다. 이를 해결하기 위해 [15]의 dilation을 사용한다. dilation은 edge를 두껍게 하여 살짝 끊어진 edge를 다시 이어지도록 해준다. 본 논문에선 dilation의 방법으로 3×3 window를 움직여가며 window안에 edge 즉 ‘1’이 존재할 경우 window의 중앙 픽셀의 값을 ‘1’로 설정하는 방법을 사용했다.

그림 4는 그림 3의 (d)에서 잘못 찾아진 영역이 dilation을 이용한 후에는 잘 찾아지는 것을 보여준다. 그림 4의 (a)는 그림 3의 (b)에 dilation을 적용한 후의 결과로 edge가 굽어진 것을 확인할 수 있다. 그림 4의 (b), (c), (d), (e), (f)는 그림 4의 (a)에서 BFS를 이용하여 edge로 둘러싸인 영역을 추출한 결과로 영역을 빨간색을 이용해 나타냈다. 추출된 영역은 모두 평면으로 평면을 모델링하기 적합한 영역을 잘 추출함을 확인할 수 있다.

2. 평면 모델링을 통한 깊이 정보 값 부여

앞서 그림 4의 (b), (c), (d)와 같이 영상의 edge로 둘러싸인 영역을 추출하고 나면 이 영역에 해당하는 픽셀들 중 일부는 point cloud에 point가 존재하고 나머지는 point가 존재하지 않는다. 우리의 목적은 이 point들을 이용해 평면을 모델링 한 후 point가 존재하지 않는 픽셀들에 올바른 깊이 정보를 부여하고 point로 생성되도록 하는 것이다.

기본적으로 평면을 모델링하기 위해선 3개의 point가 필요하다. 우리는 평면을 모델링할 때 고르는 3개의 point에 outlier가 포함되는 것을 방지하기 위해 RANSAC [16]을 사용했다. 월드 좌표계에 point cloud로 존재하는 하나의 point (x_0, y_0, z_0) 를 지나고 법선벡터가 $\vec{n} = (a, b, c)$ 인 평면의 방정식은 식 (1)과 같다.



(a) Edges after dilation



(b) First example of region resulted from BFS



(c) Second example of region resulted from BFS



(d) Third example of region resulted from BFS



(e) Fourth example of region resulted from BFS



(f) Fifth example of region resulted from BFS

그림 4. Dilation을 적용한 후 BFS [14]로 추출한 영역 예시

Fig. 4. Examples of regions by applying BFS [14] after dilation of edges

$$a(x - x_0) + b(y - y_0) + c(z - z_0) = 0 \quad (1)$$

RANSAC으로 랜덤으로 고른 3개의 point를 이용해 평면의 법선벡터 $\vec{n} = (a, b, c)$ 를 구할 수 있고 식 (1)의 평면의 방정식은 $d = -(ax_0 + by_0 + cz_0)$ 라고 하면 간단하게 식 (2)로 표현할 수 있다.

$$ax + by + cz + d = 0 \quad (2)$$

모델링한 평면의 유효성을 확인하기 위해 앞서 추출한 영역 내에 존재하는 픽셀 중 point가 존재하는 픽셀의 point와 모델링한 평면에 수직으로 내린 거리가 일정 임계값 τ 보다 작은 픽셀의 수를 구한다. 영역 내에 존재하는 픽셀 중 point가 존재하는 픽셀의 수와 앞서 구한 수의 비율을 구하여 이 비율이 일정한 값 γ 이상인 경우에 유효한 평면으로 판단했다. 모델링한 평면의 유효성을 판단하기 위해 사용되는 N 개의 point들 중 i 번 째 point (x_i, y_i, z_i) 에서 모델링한 평면에 수직으로 내린 거리 h_i 를 구하는 과정은 다음과 같다.

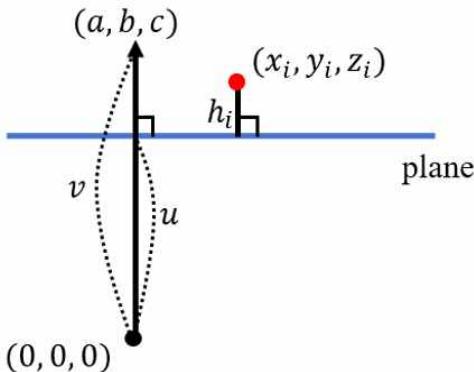


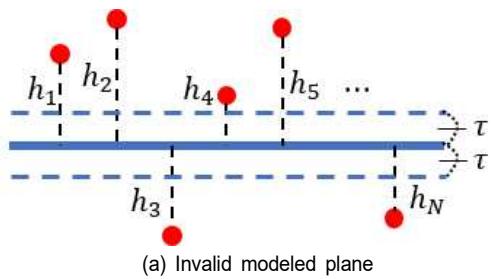
그림 5. 모델링한 평면과 평면의 유효성을 판단하기 위한 point와의 관계
 Fig. 5. The relationship between the modeled plane and a point to judge the plane's validity of the modeled plane

그림 5와 같이 평면의 법선벡터 $\vec{n} = (a, b, c)$ 의 크기가 v , 원점에서 평면까지의 거리가 u 일 때, 수식 (2)에서 d 의 값은 $-uv$ 다. 앞서 3개의 point를 사용해 평면을 모델링 했으므로 우리는 d 의 값을 알고 있고 v 도 법선벡터의 크기로 쉽게 구할 수 있으므로 u 역시 우리가 구할 수 있는

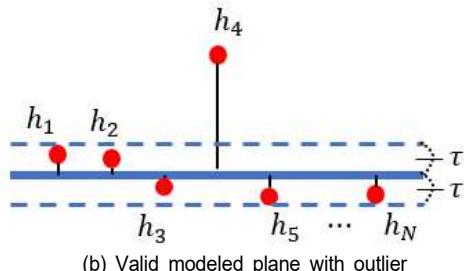
값이다. 평면의 법선벡터 (a, b, c) 와 (x_i, y_i, z_i) 와의 내적은 $v(h_i + u)$ 이므로 point와 모델링한 평면과의 거리 h_i 는 식 (3)과 같다.

$$h_i = \frac{ax_i + by_i + cz_i}{v} - u \quad (3)$$

그림 6은 모델링한 평면과 평면의 유효성을 평가하기 위한 point들을 나타낸다. 그림 6에서 (a)는 모델링한 평면과 모델링에 사용되지 않은 point들의 거리가 대부분 임계값 이상인 경우로 유효하지 못한 평면을 나타낸다. 이와 같은 경우는 추출한 영역에 하나의 평면만 존재하지 않고 여러 개의 평면 혹은 곡면이 존재할 경우 발생할 수 있다. 그림 6에서 (b)는 대부분의 point들과 평면의 거리가 임계값 이하인 경우로 유효한 평면을 나타낸다. point cloud에는 간혹 심하게 잘못 만들어진 h_4 와 같은 outlier가 존재한다. outlier가 존재하기 때문에 유효한 평면을 판별할 때 각 point들과 모델링된 평면 사이의 거리의 평균을 이용하지 않고 일정 임계값을 정한 후 이 임계값 안에 들어오는 point들과 전체 point 수의 비율을 이용하는 것이 유리하다.



(a) Invalid modeled plane



(b) Valid modeled plane with outlier

그림 6. 유효하지 못한 평면과 outlier가 포함된 유효한 평면
 Fig. 6. (a) is invalid modeled plane and (b) is valid modeled plane which contains outlier

모델링한 평면이 유효하다고 판단되면 앞서 추출한 영역 내에 존재하는 픽셀 중 point가 존재하지 않는 픽셀에 새로 깊이 정보값을 부여해야 한다. 이때 깊이 정보 d 는 해당 픽셀의 point가 모델링된 평면 위에 존재하도록 부여한다. 깊이 정보 d 를 구하기 위해 우선 회전행렬 R 과 이동벡터 T 를 먼저 정의할 필요가 있다. 회전행렬 R 과 이동벡터 T 가 각각 식 (4)와 식 (5)와 같을 때, 본 논문에선 이동벡터 T 가 월드 좌표계의 원점을 카메라 좌표계의 원점으로 이동시켜 주고 회전행렬 R 은 월드좌표계를 카메라 좌표계로 회전시켜주는 행렬로 정의했으며 이를 식 (6)에 나타냈다.

$$R = \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} \quad (4)$$

$$T = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \quad (5)$$

$$\begin{pmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{pmatrix} = \begin{pmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{pmatrix} \begin{pmatrix} x_{world} - t_1 \\ y_{world} - t_2 \\ z_{world} - t_3 \end{pmatrix} \quad (6)$$

위와 같이 회전행렬과 이동벡터를 정의했을 때, 영상에서 픽셀의 좌표가 (x_{img}, y_{img}) 이고 이에 해당하는 깊이 정보가 d 라면 이를 카메라 좌표계로 변환하는 수식은 식 (7)과 같고 카메라 좌표계의 $(x_{cam}, y_{cam}, z_{cam})$ 을 월드좌표계로 변환하는 수식은 식 (8)과 같다.

$$\begin{pmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{pmatrix} = \begin{pmatrix} (x_{img} - K(0, 2)) \times d / K(0, 0) \\ (x_{img} - K(1, 2)) \times d / K(1, 1) \\ d \end{pmatrix} \quad (7)$$

$$\begin{pmatrix} x_{world} \\ y_{world} \\ z_{world} \end{pmatrix} = \begin{pmatrix} r_1 & r_4 & r_7 \\ r_2 & r_5 & r_8 \\ r_3 & r_6 & r_9 \end{pmatrix} \begin{pmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \quad (8)$$

이때 식 (7)과 식 (8)의 $K(m, n)$ 은 [3][4]와 같은 SfM에서 구해진 카메라 내부 파라미터 행렬 K 의 m번째 행, n번째 열을 의미한다.

식 (7)과 식(8)을 이용해 새로 깊이 정보 d 를 부여하고 싶은 픽셀을 미지수 d 가 포함된 상태로 월드좌표계의 좌표

로 변환할 수 있으며 이 좌표는 모델링한 평면 위에 존재해야 하므로 모델링한 평면의 방정식에 대입하여 깊이 정보 d 를 구할 수 있다.

깊이 정보 d 를 구하고 나면 point를 생성하여 보다 dense한 point cloud를 생성할 수 있다. 이웃 이미지에서 중복되는 point 생성을 막기 위해 깊이 정보 d 를 구해준 픽셀과 일치하는 이웃 이미지의 픽셀의 깊이 정보를 앞에서 구해준 깊이 정보 d 와 상응하는 깊이 정보로 채워준다.

IV. 실험 결과

제안하는 알고리즘은 point cloud가 생성된 상태에서 추가적으로 point를 생성하기 위한 알고리즘으로 [6]의 방법에 추가적으로 제안하는 알고리즘을 적용하여 [6][7]을 이용한 실험 결과와 다양한 영상들을 이용해 비교했다. 그림 7은 휴대폰 카메라를 이용하여 촬영한 영상들을 이용하여 point cloud를 생성한 결과를 나타낸다. 그림 7의 (a)는 원본 영상을 나타내고 (b), (c)는 각각 [6], [7]을 이용한 결과를 나타낸다. (d)는 제안하는 알고리즘을 적용한 결과로 [6], [7]의 결과보다 dense한 point cloud를 생성했음을 확인할 수 있다.

그림 8 역시 그림 7과 마찬가지로 휴대폰 카메라를 이용하여 촬영한 영상들을 이용하여 point cloud를 생성한 결과를 나타낸다. 그림 8의 (a), (b), (c)는 각각 다른 시점의 원본 영상을 나타내고 (d), (e)는 [6], [7]을 이용한 결과를 나타낸다. (f)는 제안하는 알고리즘을 적용한 결과로 [6], [7]의 결과보다 dense한 point cloud를 생성했음을 확인할 수 있다. 그림 8에서 (a)의 원본 영상에 존재하는 가방의 경우 곡면을 가지고 있어 유효한 평면이 모델링될 수 없다. 평면 뿐 아니라 곡면도 모델링하여 둘 중 더 잘 어울리는 모델을 선택하여 point를 추가할 수 있다면 좋겠지만 곡면의 경우 모델링에 사용할 point들의 기하 정보가 대부분 부족하다. 그림 8에서 [6], [7]에서 생성된 결과를 보면 가방의 가장자리 부분은 영상에서 edge 근처에 해당하는 영역으로 point가 잘 생성된 반면 가방의 가운데 부분은 생성된 point가 없는 것을 확인할 수 있다. 이는 깊이 정보를 추정할 때 edge 부근에서 깊이가 잘 추정되고 point cloud 역시 edge에

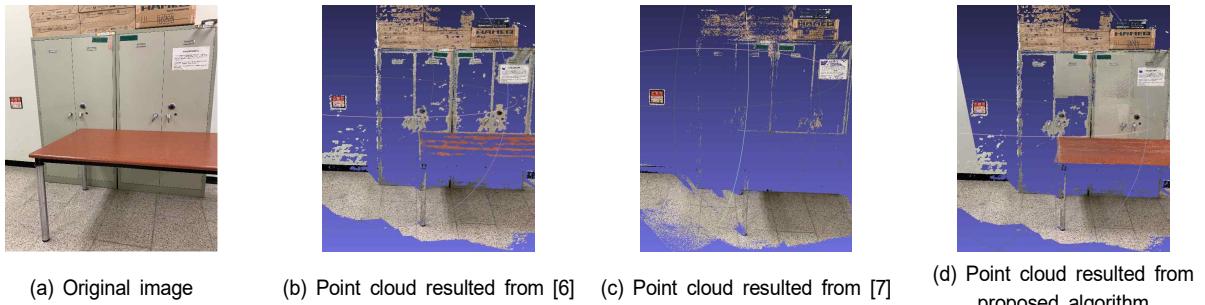


그림 7. 휴대폰 카메라를 이용해 얻은 영상들에 다양한 알고리즘을 적용한 결과 생성된 point cloud
 Fig. 7. Point clouds resulted from various algorithms with pictures taken by cell phone

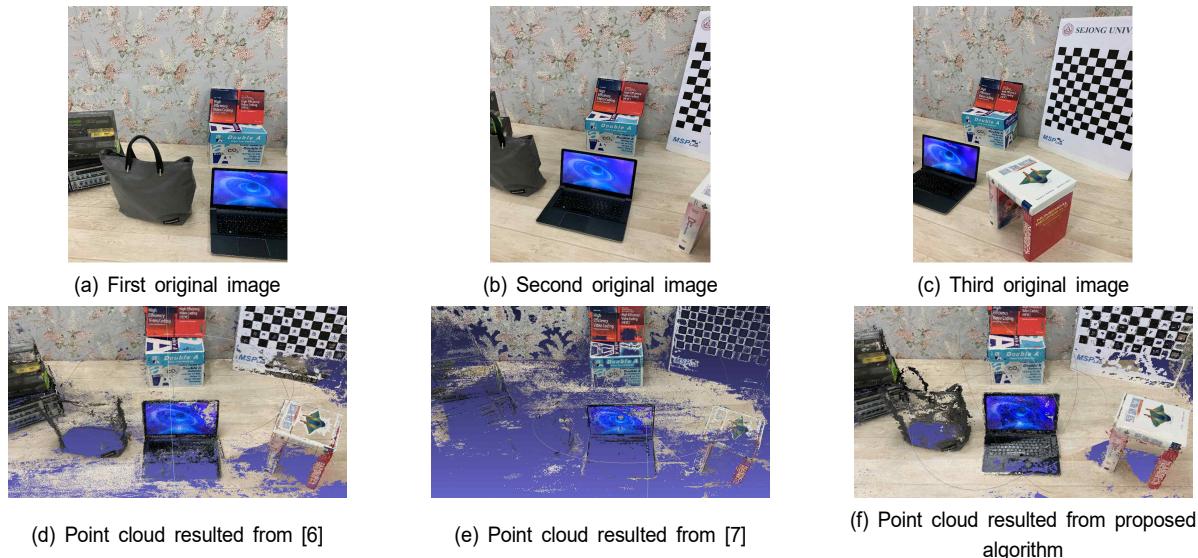


그림 8. 휴대폰 카메라를 이용해 얻은 영상들에 다양한 알고리즘을 적용한 결과 생성된 point cloud
 Fig. 8. Point clouds resulted from various algorithms with pictures taken by cell phone

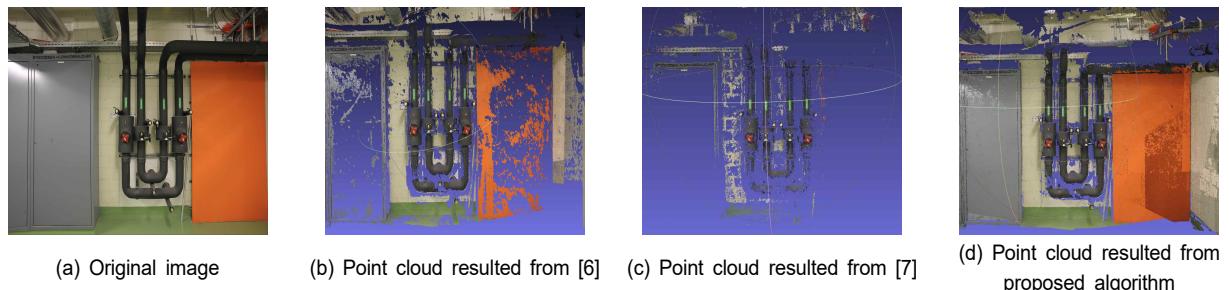


그림 9. [17]에서 제공한 영상들에 다양한 알고리즘을 적용한 결과 생성된 point cloud
 Fig. 9. Point clouds resulted from various algorithms with pictures given by [17]

서 dense하게 생성되기 때문이다. 평면의 경우 edge 부근의 point만을 이용하더라도 유효한 평면을 모델링할 수 있지

만 곡면의 경우 올바른 모델링을 하기 위해선 가운데 영역의 point의 기하 정보가 꼭 필요하다. 본 논문에선 정확한

point만을 추가하기 위해 기하 정보가 부족하여 정확한 모델링이 불가능한 곡면에 대해선 point를 추가하지 않고 edge 부근의 point만으로도 정확한 모델링이 가능한 평면에 대해서만 point를 추가하는 방법을 선택했다.

그림 9는 [17]에서 MVS를 평가하기 위해 제공하는 데이터 중 pipe를 이용한 결과를 나타낸다. 그림 9에서 (a)는 원본 영상을, (b)와 (c)는 각각 [6]과 [7]을 이용한 결과를 나타내고 (d)는 제안하는 알고리즘을 적용한 결과를 나타낸다. 제안하는 알고리즘을 적용한 결과는 기존의 방법을 이용한 결과보다 dense한 point cloud를 생성함을 확인할 수 있다.

표 1은 제안하는 알고리즘을 이용한 point cloud가 제안하는 알고리즘을 적용하지 않은 point cloud보다 좋은 성능을 보인다는 것을 객관적으로 보여준다. 표 1의 F1 score는 완성도와 정확성을 고려하여 더 좋은 결과일수록 더 높은 F1 score를 부여하는 [17]에서 제안한 방법이다.

표 1. F1 점수

Table 1. F1 Scores

Method	ETH3D data sets		
	pipe	terrains	courtyard
[6]	0.395	0.508	0.039
[6]+proposed	0.399	0.510	0.040

표 2는 제안하는 알고리즘을 [6]에서 제안한 방법에 추가로 적용했을 때 추가적으로 걸리는 시간을 이미지의 수와 생성된 point의 수에 따라 나타낸다. 표 2를 보면 제안하는 알고리즘을 추가로 적용했을 때 걸리는 시간이 이미지의 수와 생성되는 point의 수가 많더라도 3분 이내임을 확인할 수 있다. 이는 앞서 깊이 지도를 구하는 과정에서 수십 분이 소요된다는 점을 생각했을 때 상당히 적은 시간으로 볼 수 있다.

표 2. 이미지 수와 새로 추가된 point의 수에 따른 계산 시간
Table 2. Calculation times at various number of images

The number of images	23	44	76
The number of new added points	217,071	3,335,570	297,100
[sec]	61	173	47

V. 결 론

본 논문은 영상들로부터 point cloud를 생성하는 기존의 MVS 방법들에 존재하는 sparse한 영역에 point를 추가하여 dense하게 만드는 알고리즘을 제안한다. 제안하는 방법은 생성된 point cloud를 이용하여 평면을 모델링하여 point가 존재하지 않는 픽셀에 새로 깊이 정보를 부여해 point를 생성할 수 있도록 하는 알고리즘이다. IV장에서 실험을 통해 제안하는 알고리즘이 효과적으로 sparse한 영역에 point를 추가하여 dense하게 바꿀 수 있음을 증명했지만 여전히 연구가 필요한 부분이 존재한다.

우선 평면을 모델링할 때, point가 충분하지 않은 경우 평면을 모델링할 수 없어 그림 7과 같이 여전히 sparse한 영역이 남을 수 있다. 또한 모델링한 평면의 유효성을 판단할 때 사용한 임계값 τ 와 γ 에 대한 고찰도 필요하다. 본 논문에선 여러 실험에 근거하여 각각 0.004와 0.99를 임계값으로 사용했지만 실험 결과에 노이즈가 많이 포함된 경우 보다 최적화된 임계값을 적응적으로 결정하여 사용한다면 더 좋은 결과를 얻을 수 있을 것이다.

추가적으로 연구가 필요한 부분도 존재하지만 실험결과에서 보였듯이 제안하는 알고리즘은 평면 영역에서 효과적으로 point cloud의 sparse한 영역을 개선시킬 수 있다. 기존의 알고리즘으로 생성된 point가 그림 9처럼 충분하다면 제안하는 알고리즘을 적용했을 때 더욱 큰 효과를 볼 수 있고 모델링된 평면의 유효성을 판단할 때 고정된 임계값을 사용하더라도 기존의 방법을 능가하는 성능을 보임을 확인했다.

참 고 문 헌 (References)

- [1] D. Lowe, "Distinctive image features from scale-invariant keypoints," IJCV, 2004
- [2] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE features," In Eur. Conf. on Computer Vision (ECCV), pages 214 - 227, 2012.
- [3] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," Int. J. Comput. Vis., vol. 80, no. 2, pp. 189 - 210, 2008.
- [4] Schönberger JL, and Frahm JM: Structure-from-motion revisited. In: Agapito L, Berg T, Kosecka J, et al. (eds) Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las

Vegas, NV, June, 2016

- [5] M. Goesele, B. Curless, and S. M. Seitz, "Multi-View Stereo Revisited," Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, Washington, DC, USA, pp. 2402-2409, 2006.
- [6] 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.
- [7] Johannes L. Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys, "Pixelwise View Selection for Unstructured Multi-View Stereo," In European Conference on Computer Vision, pages 501 - 518. Springer, 2016.
- [8] M. Jancosek and T. Pajdla. "Multi-view reconstruction preserving weakly-supported surfaces," In Proc. CVPR, 2011.
- [9] Zhou, Y.; Shen, S.; Hu, Z. "Detail Preserved Surface Reconstruction from Point Cloud," Sensors, 2019, 19(6): 1278, doi:10.3390/s19061278.
- [10] Michael Bleyer, Christoph Rhemann, and Carsten Rother, "Patchmatch Stereo-Stereo Matching with Slanted Support Windows," In BMVC, volume 11, pages 1 - 11, 2011.
- [11] A. Romanoni and M. Matteucci. "TAPA-MVS: Texturelessaware PatchMatch multi-view stereo," In IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 2019.
- [12] Michael Van den Bergh, Xavier Boix, Gemma Roig, and Luc Van Gool, "SEEDS: Superpixels Extracted via Energy-Driven Sampling," International Journal of Computer Vision, 111(3):298 - 314, 2015.
- [13] B. Green, "Canny Edge Detection Tutorial", http://www.pages.drexel.edu/~weg22/can_tut.html, 2002.
- [14] E. F. Moore, "The shortest path through a maze," In Int. Symp. on Th. of Switching, pp. 285 - 292, 1959.
- [15] Pierre Soille, Morphological image analysis: Principles and applications. 2nd edition. Berlin, Germany: Springer-Verlag, 2003.
- [16] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography", Communications of the ACM, vol. 24, no. 6, pp. 381-395, Jun. 1981.
- [17] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, A. Geiger, "A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos," Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

저자 소개



한 현 덕

- 2020년 9월 현재 : 세종대학교 전자정보통신공학과 학사과정
- ORCID : <https://orcid.org/0000-0001-6323-593X>
- 주관심분야 : VR, 영상 신호처리



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 : 비디오 코덱, 영상 신호처리, 정보 압축, 방송 시스템