

OpenVSLAM에 기반한 구면 카메라 스트림에서의 실내 공간 인식

홍철기, 박종승
인천대학교 컴퓨터공학과
mico9901@inu.ac.kr, jong@inu.ac.kr

Indoor Space Recognition from Spherical Camera Stream based on OpenVSLAM

Cheol-gi Hong, Jong-Seung Park
Dept. of Computer Science and Engineering,
Incheon National University

요 약

본 논문에서는 구면 영상을 사용한 vSLAM에 의해 생성된 환경 지도에서 실내 공간을 인식하는 방법을 제안한다. 환경 지도는 오픈 소스 라이브러리 OpenVSLAM을 사용하여 생성했다. 카메라 방향과 위치를 기준으로 랜드 마크를 분류하고 허프 변환을 사용해서 실내 공간의 각 벽의 위치를 찾아냈다. 실험 결과 추정된 평면들이 실제 벽면과 유사한 위치에 나타남을 알 수 있었다. 제시하는 알고리즘은 현재의 AR 콘텐츠보다 진보된 AR 콘텐츠를 제작하는 데 사용할 수 있다.

장에서 결론을 맺는다.

1. 서론

vSLAM에서는 영상센서를 이용해서 주변의 환경 맵을 생성하고 카메라의 위치를 추정한다. 평면 동영상을 사용하는 경우 한 장의 영상에서 전체 공간 중에서 일부분만을 얻을 수 있으므로 다 방향에 대한 정보를 얻기 위해서는 카메라의 많은 회전량이 요구된다. 하지만 구면 동영상을 사용하게 되면 카메라를 회전시키지 않아도 전 방향에 대한 정보를 모두 얻을 수 있으므로 적은 데이터 량과 최소한의 카메라 움직임으로 공간 형태를 파악할 수 있다.

최근에 등장한 오픈소스 라이브러리 OpenVSLAM[1]에서는 평면 카메라 모델과 구면 카메라 모델, 단안 렌즈 카메라와 쌍안 렌즈 카메라에 대한 SLAM을 모두 제공한다. 이 라이브러리는 ORB SLAM[2][3], ProSLAM[4], UcoSLAM[5]을 기반으로 SLAM을 수행한다.

본 논문에서는 오픈 소스 라이브러리를 사용해 찾은 환경 지도로부터 실내 공간을 인식하는 방법을 제안한다. 2장에서 환경 지도의 랜드 마크 집합으로부터 실내 공간의 각 벽을 찾는 방법을 설명하고 3장에서 실험을 통해 제시한 방법으로 찾은 벽면이 실제 환경과 얼마나 유사한지 보인다. 마지막으로 4

2. 실내 공간 인식 방법

본 논문에서 제안하는 방법은 SLAM으로 찾은 특정 키 프레임까지의 랜드 마크를 분류하고 허프 변환[6]을 사용해 평면을 찾은 후, 각 평면들을 기반으로 사각기둥 형태의 공간을 추정하는 단계로 구성된다. 첫 번째 단계에서는 0번부터 n 번까지의 프레임에서 나타난 랜드 마크의 집합 F 와 월드 좌표를 카메라의 방향 C^R 와 위치 C^L 와 비교해서 방향을 분류한다. 두 번째 단계에서 각 방향에 대한 랜드 마크 집합에 대해 카메라의 정면 벡터 \mathbf{d} , 법선 벡터 $\mathbf{n}_F, \mathbf{n}_C, \mathbf{n}_L, \mathbf{n}_R$, 주어진 해당 평면의 조건을 기준으로 허프 변환으로 평면 검출을 시도한다. 마지막 단계에서는 무한 평면의 식으로 나타나는 평면들을 실제 공간과 유사한 형태를 갖도록 배치한다. 이에 대한 의사 코드를 (그림 1)에 나타내었다. 본 논문에서는 정면 방향의 좌표축을 x , 상하 방향의 좌표축을 z 로 한다.

2.1. 랜드 마크 집합 분류

구면 동영상을 카메라의 기울어진 정도가 미세하도록 촬영했다고 가정하면 각 평면은 xy 평면에 대

해 수직하거나 평행하다고 추정할 수 있다. 동시에 상하 방향의 평면과 좌우 방향의 평면은 서로 수직 관계에 있으므로 각 평면은 월드의 x 축에 평행하다. 두 조건에 의해 F 를 다음과 같이 분류할 수 있다.

$$\begin{aligned} F_F &= \{f | f.z < C^L.z\} \\ F_C &= \{f | f.z > C^L.z\} \\ F_L &= \{f | f.y < C^L.y\} \\ F_R &= \{f | f.y > C^L.y\} \end{aligned} \quad (1)$$

```

Input:  $F, C^R, C^L$ 
Output:  $S$ 
Begin
   $d \leftarrow C^R.x$ 
   $F_F, F_C, F_L, F_R \leftarrow sectionFeaturePnts(F, C^L)$ 
   $n_F, n_C, n_L, n_R \leftarrow$  normal vectors of planes
   $P_F \leftarrow findPlane(F_F, d \perp n_F)$ 
   $P_C \leftarrow findPlane(F_C, n_F \parallel n_C)$ 
   $P_L \leftarrow findPlane(F_L, d \perp n_L \& n_F \perp n_L)$ 
   $P_R \leftarrow findPlane(F_R, n_L \parallel n_R \& n_F \perp n_R)$ 
   $S \leftarrow createSpace(P_F, P_C, P_L, P_R)$ 
End
    
```

(그림 1) 제안 방법의 개략적인 의사 코드

```

Input:  $F', Cond$ 
Output:  $P'$ 
Begin
  while  $P'$  is not defined
     $P'' \leftarrow HoughTransform(F')$ 
     $n \leftarrow$  normal vector of  $P''$ 
     $p \leftarrow F'_0.y$  or  $F'_0.z$ 
    if  $Cond$ 
       $P' \leftarrow P''$ 
    else
      exclude  $P''$ 
    end
  end
End
    
```

(그림 2) $findPlane$ 함수의 의사 코드

2.2. 허프 변환으로 평면 검출

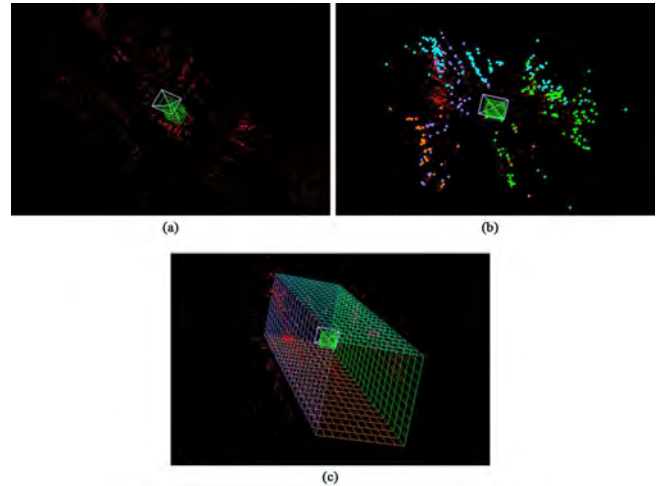
직교 좌표계에서 찾고자 하는 패턴이 있을 때 기존 좌표계를 허프 공간으로 치환함으로써 해당 패턴을 찾을 수 있으며[8], 이는 3차원 직교좌표계에서 평면을 찾고자 할 때도 적용할 수 있다[9]. 분류된 F 의 각 부분집합에 대해 평면 검출을 시도했을 때

주어진 조건에 맞는 평면이 검출될 때까지 새로운 평면 검출을 시도한다. 이는 $findPlane$ 함수에 해당되며 (그림 2)에 의사코드로 나타냈다.

검출된 평면들은 중심점으로부터 떨어진 거리 값과 법선 벡터만을 가지는 무한 평면이다. $createSpace$ 함수에서는 각 평면의 교선과 임의로 제한된 x 축의 값을 사용해 실내 복도의 모습을 완성한다.

3. 실험

본 논문의 실험은 Intel Core i7 CPU, NVIDIA Geforce GTX 970, 16GB RAM, Windows 7 64bit OS의 시스템에서 수행되었고, C++언어와 OpenCV, OpenVSLAM 라이브러리를 사용한 코드로 구현되었다. 실험에 사용된 영상은 OpenVSLAM에서 제공하는 데이터 소스를 사용했다. n 번째 키 프레임까지의 랜드마크 집합은 (그림 3)의 (a)에서 붉은 점으로 나타난다. 각 방향에 대해 분류된 랜드마크에서 허프 변환으로 찾은 평면에 속하는 점들은 4가지 색으로 분류되어 (그림 3)의 (b)와 같이 나타난다. 무한 평면으로 표현되는 각 평면들로부터 실내 환경을 재현한 모습은 (그림 3)의 (c)와 같다. 검출된 평면을 구면 동영상 데이터에 투영한 결과는 (그림 4)에 나타난다.

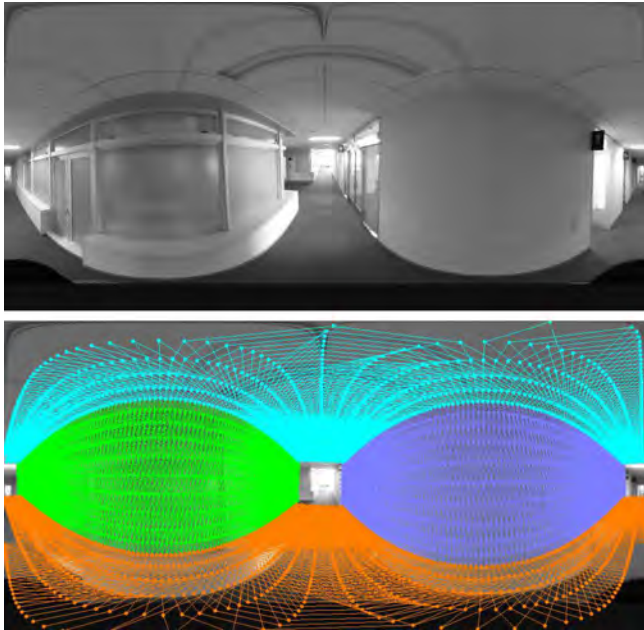


(그림 3) n 번째 키 프레임까지의 랜드마크들(a), 허프 트랜스폼으로 결정된 평면에 속하는 점들(b), 유한하게 정리된 평면들(c)

4. 결론

본 논문에서는 구면 동영상을 통해 실내 공간의 형태를 추정하는 방법을 제안했다. 제안하는 방법은 특징점이 충분히 검출되는 영역에서는 실제 벽의 위

치를 정확하게 추정할 수 있지만 그렇지 않은 영역에서는 정확도가 떨어지는 모습을 보였다. 추후 연구는 특징점 이외의 요소 또한 사용하는 방식으로 진행될 예정이다. 본 연구를 통해 하나의 평면에 국한되었던 기존의 AR 콘텐츠를 더 입체적으로 제작할 수 있을 것으로 기대한다.



(그림 4) 원본 구면 동영상(위), 구면 동영상에 투영된 평면(아래)

Acknowledgments

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2019R1F1A1060828).

참고문헌

[1] Shinya Sumikura, Mikiya Shibuya and Ken Sakurada, "OpenVSLAM: A Versatile Visual SLAM Framework", ACM Multimedia 2019 Open Source Software Competition, 2019.
 [2] Raul Mur-Artal, J. M. M. Montiel and Juan D. Tardós, "ORB - SLAM: a Versatile and Accurate Monocular SLAM System", IEEE Transactions on Robotics Vol. 31, No. 5, pp. 1147 - 1163, 2015.
 [3] Raúl Mur-Artal and Juan D. Tardós., "ORB - SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras", Transactions on Robotics Vol. 33, No. 5, pp. 1255

- 1262, 2017.

[4] Dominik Schlegel, Mirco Colosi and Giorgio Grisettik, "ProSLAM: Graph SLAM from a Programmer's Perspective" IEEE International Conference on Robotics and Automation. pp. 3833-3840, 2018.
 [5] Rafael Muñoz-Salinas and R. Medina-Carnicer, "UcoSLAM: Simultaneous localization and mapping by fusion of keypoints and squared planar markers", Pattern Recognition, Vol 101, 2019.
 [6] P. V. C. Hough, "Method and Means for Recognizing Complex Patterns", US Patent 3069654, 1962.
 [7] Dorit Borrmann, Jan Elseberg, Kai Lingemann and Andreas Nüchter, "The 3D Hough Transform for Plane Detection in Point Clouds: A Review and a new Accumulator Design", 3D Research, 2011.
 [8] J. Illingworth and J. Kittler, "The Adaptive Hough Transform", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 9, No. 5, 1987.
 [9] Dorit Borrmann, Jan Elseberg, Kai Lingemann and Andreas Nüchter, "The 3D Hough Transform for Plane Detection in Point Clouds: A Review and a new Accumulator Design", 3D Research, 2011.