

# 실시간 이미지 처리 방법을 이용한 개선된 차선 인식 경로 추종 알고리즘 개발

서은빈\* · 이승기\* · 여호영\* · 신관준\* · 최경호\*\*,<sup>†</sup> · 임용섭\*\*\*,<sup>‡</sup>

## Development of an Improved Geometric Path Tracking Algorithm with Real Time Image Processing Methods

Eunbin Seo\*, Seunggi Lee\*, Hoyeong Yeo\*, Gwanjun Shin\*,  
Gyeongho Choi\*\*,<sup>†</sup>, Yongseob Lim\*\*\*,<sup>‡</sup>

*Key Words: Pure pursuit(퓨어 퍼슈트), Segmentation(영상 분할), Dataset Augmentation(데이터셋 증강), Post processing(후처리), Look-ahead distance(전방 주시 거리), Smoothing filter(스무딩 필터)*

### ABSTRACT

In this study, improved path tracking control algorithm based on pure pursuit algorithm is newly proposed by using improved lane detection algorithm through real time post-processing with interpolation methodology. Since the original pure pursuit works well only at speeds below 20 km/h, the look-ahead distance is implemented as a sigmoid function to work well at an average speed of 45 km/h to improve tracking performance. In addition, a smoothing filter was added to reduce the steering angle vibration of the original algorithm, and the stability of the steering angle was improved. The post-processing algorithm presented has implemented more robust lane recognition system using real-time pre/post processing method with deep learning and estimated interpolation. Real time processing is more cost-effective than the method using lots of computing resources and building abundant datasets for improving the performance of deep learning networks. Therefore, this paper also presents improved lane detection performance by using the final results with naive computer vision codes and pre/post processing. Firstly, the pre-processing was newly designed for real-time processing and robust recognition performance of augmentation. Secondly, the post-processing was designed to detect lanes by receiving the segmentation results based on the estimated interpolation in consideration of the properties of the continuous lanes. Consequently, experimental results by utilizing driving guidance line information from processing parts show that the improved lane detection algorithm is effective to minimize the lateral offset error in the diverse maneuvering roads.

### 1. 서론

최근 자율주행 자동차는 고속도로에서 운전자를 보조하는 단계를 넘어서 도심과 골목길 같은 도심도로에서 저속으로 주행해야 하는 자율주행으로 발전하고 있다. 고속도로나 자동차 전용도로와 같이 차선의 구성이 단조로운

\* 대구경북과학기술원(DGIST) 융복합대학, 학부생

\*\* 대구경북과학기술원(DGIST) 융합전공, 교수

\*\*\* 대구경북과학기술원(DGIST) 로봇공학전공, 부교수

<sup>†</sup>공동 교신저자: ghchoi@dgist.ac.kr

<sup>‡</sup>공동 교신저자: yslim73@dgist.ac.kr

E-mail: seb5428@dgist.ac.kr

곳에서는 차선 인지 및 차량제어의 난이도가 낮으나 곡률이 심한 도심 곡선 구간에서 안정적인 주행을 위하여 고도의 알고리즘이 필요하다.

지금까지 제안된 경로 추종 알고리즘 방법에는 크게 기하학적 경로 추종법과 모델기반 경로 추종법이 있으며,<sup>(1)</sup> 기하학적 경로 추종법에는 퓨어 퍼슈트(pure pursuit), 스탠리 방법(stanley method), 그리고 벡터 퍼슈트(vector pursuit)이 있다.<sup>(2-4)</sup> Pure pursuit은 다른 추종법에 비해 간단하고, 큰 오차에 대해 비교적 성능이 좋은 편으로 알려져 있다.<sup>(1)</sup> 그러나 기존의 pure pursuit은 차량속도가 빠를 때 추종 성능이 저하되거나, 전방 주시 거리(look-ahead distance)에 차량속도와 관련한 항이 있을 경우 계산된 조향각이 차량 속도에 민감해져 차체가 불안정하게 요동치는 경향이 있다.<sup>(5)</sup>

본 논문은 차선 유지 시스템의 전체적인 구조와 각 알고리즘 간의 데이터 통신 및 후처리 방식에 대해서 개선된 알고리즘을 제안하고자 한다. 이를 탑재한 차량 플랫폼은 디지스트(DGIST) 캠퍼스내에서 비전 중심의 차량 주행을 안정적인 운행을 수행하였다.

## 2. Path tracking algorithm

### 2.1. Original Pure Pursuit

Pure pursuit은 경로를 이루는 목표점(Target point)을 따라가는 경로 추종 알고리즘 중 하나이다. 이는 Bicycle Model을 기반으로 만들어진 알고리즘으로 다른 알고리즘에 비해서 간단하며 자율주행 기술에 많이 사용되어 왔다. 알고리즘 설명을 위해 도식화한 것을 Fig. 1(a), (b)에 나타냈다. 목표점과 차량의 후륜이 이루는 각도를  $2\alpha$  그리고 회전반경(R)을 가진 부채꼴을 그린다. 그 후에 부채꼴을 기준으로 차량 전륜의 조향각( $\delta$ )을 식 (1)과 같이 계산하게 된다.

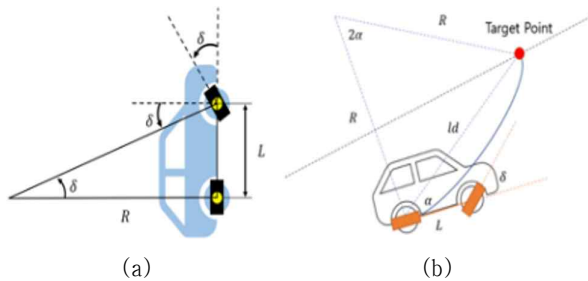


Fig. 1 Pure Pursuit Geometry

$$\delta = \tan^{-1} \frac{L}{R} \quad (1)$$

$$\frac{ld}{\sin(2\alpha)} = \frac{R}{\sin\left(\frac{\pi}{2} - \alpha\right)} = \frac{R}{\cos(\alpha)} \quad (2)$$

$$\therefore \frac{1}{R} = \frac{2\sin(\alpha)}{ld} = k(k = \text{curvature}) \quad (3)$$

식 (2)는 Fig. 1의 (a)에서 사인법칙을 사용하면 유도할 수 있다. 식 (3)에서 구한 R을 식 (1)에 대입하여 정리하면, 최종적으로 차량의 조향각을 식 (4)와 같이 구할 수 있다.

$$\delta = \arctan\left(\frac{2L\sin\alpha}{ld}\right) \quad (4)$$

식 (4)의 ld는 전방 주시 거리, L은 차축거리를 의미한다. 보통 전방 주시 거리는 속도에 대한 일차함수로 나타내지만 실제로 주행을 해보면 속도에 대한 이차함수나 시그모이드(Sigmoid)함수로 나타냈을 때 조금 더 향상된 성능을 보였다. 따라서 본 논문은 일차함수 아닌 다른 전방 주시 거리 식을 사용하였다.

### 2.2. Proposed Pure Pursuit

본 절에서는 앞서 설명한 pure pursuit에 전방 주시 거리와 스무딩 필터를 어떤 방식으로 적용했는지 서술하고자 한다.

#### 2.2.1. Look-Ahead Distance

본 논문에서 사용한 전방 주시 거리는 두 가지 종류가 있다. 한가지 수식은 식 (5)와 같은 이차함수<sup>(6)</sup>이다.

$$ld = Av^2 + Bv + C + DN \quad (5)$$

where A: Maximum braking acceleration,

B: Constant

C: Minimum turning radius of the vehicle

D: Sensitivity level which depends on N

N: Vertical distance between the vehicle and ideal path

주행경로를 비교한 논문<sup>(6,7)</sup>을 참고하여 실제 주행 시, 전방 주시 거리에 대하여 이차함수가 일차함수를 적용한

경우보다 추종 성능이 더 좋음을 확인하였다. 하지만 이 차함수를 사용한 경우에도 만족스러운 추종 성능을 얻기에는 힘들었다. 따라서, 본 연구에서는 식 (6)과 같은 시그모이드 함수를 또 다른 수식으로 사용하였으며, 이를 식 (5)의 이차함수 결과와 비교하였다. 시그모이드 함수가 이차함수에 비해 추종 성능이 더 좋았으며, 결과는 4.3절에서 설명하고자 한다.

$$ld = 190 + 100 \times \left[ \frac{1}{\exp\left\{-\left(\frac{v-20}{15}\right)\right\}} - 0.5 \right] \quad (6)$$

### 2.2.2. Smoothing Filter

스무딩 필터(Smoothing filter)란 pure pursuit을 통해 계산된 조향각이 갑자기 증가하거나 감소할 때 핸들의 뒤틀림을 방지하기 위한 필터이다. 적절한 전방 주시 거리를 선택하였음에도 차량 조향각의 불안정성이 존재하였기에 스무딩 필터를 이용하여 조향의 떨림을 완화시키고자 하였다.

---

*Algorithm 1:* Filter to smooth the steer

---

**Input:** One float steer value **a**

**Output:** One float steer value with filter applied **b**

**function** Filter(a);

    integral += a \* (time difference);

    diff = (a - prev\_a) / (time difference);

    if integral > constant;

        integral = constant;

    elif integral < - constant;

        integral = - constant;

    b = k<sub>1</sub> \* a + k<sub>2</sub> \* integral + k<sub>3</sub> \* diff;

    prev\_a = a;

**return** b;

---

Code. 1 Pseudo code for the smoothing filter

## 3. Lane Detection

### 3.1. Deep Learning based Lane Detection Algorithm

본 논문에서는 Hou의 ENet-SAD<sup>(8)</sup>를 PyTorch로 구

현하여 차선인지 알고리즘으로 사용하였다. Hou가 공개한 베이스 코드에서 몇몇 데이터의 병목현상을 줄이기 위해 SAD 모듈 및 CNN 모듈에 대한 데이터를 GPU에서 직접 처리하도록 커스텀 모듈로 묶어서 구현하였다. 이를 통해서 공식적으로 배포한 Repo의 성능보다 5배 더 높은 추론 속도를 얻을 수 있었다.

### 3.2. Pre-processing for an improved performance of deep learning network

전처리 방법은 차선인지 딥러닝 네트워크에서 좋은 결과를 얻기 위해서 데이터셋을 가공하는 방법으로 ENet-SAD 학습 전 단계이다. 본 연구에서는 전처리로 데이터셋 증강(Augmentation) 방식을 적용했고, 데이터 라벨링 방식을 고려하였다.

#### 3.2.1. Dataset Augmentation

딥러닝 네트워크에서 많은 매개변수를 학습할 때, 양질의 데이터셋을 많이 확보할수록 과적합(overfitting)이 방지되어 결과적으로 다양한 환경에서 강건하게 인식할 수 있도록 도와준다. 즉, 날씨변화(비, 눈, 안개), 태양의 위치에 따른 이미지 변화(태양 플레어, 그림자), 이미지 보정(RGB shifting, 감마, 밝기), 모션 블러, 해상도 변화로 총 11가지에 대하여 데이터를 증강시켰다.

그 중 본 연구에서는 인식이 힘든 그림자에 집중하였고 데이터증강으로 개선시키고자 하였다. 날씨가 화창한 날에도 많은 나뭇잎이 도로 위에 그림자가 투영되면서 차선에 불규칙적인 경계가 만들어져 딥러닝 네트워크의 성능이 좋지 않았다. 불규칙적인 경계가 만들어진다는 점에 초점을 맞춰 Fig. 2의 (a)와 같이 날씨가 흐린 날의 이미지에 여러 개의 작은 그림자를 생성하였다. 결과적으로 (a)가 (b)처럼 화창한 날의 이미지와 비슷한 이미지가 되었고, 이러한 데이터셋 증강은 딥러닝 네트워크의 성능향상에 도움을 주었다.



Fig. 2 Comparison of images after the shadow augmentation and images on a sunny day

### 3.2.2. Ground Truth Labeling

딥러닝 네트워크가 학습하는데 각 이미지당 차선의 정답 정보를 제공하는 라벨링이 필요하다. ENet-SAD가 학습할 데이터셋을 제작하기 위해 CULane 호환 라벨링 툴을 개발하였다. 차선 정보가 들어있는 영상 분할(Segmentation) 이미지와 각 차선이 왼쪽-왼쪽 차선, 왼쪽 차선, 오른쪽 차선, 오른쪽-오른쪽 차선 중 어떤 차선인지 정보가 들어있는 이미지들에 대한 목록으로 이루어져 있다.

카메라 특성상 차선의 소실점으로 가까워질수록 차선의 두께가 얇아진다. 실제 이미지와 영상 분할 이미지 정보가 비슷할수록 성능 향상에 도움을 준다. 따라서 차선에 대한 영상 분할 방법을 반영하여 ENet-SAD를 학습시켰다. Fig. 3의 (a)는 원본 이미지이고, (b)는 (a)에 대한 차선 영상 분할 이미지로 ENet-SAD가 최종적으로 학습할 이미지이며, (b)에서 소실점에 가까워질수록 점점 얇아지는 것을 볼 수 있다.



Fig. 3 Example of segmentation image

### 3.3. Real-Time Lane Fitting Algorithm

딥러닝 네트워크에서 나오는 영상 분할 이미지를 직접적으로 주행하는데 사용하기 어렵다. 따라서 본 논문에서는 실시간 차선 적합(Real-time lane fitting) 알고리즘을 제안한다. 실시간 차선 적합 알고리즘은 딥러닝 네트워크에서 추론한 차선 영상 분할 이미지에 대하여 차선 정보를 가공하여 주행 유도선을 제공하는 알고리즘이다. 실시간 차선 적합 알고리즘은 역투영 변환과 앙상블(Ensemble) 방법을 이용한 보간 알고리즘으로 구성되어 있다. 역투영 변환을 적용하여 2차원 평면에서 투영 변환 행렬을 구하고 차선이 있는 영역을 4개 점으로 매핑하여 평면도 영상으로 처리한다. 역투영 변환을 거친 영상분할 이미지의 픽셀이 각 차선이 어떤 차선인지 확인하고, 차선마다 픽셀 좌표에 대하여 추정보간 알고리즘을 적용한다. 본 연구에서 활용한 추정보간 알고리즘은 선형 보간법, 다항식 보간법, 그리고 큐빅 스플라인 보간법<sup>(9)</sup>이다.

세 가지 보간법 알고리즘으로 차선 적합 식을 구한 후 앙상블 방법으로 차선 정보의 분산 값이 가장 작은 차선 적합 식을 택하여 최종적 차선 정보로 결정하였다. 최종적으로 결정된 차선들은 차선의 폭이 일정하다는 성질을 이용하여 주행유도선을 점들의 집합으로 경로 추종 알고리즘에 사용된다.

## 4. Field Test

본 논문에서 작성된 알고리즘은 디지스트 캠퍼스를 주행하는 자율주행 차량에 탑재되었다. 디지스트 캠퍼스는 왕복 2차선, 4차선 도로와 다양한 곡률의 도로로 구성되어 있어 새로운 알고리즘이 탑재된 자율자동차를 검증하기에 최적의 도로 조건이라고 판단하였다. 본 연구에서는 현대 아이오닉 전기자동차를 자율주행 차량으로 개조하여 사용하였으며, 카메라는 차량 정면을 기준으로 중앙에, 높이 155cm, 차량의 앞부분에서 213cm떨어진 곳에 부착하였다.

Table 1 Test Course Detail

	회전 반경(R)	구간	제한 속도
Straight	-	123.1 m	30KPH
Curve 1	120.7	165.1 m	30KPH
Curve 2	30.3	89.7 m	30KPH

필드 테스트는 디지스트 캠퍼스 경로를 따라 주행하였고 그 중에서 위 알고리즘의 성능을 확인할 수 있는 직선 구간 1곳과 곡선 구간 2곳을 선정하였으며, Table 1에 주행 경로에 대한 회전반경, 구간거리 등의 정보를 나타내었다.

### 4.1. 차선 인식 결과

#### 4.1.1. Augmentation에 따른 결과

그림자와 관련된 이미지 증강을 진행한 결과 성능 차이를 볼 수 있었다. Fig. 4의 (a)는 여러 개의 작은 그림자에 대한 데이터셋 증강을 포함하지 않은 학습 가중치에 따른 결과이고 (b)는 그림자 데이터셋 증강을 포함한 학습 가중치에 따른 결과이다. Fig. 4에서 확인할 수 있듯이, 그림자 데이터셋 증강을 제외하고 학습 데이터셋과 검증 데이터셋을 동일하게 학습시켰을 때, 그림자에 대한



Fig. 4 The result images with and without shadow augmentation respectively

데이터셋을 추가해 학습시킨 것이 더 나은 차선 추론 성능을 보였다.

#### 4.1.2. 전/후처리 이후 차선 인식 결과

카메라에서 들어온 미가공 데이터를 3장에서 제안한 알고리즘을 모두 적용한 후 차선이 인식된 결과이다. 차선이 끊기지 않고 어떤 차선인지 혼동하지 않으며 차선의 두께와 비슷하게 인식한다. Fig. 5의 (a)는 앞서 3장에서 소개한 처리 방식을 모두 적용하지 않은 이미지이고, (c)는 모두 적용한 결과이다. (b)는 (a)의 박스 영역을 확대한 이미지이고, (d)는 (c)의 (b)와 같은 영역의 이미지이다. Fig. 5의 (b)에서 차선을 끊어 인식하거나 차선의 종류를 혼동한 결과를 볼 수 있다. 반면에 Fig. 5의 (c)와 (d)를 보면 차선을 혼동하지 않고 각각의 차선을 잘 인식하고 있는 것을 볼 수 있으며, 추정정보간섭을 활용함으로써 끊겨 있는 차선을 잘 연결해 인식하였다.

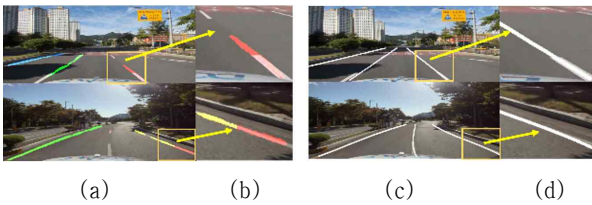


Fig. 5 Comparison of the images without the lane detection algorithm and images with algorithm

### 4.2. 차선 추종 주행 결과

#### 4.2.1. Look-Ahead Distance Comparison

전방 주시 거리의 종류에 따른 실제 이동 경로와 이상적 경로(Ideal path)의 오차(lateral offset)에 대한 평균과 분산을 Table 2에 나타냈다.

이때, 오차는 이상적 경로와 가장 가까운 실제 이동 경로의 점 사이 거리로 정의하였다.<sup>(6)</sup> 오차의 평균과 분산

Table 2 Look-ahead Distance Comparison

	Second order LD Function	Sigmoid LD Function
Mean (m) ↓	0.579	<b>0.488</b>
Variance ↓	0.0353	<b>0.0307</b>

은 이차함수를 사용하였을 때 보다 시그모이드 함수를 사용하였을 때 각각 0.091, 0.0046만큼 작으므로 전방 주시 거리 계산시 시그모이드 함수를 사용하였을 때 추종 성능이 더 뛰어나다는 것을 알 수 있다. 또한 시그모이드 함수를 사용하였을 때 평균과 분산 자체의 값이 0.488, 0.0307로 매우 작았기 때문에 이를 경로 추종 알고리즘으로 사용하기 적합하다고 판단되었다.

#### 4.2.2. Proposed Pure Pursuit with Smoothing Filter

이상적 경로에 대하여 pure pursuit의 스무딩 필터가 여부에 따른 경로를 비교하였다. 이때, 전방 주시 거리는 4.3.1항에서 추종 성능이 더 좋았던 시그모이드 함수를 이용하였다. Fig. 6의 (a)는 직선 구간과 곡선 구간에 대한 전체 경로이고, (b)는 (a)의 박스 친 영역을 확대한

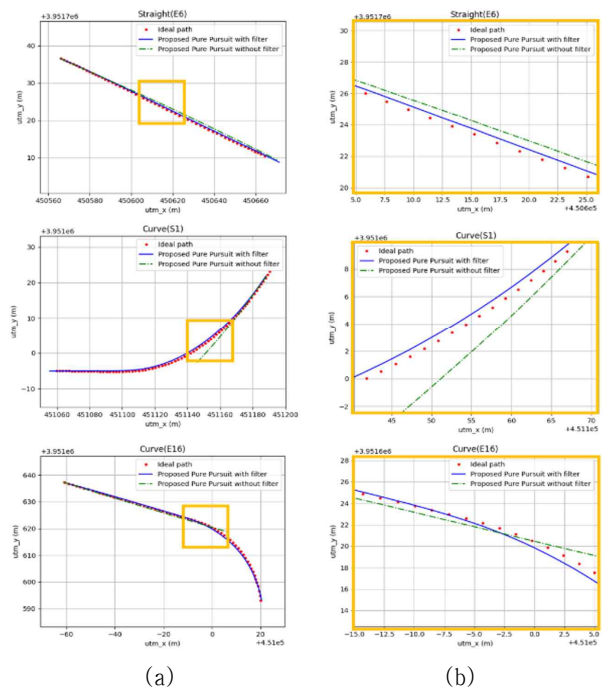


Fig. 6 Actual driving path according to the presence of filters added to pure pursuit

그림이다. (b)를 보면, 스무딩 필터를 사용하였을 때의 이동경로가 사용하지 않았을 때보다 이상적 경로에 근접하다는 것을 시각적으로 확인할 수 있다. 스무딩 필터의 추가 여부에 따른 추종 성능 비교를 위하여 이상적 경로에 대한 이동 경로의 오차를 구하였다. 곡선구간에서 필터를 추가하지 않았을 때 오차 평균은 1.584m 이상으로, 정상적인 주행이 불가능하였다. 반면, 스무딩 필터를 추가했을 때 곡선구간에서 오차 평균은 0.488m, 분산은 0.0307으로 좋은 성능을 보였다.

추종 성능에 대한 평가는 경로 그래프로도 확인할 수 있지만, 조향각의 그래프로도 확인할 수 있었다.

Fig. 7의 (a)는 스무딩 필터를 적용하지 않았을 때, (b)는 스무딩 필터를 적용하였을 때의 조향각을 표시한 그래프이다. 두 개의 그래프를 비교해보면, 스무딩 필터를 추가하였을 때 같은 값이 여러 번 들어오는 것을 방지하여 조향각의 떨림이 많이 줄었고, 조향 안전성에 더 가까운 것을 확인할 수 있다. 최종적으로 본 논문이 제안한 pure pursuit이 경로 추종에 적절하다고 할 수 있다.

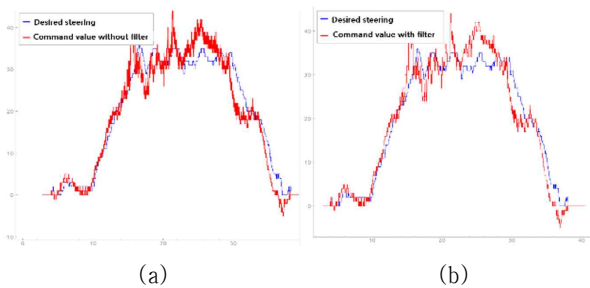


Fig. 7 Steer graph according to the presence of filters added to pure pursuit

## 5. 결 론

본 논문에서는 추정보간 실시간 후처리 방법을 통해 딥러닝 네트워크에서 차선 정보가 지연없이 차량의 제어에 사용할 수 있도록 알고리즘을 제안하였고 이를 이용한 경로 추종 알고리즘을 제안하였다. 제안된 알고리즘을 적용하여 다음과 같은 성공적인 결과를 얻을 수 있었다.

- 1) 제안한 알고리즘들을 적용한 자율주행차량을 디지털 캠퍼스에서 실주행을 통해 검증하였다. 이상적 경로와 차로의 중심과의 오차는 평균 0.488m, 분산 0.0307으로, 이상적 경로에 밀접하게 추종한 것을 확인하였다.

- 2) 도심구간의 저속에 대응하기 위해 시그모이드 형태의 전방 주시 거리 함수와 스무딩 필터를 제안했다. 그래프를 통하여 조향각의 떨림이 줄어들었고 조향안정성을 확인하였다.
- 3) 최종적으로 후처리 알고리즘의 속도를 높여 인식 지연을 줄였으며 조향 안정성을 확보한 제안한 pure pursuit 알고리즘을 이용하여 차량의 주행을 안정적으로 성공하였다.

## 후 기

본 연구는 과학기술정보통신부에서 지원하는 DGIST 기관고유사업에 의해 수행되었다(21-BRP-08 & 21-BRP-09).

## 참고문헌

- (1) Park, M., Lee, S., and Han, W., 2014, "Development of lateral control system for autonomous vehicle based on adaptive pure pursuit algorithm", 2014 14th International Conference on Control, Automation and Systems (ICCAS 2014), Seoul, pp. 1443~1447.
- (2) Wallace, R. S., Stentz, A., Thorpe, C. E., Moravec, H. P., Whittaker, W., and Kanade, T., 1985, "First Results in Robot Road-Following", International Joint Conference on Artificial Intelligence (IJCAI), pp. 1089~1095.
- (3) Hoffmann, G. M., Tomlin, C. J., Montemerlo, M., and Thrun, S., 2007, "Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing", 2007 American Control Conference, pp. 2296~2301.
- (4) Wit, J., Crane III, C. D., and Armstrong, D., 2004, "Autonomous ground vehicle path tracking", Journal of Robotic Systems, Vol. 21, No. 8, pp. 439~449.
- (5) Martínez, J. L., Morales, J., Mandow, A., and García-Cerezo, A., 2008, "Steering limitations for a vehicle pulling passive trailers", IEEE Transactions on Control Systems Technology, Vol. 16, No. 4, pp. 809~818.
- (6) Wang, W. J., Hsu, T. M., and Wu, T. S., 2017, "The improved pure pursuit algorithm for auto-

- nomous driving advanced system”, 2017 IEEE 10th International Workshop on Computational Intelligence and Applications (IWCI), pp. 33~38.
- (7) Wang, R., Li, Y., Fan, J., Wang, T., and Chen, X. 2020, “A Novel Pure Pursuit Algorithm for Autonomous Vehicles Based on Salp Swarm Algorithm and Velocity Controller”, IEEE Access, 8, pp. 166525~166540.
- (8) Hou, Y., Ma, Z., Liu, C., and Loy, C. C., 2019, “Learning lightweight lane detection cnns by self attention distillation”, In Proceedings of the IEEE International Conference on Computer Vision, pp. 1013~1021.
- (9) Gasca, M., and Sauer, T., 2000, “Polynomial interpolation in several variables”, Advances in Computational Mathematics, Vol. 12, No. 4, p. 377.