

후보 궤적을 이용한 비구조화된 환경에서의 교차로 인식 및 주행

김민수*¹⁾ · 안준우¹⁾ · 김민성¹⁾ · 박재흥^{1),2)}

서울대학교 지능정보융합학과¹⁾ · 차세대 융합 기술원²⁾

Intersection Detection and Navigation at Unstructured Environment using Candidate Trajectory

Minsoo Kim*¹⁾ · Joonwoo Ahn¹⁾ · Minsung Kim¹⁾ · Jaeheung Park^{1),2)}

¹⁾ Graduate School of Convergence Science and Technology, Seoul National University, Seoul, Korea.

²⁾ Advanced Institutes of Convergence Technology, Gwanggyo-ro, Yeongtong-gu, Suwon, Korea.

Abstract : In this paper, we present intersection detection and navigation method in the unstructured environment. The presented method uses only a camera image. It generates candidate trajectories in the segmented camera image and decides whether the vehicle enters an intersection using a histogram of the candidate trajectories. In addition, the vehicle selects the target trajectory and tracks it with considering the vehicle velocity, static obstacles, and navigation information. The proposed method was tested and evaluated in a simulation.

Key words : Intersection Detection(교차로 인식), Vehicle Navigation(차량 주행), Decision Making(판단 결정), Collision Avoidance(충돌 회피), Unstructured Environment(비-구조화된 환경)

1. 서론

최근 자율 주행 기술들이 상용화 되면서, 구조화된 도로뿐만 아니라 차선이 없는 골목길과 주차장 같은 비-구조화된 환경까지 다루는 기술들이 연구되고 있다.¹⁾⁻²⁾ 비-구조화된 환경에서는 교차로가 빈번하게 존재하며, 이를 인식하고 주행하는 방법이 필요하다. 기존의 차량 위치를 인지하는 로컬라이제이션(Localization)기반의 방법으로는 위 환경에서 교차로 인식이 어렵다. Global Positioning System (GPS)을 이용하여 로컬라이제이션을 하는 경우, 구조화된 환경에 비해 좁은 도로 폭과 도로들 간의 가까운 간격 때문에 차량의 위치 인식 오차가 크게 발생한다. 또한,

Simultaneously Localization And Mapping (SLAM)을 이용하였을 경우에는 빈번히 존재하는 코너 구간 때문에 로컬라이제이션의 방향 오차가 크게 발생한다. 따라서, 본 논문에서는 로컬라이제이션 정보 없이 카메라에 기반하여 비-구조화된 환경 내 교차로를 인식하고 주행하는 방법을 제시한다.

교차로를 인식하는 기존 연구로는 표지판 정보와 차선이 명확한 환경에서 딥러닝 기법을 이용한 방법이 있다.³⁾ 카메라 이미지를 입력으로, Recurrent Neural Network와 Convolutional Neural Network를 이용하여 이미지의 특징을 찾아내고 교차로 여부를 판단한다. 하지만, 주차장과 같이 차선이 없는 환경에서는 위 방법으로 교차로를 인식하기 어렵다. 다른 연구로는 주행 가능 영역이 구분된 전방 카메라

* 김민수, E-mail: msk930512@snu.ac.kr

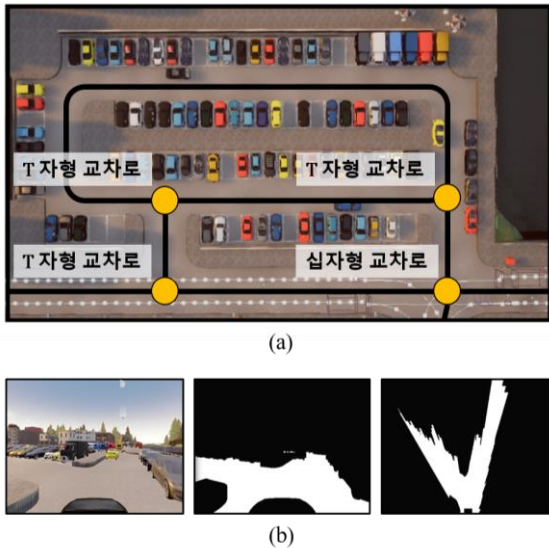


Fig. 1 CARLA 시뮬레이터에서 구축한 (a) 4개의 교차로가 존재하는 주차장 환경 및 (b) 차량의 전방 카메라 이미지(좌), 주행 가능/불가능 (흰색/검은색)영역이 구분된 이미지(중), Bird' s-eye view 변환 이미지(우).

라 이미지만을 이용하여 교차로를 인식하는 연구가 있다.⁴⁾ 전방 이미지에 차량을 기준으로 수직선을 그린 후, 선 위에 중심점을 정의한다. 중심점으로부터 0°에서 180°까지의 방향으로 직선을 그려 주변 장애물까지의 거리를 측정한다. 측정된 거리 정보를 Hidden Markov Model의 입력으로 이용하여 주행이 가능한 방향을 판단한 후, 주행 가능한 방향이 2곳 이상일 때 교차로를 인식한다. 위 방법은 차선이 명확하지 않은 환경에서도 교차로를 인식하지만, 이미지 내 중심점의 위치에 따라 교차로 인식이 안 되는 경우가 발생한다.

본 논문에서는 차량의 후보 궤적을 생성하여 교차로를 인식하고 주행하는 방법을 제안한다. 후보 궤적의 길이를 이용하여 주행 거리의 분포를 생성하고, 이 분포의 극대점의 개수를 이용하여 교차로를 인식한다. 교차로가 인식된 경우, 전역 계획 정보를 토대로 하나의 후보 궤적이 선택된다. 선택된 궤적을 추종하기 위해, 차량의 속도, 주변의 장애물 그리고 선택된 궤적의 방향을 고려한 목적함수가 설정되며, 목적함수를 최대로 하는 조향각과 속도를 결정한다. 본 연구는 시뮬레이션에서 비-구조화된 환경인 주차장을 구성하고, 제안된 방법에 대한 테스트 결과를 포함한다.

2. 교차로 주행 방법

제안하는 방법은 전방 카메라 이미지만을 이용한 다. 십자형 및 T자형 교차로(Fig. 1a)가 존재하는 환경을 가정하며, 이미지는 주행 가능/불가능 영역이 구분되는 Bird' s-eye view 이미지로 변환되어 사용된다(Fig. 1b). 본 절에서는 비-구조화된 환경에서의 교차로 주행 방법을 i) 후보 궤적을 이용한 교차로 인식 및 추종 궤적 선택과 ii) 궤적을 추종하는 주행 방법으로 나누어 설명한다.

2.1 교차로 인식 및 목표 궤적 설정

2.1.1 차량 모델을 이용한 후보 궤적 생성

차량의 모델 정보를 이용하여 차량 전방에 후보 궤적을 생성한다. 일반적으로 비-구조화된 환경에서 차량은 저속으로 이동하기에, 차량 모델은 Kinematic Bicycle Model⁵⁾을 사용한다. Kinematic Bicycle Model의 식은 다음과 같다.

$$\begin{aligned} \dot{x}_r &= v \cdot \cos(\theta_r) \\ \dot{y}_r &= v \cdot \sin(\theta_r) \\ \dot{\theta}_r &= \frac{v \cdot \tan\delta}{L} \end{aligned} \quad (1)$$

$\dot{x}_r, \dot{y}_r, \dot{\theta}_r$ 는 차량의 후륜(rear) 중심 기준, 움직임의 변화량으로 후륜 중심의 위치 및 방향을 의미하며, v 는 차량의 속도를 의미한다. δ 는 차량을 기준으로 한 바퀴의 각도이다. L 은 차량 전륜과 후륜 중심 사이의 거리를 의미한다.

식 (1)의 속도와 조향각을 고정하여 차량이 주행할 궤적을 생성하고, 이를 후보 궤적(τ)으로 정의한다. 생성된 후보 궤적(τ)은 Fig. 2a에서 녹색선으로 표시됐다. 이는 현재 차량의 상태에서 특정 조향각과 속도로 제어하였을 때, 장애물 지점까지 후륜 중심이 이동하게 될 궤적을 의미한다. 차량의 최소 조향각부터 최대 조향각까지 일정하게 증가시켜 위 과정을 반복하여 후보 궤적의 집합(T)을 정의한다.

2.1.2 후보 궤적 히스토그램을 이용한 교차로 인식

본 단락에서는 후보 궤적 집합(T)을 이용하여 후보 궤적 히스토그램을 생성하고, 교차로를 인식하는 방법을 설명한다. 교차로 인식의 전체 과정은 하단의 Algorithm 1과 같다.

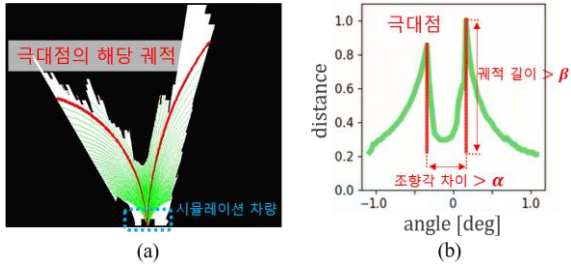


Fig. 2 후보 궤적을 생성하고, 이를 통해 교차로를 인식하는 과정. (a) 후보 궤적 생성, (b) 조향각에 따른 후보 궤적의 거리를 이용한 히스토그램. 각도 차이와 궤적 길이가 각각 α 와 β 이상을 만족하는 극대점 표시.

Algorithm 1: Intersection Detection method

Input: $T = \{\tau_i\}_{1..n}$, parameters α, β
 $H \leftarrow \{\text{Length}(\tau) | \tau \in T\}$
 $\{\tau_G\} \leftarrow \text{FindLocalMaximas}(H, \alpha, \beta)$
 $I \leftarrow \text{IntersectionDetection}(\{\tau_G\})$
return I

후보 궤적 집합(T)의 모든 후보 궤적(τ)의 길이가 계산된다 ($\text{Length}(\tau)$). 궤적의 길이는 Fig. 2a의 녹색선의 길이가 되며, 이는 최대값을 1.0로 갖는 정규화된 히스토그램으로 표현된다. 이 후보 궤적 히스토그램(H)은 차량이 해당 후보 궤적을 추종하였을 때, 궤적의 길이 분포를 의미하며 Fig. 2b에 표시하였다. 후보 궤적 히스토그램(H)에서 파라미터 α 와 β 를 이용하여 극대점을 찾는다. α 는 극대점에 해당하는 조향각끼리의 최소 차이를 의미하며, β 는 극대점에 해당하는 궤적의 최소 길이를 의미한다. 예를 들어, Fig. 2b에서와 같이 최소 길이(β) 이상인 극대점을 찾는다. 찾은 극대점에 해당하는 조향각의 차이가 최소 차이(α) 이상을 만족하는 극대점의 궤적을 목표 궤적(τ_G)으로 정의한다. 최소 차이(α) 이하의 경우는 궤적의 길이가 최대인 극대점만 목표 궤적(τ_G)이 된다. 이 목표 궤적(τ_G)은 Fig. 2a에 붉은색 선으로 표시되었으며, 해당 궤적을 따라 주행 가능한 방향이 존재함을 의미한다. 목표 궤적(τ_G)의 개수가 2개 이상일 때, 차량의 전방 도로는 교차로로 인식된다.

2.1.3 추종할 목표 궤적 결정 (τ_G^*)

교차로 인식에서 사용된 목표 궤적(τ_G)과 교차로에서의 전역 계획을 이용하여 추종할 목표 궤적



Fig. 3 십자형 교차로에서의 [직진, 좌회전, 우회전]로 구분된 목표 궤적. 전역 계획이 우회전임에 따라, 추종하게 될 우회전 궤적(τ_G^*).

을 결정한다. 본 논문에서는 십자형과 T자형 교차로를 고려하여, 목표 궤적을 [직진, 좌회전, 우회전]으로 구분한다. Fig. 3에서와 같이 차량의 현재 방향과의 차이가 가장 작은 궤적을 직진 궤적으로 정의한다. 이 직진 궤적을 기준으로 좌측의 목표 궤적은 좌회전 궤적이 되며 우측의 목표 궤적은 우회전 궤적이 된다. 교차로에서 전역 계획에 따라 [직진, 좌회전, 우회전] 궤적 중 하나가 선택되며, 이는 교차로에서 추종할 목표 궤적(τ_G^*) 이 된다.

2.2 목적 함수를 이용한 궤적 추종 방법

목표 궤적(τ_G^*)을 추종하기 위해, 목적 함수를 기반으로 조향각과 속도를 결정하는 방법이 사용된다. 전체 방법은 Algorithm 2와 같다.

Algorithm 2: Tracking method

Input: steering(θ), velocity(v), τ_G^*, θ_D, v_D
 $\theta_{cand} \leftarrow \{\theta | \theta \in [-\theta_{max}, \theta_{max}]\}$
 $v_{cand} \leftarrow \{v | v \in [v_{min}, v_{max}]\}$
 $\theta^*, v^* \leftarrow \text{ObjectiveFunction}(\theta_{cand}, v_{cand}, \tau_G^*, T)$
return θ^*, v^*

조향각과 속도 집합, θ_{cand}, v_{cand} 을 정의하여 목적 함수에서 탐색할 조향각 및 속도의 범위를 결정한다. 위 조향각(θ_{cand}) 및 속도(v_{cand}) 집합은 이산화된 조향각과 속도의 집합이며, 이 집합에서 목적 함수를 고려하여 목표 궤적(τ_G^*)을 추종하기 위한 조향각과 속도를 결정한다.

목적함수, J 는 차량의 속도, 장애물 그리고 목표 궤적의 방향을 고려하며 다음과 같이 정의된다.

$$J = k_v f_1(v) + k_f f_2(\theta, v, T) + k_h f_3(\theta, v, T, \tau_G^*) \quad (2)$$

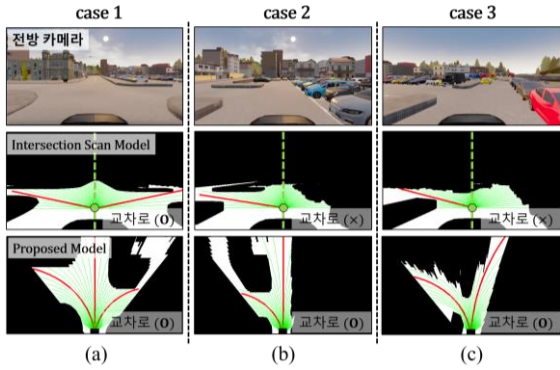


Fig. 4 제안한 방법의 교차로 인식 결과. 전방 이미지(상단), 기존 방법(중단), 제안한 방법(하단). (a) 십자형 교차로, (b) T자형 교차로, (c) T자형 교차로

f_1 은 정규화된 속도의 크기를 의미하며 최대 속도(v_{max})인 경우 1.0, 최소 속도(v_{min})인 경우 0.0의 값을 갖는다. f_2 는 장애물을 고려하는 함수이다. 이는 조향각(θ)과 속도(v)로 차량을 일정 시간 T 까지 제어하였을 때, 주행 궤적 내의 주행 가능한 영역의 비율을 의미한다. 예를 들면, Fig. 3에서 차량의 주행 궤적(파란색 선)에는 장애물이 존재하지 않는다. 따라서, f_2 값은 1.0이 된다. 함수 f_3 는 목표 궤적(τ_G^*)의 방향을 고려한다. 차량이 일정 시간 T 초 주행한 후의 차량의 방향과 목표 궤적(τ_G^*)의 끝에 도달하였을 때의 방향 차이를 의미한다. Fig. 3에서 목표 궤적(τ_G^*) 끝에서의 방향을 파란색 화살표로 나타냈다. 이 화살표의 방향과 차량 궤적(파란색 선) 끝에서의 방향의 차이가 180° 일 때, f_3 의 값은 0.0이 되며, 0° 일 때 1.0의 값을 갖는다. 식 (2)의 k_v, k_f, k_h 는 각 함수의 비율로 실험을 통해 결정된다. 위 정의된 목적 함수(J)를 최대화하는 조향각(θ^*)과 속도(v^*)로 차량을 제어하여 목표 궤적(τ_G^*)을 추종한다.

3. 교차로 인식과 주행 성능 평가 및 분석

3.1 시뮬레이션 환경

우리는 제안한 방법을 CARLA 시뮬레이터에서 연구소 내 주차장의 실측을 바탕으로 비-구조화된 환경을 구축하여 성능을 평가하였다 (Fig. 1a). 카메라는 차량의 무게 중심 대비 1[m] 위에 설치되었으며 5° 아래로 기울여 차량 양쪽 옆 영역도 인식이 되도록 하였다. 화각은 90° 이며, 이미지는 주

행 가능/불가능 영역이 구분된다. Bird's-eye view로 변환된 이미지는 약 전방 30[m]까지 관측이 가능하며, 제한한 방법의 입력이 된다(Fig. 1b). 실험에서 파라미터, α 와 β 는 각각 0.27, 0.40로 설정하였다. 최대 조향각(θ_{max})은 540° 로 정하였으며, 주차장 환경을 고려하여 최소 속도(v_{min})와 최대 속도(v_{max})는 3[km/h]와 8[km/h]로 정하였다. 식 (2)의 k_v, k_f, k_h 는 0.1, 1.0, 0.5로 설정하였으며, 시뮬레이션의 제어 주기는 0.2초로 고정하였다.

3.2 결과 및 분석

우리는 두 실험을 구성하여 제안한 방법을 평가하였다. 첫번째 실험으로 Fig. 1a의 교차로의 각 진입 도로에서 제안한 방법의 인식 성능을 기존 방법과 비교 및 분석한다. 두번째로 인식된 교차로에서 주행 성능 또한 정성적으로 평가한다.

Fig. 1a의 각 교차로의 진입 도로에서 교차로 인식률과 False Positive를 측정하였다. False Positive는 교차로가 인식되었을 때, 십자형 교차로를 T자형 교차로로, 혹은 그 반대로 인식한 횟수를 의미한다. 시뮬레이션 환경의 총 진입 도로의 수는 13개이며 인식률(Detection Rate)과 False Positive를 Table 1에 나타냈다.

Table 1 기존 방법(ISM)과 제안 방법의 교차로 인식 성능

	Detection Rate	False Positive
ISM ⁴⁾	84.6[%]	3회
Proposed Model	100.0[%]	1회

비교할 기존 방법(Intersection Scan Model, ISM⁴⁾)은 전방 이미지에서 차량을 기준으로 수직선(Fig. 4의 녹색 점선)을 그린 후, 일정 거리 앞에 중심점을 정의한다. 이 중심점에서 0° 에서 180° 방향으로 직선(녹색 선)을 그려 장애물까지의 거리를 측정하고, 주행이 가능한 방향(붉은색 선)을 판단하여 교차로를 인식한다. 본 실험에서 기존 방법(ISM)의 중심점은 교차로 인식률이 최대가 되도록 설정하였으며, 84.6[%]로 교차로가 인식되었다. 제안한 방법(Proposed Model)은 십자형 교차로를 성공적으로 인식하였으며 Fig. 4a에 나타냈다. 기존 방법 또한 교차로를 인식하였지만 주행 방향(붉은색 선)을 2 곳만 인식하여 십자형 교차로를 T자형 교차로로 인식하였다 (False Positive). Fig. 4b와 c와 같은 주차

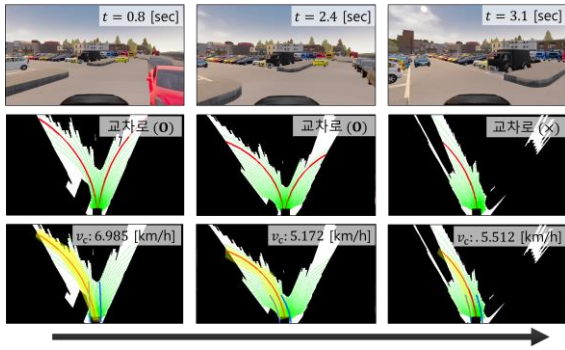


Fig. 5 제안한 방법의 교차로 인식 및 다음 도로 진입. 차량의 전방 이미지 (상단), 교차로 인식 (중단), 목표 궤적 (노란색)을 추종하기 위한 조향각과 속도 결정 (하단).

장 내 작은 교차로에서 기존 방법은 교차로를 인식하지 못하였다. 이는 중심점의 위치에 따라, 각 환경에서 교차로를 인식하는 성능의 차이가 존재하기 때문이다. 반면에 제안한 방법은 후보 궤적을 통해 실제로 차량이 주행 가능한 방향을 결정하여 교차로를 인식하였고, 실험 환경의 모든 진입 도로에서 교차로를 100[%]로 인식하였다.

Fig. 5에 제안한 방법으로 교차로를 인식($t=0.0s$)하고 다음 도로까지 진입하는 것을 나타냈다. 차량 전방에 후보 궤적(녹색 선)을 생성하고, 목표 궤적(붉은색 선)을 설정하여 교차로를 인식하였다. 제안한 주행 방법은 전역 계획이 '좌회전'인 것을 고려하여 추종할 목표 궤적(τ_c^*)을 결정하였으며, 차량의 속도, 주변 정적 장애물 그리고 궤적의 방향을 고려한 목적 함수로 조향각과 속도를 결정하였다. 교차로 진입 초기($t=0.8s$)에는 목표 궤적(τ_c^*)으로 바로 진입할 시 장애물과의 충돌 위험이 존재하여, 차량 궤적(파란색 선) 상에 장애물 비율이 적은 방향으로 주행하며 공간을 확보하였다. 교차로 내에서($t=2.4s$) 공간이 충분히 확보되어 목표 궤적(τ_c^*)의 방향으로 주행하며 다음 도로로 진입($t=3.1s$)하였다. 이로써, 제안한 궤적 기반의 방법으로 교차로를 인식할 수 있었고 성공적으로 교차로를 주행하였다.

4. 결론

본 논문에서는 비-구조화된 환경에서 교차로를 인식하고 주행하는 방법을 제안하였다. 제안한 방법은 로컬라이제이션 정보 없이, 차량의 전방 카

메라 이미지만을 이용한다. 주행 가능/불가능한 영역이 구분된 이미지에서 후보 궤적을 생성하여 교차로 진입 여부를 판단하였다. 또한, 전역 계획을 고려하여 목표 궤적을 정하고 목적 함수 기반의 주행 방법으로 정적 장애물을 회피하며 주행하였다. CARLA 시뮬레이터에서 제안한 방법의 교차로 인식 및 주행 성능이 평가되었다. 주행을 직선으로 가정 한 기존 방법 대비, 조향각까지 고려하여 실제 궤적으로 접근한 제안 방법의 교차로 인식률이 약 18[%] 상승하였으며, 목적 함수 기반의 방법으로 목표 궤적을 추종하며 다음 도로까지 진입할 수 있었다. 추후 계획으로는 실제 주차장 환경에서 교차로를 인식 및 주행하는 차량 실험 및 분석이 있다.

References

- 1) Do, Quoc Huy, Seiichi Mita, and Keisuke Yoneda. "Narrow passage path planning using fast marching method and support vector machine." *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014.
- 2) Zhang, Peizhi, et al. "Research on Parking Slot Tracking Algorithm Based on Fusion of Vision and Vehicle Chassis Information." *International Journal of Automotive Technology* 21.3 (2020): 603-614.
- 3) Bhatt, Dhavit, et al. "Have i reached the intersection: A deep learning-based approach for intersection detection from monocular cameras." *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017.
- 4) Yi, Yang, et al. "Intersection scan model and probability inference for vision based small-scale urban intersection detection." *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017.
- 5) Kim, Minsung, Gyubeom Im, and Jaeheung Park. "A Comparative Study of Parking Path Following Methods for Autonomous Parking System." *The Journal of Korea Robotics Society* 15.2 (2020): 147-159.