

## 광각 영상에서 하나의 선분을 이용한 직육면체 레이블링 도구 개발

강준모<sup>\*1)</sup>, 조상묵<sup>\*1)</sup>, 최경택<sup>\*2)</sup>, 김도윤<sup>\*1)</sup>  
 WithRobot, Inc.<sup>\*1)</sup>, 대구가톨릭대학교 전자전기공학부<sup>\*2)</sup>

### Development of a Cuboid Labeling Tool using a Single Line segment in a wide-angle Image

Jun-Mo Kang<sup>\*1)</sup>, SangMuk Cho<sup>\*1)</sup>, Kyoungtaek Choi<sup>\*2)</sup>, Doyoon Kim<sup>\*1)</sup>  
 WithRobot, Inc.<sup>\*1)</sup>, Daegu Catholic University<sup>\*2)</sup>

**Abstract** - 본 논문에서는 인공지능 모델이 이동체의 중심점을 추정하기 위해서 더 정밀한 cuboid 형태의 레이블링 방식을 제안한다. 레이블링에 사용한 영상은 왜곡이 존재하여 보정하고, 보정된 영상에서 bounding box 형태와 cuboid 형태로 레이블링 하였다. 그리고 레이블링의 결과와 이동체의 중심점인 gt(ground truth)를 픽셀 거리로 계산하여 비교하고, 더 정밀한 cuboid 형태의 레이블링 방식의 결과 얻었다. 또한, cuboid 형태 레이블링 방식은 하나의 선분만을 이용하여 그리는 방식을 제시한다.

같이 왜곡이 보정된 영상을 얻을 수 있었다.



〈그림 1〉 왜곡이 존재하는 원본 영상



〈그림 2〉 왜곡이 보정된 영상

#### 1. 서 론

오늘날 인공지능 기술이 빠르게 발전하면서 레이블링에 대한 중요성이 증가하고 있다. 레이블링이란 인공지능 학습 데이터를 만들기 위해 원천 데이터에 값을 붙여 정답지를 만드는 작업이다.

인공지능 모델이 학습하는데 레이블링은 필수적으로 해야 하는 작업이며, 학습은 레이블링 된 데이터의 양이 많고, 정확할수록 성능이 향상된다. 인공지능 모델의 사용 목적에 따라 학습에 사용되는 레이블링 형태는 bounding box, polygon, polyline, point, cuboid 등 여러 가지가 있다. 본 논문에서는 인공지능 모델이 이동체의 중심점을 검출하기 위해 더 정밀한 바닥 중심점 방식[1]을 참고하여 cuboid 형태의 레이블링 방식에 초점을 두고 있다.

본 논문에서는 가능한 넓은 영역을 영상에 담기 위해 광각 렌즈로 촬영된 데이터를 사용하였다. 광각 렌즈는 초점 거리가 짧고 화각이 넓은 특성 때문에 자동차 후방카메라, 교통 감시용 CCTV에 많이 사용되고 있다. 광각 렌즈를 사용한 영상은 넓은 범위를 볼 수 있지만, 렌즈의 특성상 상대적으로 영상 왜곡이 심해지는 문제가 있다. 이러한 문제 때문에 calibration을 진행하여 얻은 파라미터를 사용하여 왜곡 보정을 하였다.

본 논문에서는 왜곡 영상을 보정하고, calibration을 진행한 영상에서 하나의 선분을 이용하여 cuboid 형태로 레이블링 되는 방식을 제시한다.

#### 2. 영상 왜곡 보정 및 calibration

광각 렌즈로 얻은 영상에서는 실제 3차원 공간상의 직선이 렌즈 왜곡 때문에 곡선으로 투영되기 때문에, 원본 영상에서 레이블링하면 정확한 위치를 추정하는데 오차가 발생한다. 그림 1과 같이 왜곡이 존재하는 영상에서 렌즈 왜곡 파라미터를 찾아내기 위해 직선상에 존재하는 점을 찍고, 이 점들이 최대한 직선으로 투영되도록 한 후 식(1), (2)와 같은 equidistant 모델[2]을 사용하여 함수  $L(r_d)$  계수를 추정하였다. 식(1)에서  $(x_d, y_d)$ 는 왜곡된 영상의 좌표이고  $(x_u, y_u)$ 는 보정된 영상의 좌표이며, 식(2)에서는  $r_d$ 는 영상 중심  $(x_c, y_c)$ 에서 왜곡된 영상의 좌표까지 거리를 의미한다.

식(2)에서 4차 항까지 사용하여 곡선이 직선이 되도록 함수의 계수를 추정하였다. 추정된 함수의 계수를 이용하여 그림 2와

$$\begin{bmatrix} x_d - x_c \\ y_d - y_c \end{bmatrix} = L(r_d) \begin{bmatrix} x_u - x_c \\ y_u - y_c \end{bmatrix} \quad \text{식 (1)}$$

$$L(r_d) = 1 + k_1 r_d^2 + k_2 r_d^4 + \dots + k_n r_d^{2n} \quad \text{식 (2)}$$

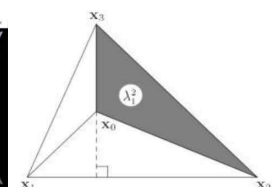
카메라 calibration은 소실점과 미리 알고 있는 물체의 높이 정보를 이용하여 카메라 내부 파라미터와 외부 파라미터의 값을 계산하여 진행했다.

일반적으로 소실점을 파악하기 위해서 calibration 차트를 이용한다. 하지만, 그림 3과 같이 환경 여건 때문에 일반적인 방법을 사용하지 못해서 왜곡이 보정된 영상에서 도로의 평행한 선분과 횡단 보도의 평행한 선분처럼 영상에 존재하는 서로 평행한 직선을 이용하여 소실점을 파악했다.

카메라의 광축이 영상 평면과 만나는 점을 주점이라 하며, 이를 편의상 영상 중심점으로 정하고, 중심점은 그림 4와 같이 세 축의 소실점들  $(X_1, X_2, X_3)$ 을 꼭짓점으로 하는 삼각형의 orthocenter  $(X_0)$ 가 된다. 두 개의 소실점과 중심점을 알고 있으면, 소실점과 중심점과의 관계를 활용하여 남은 한 개의 소실점을 계산할 수 있다[3].



〈그림 3〉 카메라 calibration을 위해 두 개의 축에 대한 평행선을 입력한 경우



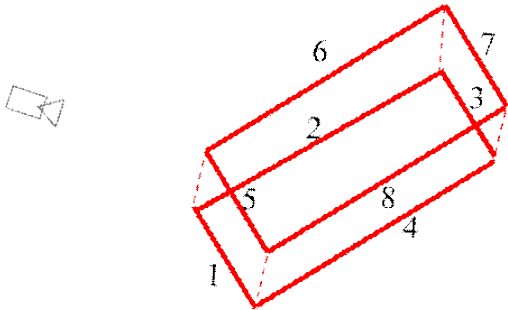
〈그림 4〉 소실점과 중심점의 관계

### 3. 레이블링

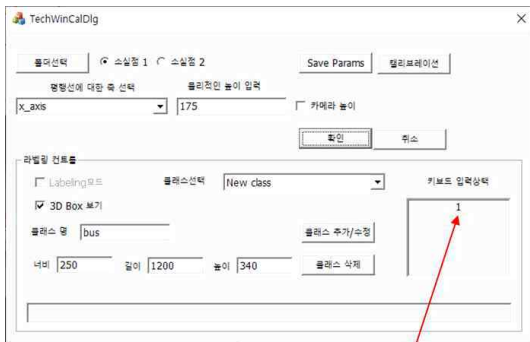
레이블링은 그림 5를 참고해서 그림 6처럼 사용자가 기준선의 번호를 선택하고 2차원 좌표인 하나의 선분을 이용해서 사용자가 설정한 템플릿(길이, 너비, 높이)에 의해 cuboid를 생성한다. 그리고 템플릿의 설정은 사용자가 초기에 설정할 수 있다.

사용자가 그린 기준선과 기준선의 다음 선분이 같은 면에 있다면 기준선의 3차원 좌표값에서 시작점의 좌표와 종점의 좌표에서 높이에 해당하는 좌표축을 제거하고, 시작점에서 종점으로 향하는 2차원 평면상의 벡터를 계산한다. 2차원 평면상의 벡터를 이용하여 단위 벡터를 얻어내고, 이를 90도 회전하여 다음 선분에 해당하는 단위 벡터를 생성한다. 단위 벡터에 초기에 설정한 템플릿을 곱하여 기준선에서 90도 회전한 다음 선분의 벡터를 생성한다. 이때 생성된 벡터는 원점 이동을 통해 계산한 선분의 종점으로 이동시킨다. 이런 방식으로 좌표를 하나씩 찾아내며, 다른 면에 대한 연산 시에는 설정한 템플릿에서 높이 값만 추가하여 계산한다.

모든 연산이 끝나면 3차원 좌표계에서 표현된 cuboid 형태의 8개의 점이 생성되고, 이를 앞서 만들어낸 projection matrix를 이용하여 다시 2차원 영상 좌표계로 변환하여 영상에 그림 7처럼 cuboid 형태로 그린다. 그리고 사용자가 cuboid의 위치를 이동하면 cuboid 그리는 방식과 유사하게 projection matrix를 이용해서 이동한 해당 좌표에 맞게 cuboid 크기가 변경된다. 이러면 사용자가 cuboid 크기를 다시 조정할 필요가 없이 작업시간을 줄일 수 있다.

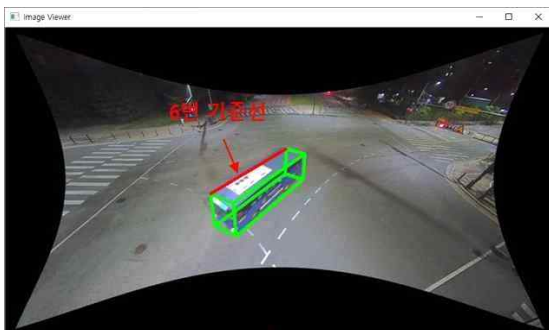


〈그림 5〉 하나의 선분인 기준선을 번호로 나타내는 예시



선택된 기준선번호를 표시함

〈그림 6〉 Cuboid 형태의 레이블링 틀 설정 창

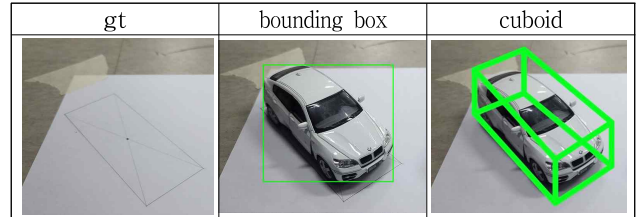


〈그림 7〉 Cuboid 형태의 레이블링 틀 화면

### 4. 레이블링 데이터 비교 결과

레이블링 데이터를 gt와 비교하려 했지만, 환경 여건 때문에 gt를 얻는 것이 어려워서 미니어처 차량으로 진행했다. 미니어처 차량 2대와 핸드폰 카메라를 이용하여 10장의 데이터를 얻었고, 영상의 크기는 4000x3000을 사용하여 실제 중심점 좌표인 gt는 opencv를 활용하여 픽셀값을 찾았다. 그리고 bounding box와 cuboid 두 개의 형태로 레이블링하여 그림[8]처럼 결과를 얻었다. 이 두 개의 형태로 얻은 중심점의 픽셀 좌표와 gt의 픽셀 좌표 간에 거리를 계산하여 비교한 결과는 표 1과 같다.

수행 결과 cuboid가 bounding box보다 평균 픽셀 오차가 적게 나온 것을 확인하였다. 이 결과를 토대로 cuboid 형태가 bounding box보다 정밀하다.



〈그림 8〉 gt와 레이블링한 bounding box, cuboid 결과

〈표 1〉 10장의 레이블링 데이터를 비교한 결과

오차	평균 픽셀 오차 (gt와 거리비교, 단위 : pixel)	
	소형 차량	대형 차량
labeling 형태		
bounding box	101.30	246.64
cuboid	18.00	19.05

### 5. 결론 및 추후 과제

본 논문에서는 bounding box의 중심점과 cuboid의 중심점을 gt와 비교하였다. 두 개의 방식을 gt와 비교한 오차는 표 1과 같다. cuboid 형태가 bounding box 형태보다 평균 픽셀 오차가 소형 차량은 약 83픽셀, 대형 차량은 약 227픽셀만큼 오차가 적게 나온 것을 확인했다. 이러한 결과를 토대로 cuboid 형태로 레이블링을 진행하면 이동체의 중심 좌표를 얻은 결과가 더 정밀했다. 그래서 하나의 선분을 이용한 cuboid 형태의 레이블링 방식을 제안하였다.

이동체의 방향을 표시해주는 기능과 사용자의 편의성을 위한 새로운 기능을 추가하고 보완할 예정이다.

#### 감사의 글

본 논문은 국토교통부 국토교통기술사업화지원사업의 연구개발지원([과제명: 자율협력주행용 LDM layer4 정보 생성을 위한 AI 카메라의 고도화 및 연동 기술 개발, 과제 고유번호: 22TBIP-C161420-02]에 의해 수행되었습니다.

#### [참고 문헌]

- [1] G.H Kim, et al. "Improvement of Vehicle Position Estimation using Deep Learning-based Vehicle Underside Center Estimation", KSAE (2021): 931-932.
- [2] C. Sahin, "Comparison and Calibration of Mobile Phone Fisheye Lens and Regular Fisheye Lens via Equidistant Model", Hindawi 2016.
- [3] P. Beardsley, et al, "Camera Calibration Using Vanishing Points", BMVC 1992.