

군집주행 차선인지를 위한 슬라이딩 윈도우 보정

김 태 욱¹⁾ · 이 현 규¹⁾ · 김 종 찬²⁾

국민대학교 자동차공학전문대학원¹⁾ · 국민대학교 자동차IT융합학과²⁾

Sliding Window Correction for Platooning Lane Detection

Tae-Wook Kim¹⁾ · Hyeongyu Lee¹⁾ · Jong-Chan Kim^{*2)}

¹⁾Graduate School of Automotive Engineering, Kookmin University, Seoul 02707, Korea
²⁾Department of Automobile and IT Convergence, Kookmin University, Seoul 02707, Korea
(Received 19 April 2021 / Revised 16 July 2021 / Accepted 17 July 2021)

Abstract : Considering fuel economy, the environment, and safety, research on truck platooning has been steadily progressing. The autonomous driving technology of platooning differs from general autonomous driving due to the nuances of driving. Platooning driving keeps a close distance of about 12.5 m while driving. At this time, a lane blocking phenomenon occurs when there is a truck in front. Furthermore, the dotted line section of the highway consists of a 12 m lane-free section, making it possible to acquire only fairly limited, actual lane data. Due to such limited data, errors may occur in the process of setting the sliding window position during the lane recognition process. In particular, more problems can arise in the curved section, leading to a degradation in lane detection accuracy. In this paper, we solved this problem through sliding window double correction, confirming an accuracy improvement of more than 10 %.

Key words : Autonomous vehicle(자율 주행차), Platooning(군집주행), Lane detection(차선인지), Sliding window(슬라이딩 윈도우), Scale truck(스케일 트럭)

1. 서론

자율주행 자동차의 한 분야인 군집주행(Truck platooning system)에 관한 연구가 꾸준히 진행되고 있다. 군집 주행은 연비 절감 및 이산화탄소 배출 감소 그리고 안전성 향상 등의 이유로 많은 관심을 받고 있으며, 트럭 산업 분야에서 제기되고 있는 인력 부족 문제를 해결 할 수 있는 대안으로 주목받고 있다.^{1,2)} 유럽과 미국을 중심으로 1990년대부터 꾸준히 이어져 온 연구는 유럽 위원회의 자금을 바탕으로 진행된 FP7 프로젝트인 SARTRE, 2008년부터 시작된 “Energy ITS” 프로젝트, 2016년 유럽대륙을 횡단하는 유럽 트럭 플래투닝 챌린지(European truck platooning challenge) 그리고 GCDC, SCANIA와 같은 프로젝트로 진행이 되었다.^{3,4)} Fig. 1은 2021년 시험 운행 목표로 한국도로공사, 국민대학교, 현대 자동차등이 산학연 협동으로 진행하고 있는 프로젝트의 시스템 구조

로 국내에서도 군집주행에 관한 연구가 진행되고 있으며, 본 논문 또한 해당 프로젝트의 기술 개발 일환으로 진행되었다.

트럭 군집 주행은 여러 대의 트럭이 V2X 통신을 바탕으로 좁은 거리를 유지하며 주행하는데, 고속도로 화물

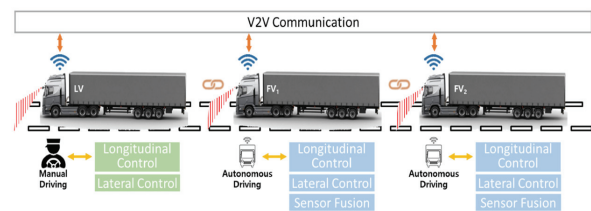


Fig. 1 Platooning system architecture of the project “Development of Safety Driving Control Technology for Platooning Truck”⁵⁾

*Corresponding author, E-mail: jongchank@kookmin.ac.kr

⁵⁾This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(http://creativecommons.org/licenses/by-nc/3.0) which permits unrestricted non-commercial use, distribution, and reproduction in any medium provided the original work is properly cited.

차 최고 속도 90 km/h 기준 0.5초 Time-gap 적용 시 차간 거리 12.5 m를 유지하며 주행하게 된다.^{5,7)} 이때 FV (Following Vehicle)는 LV(Leading Vehicle)에 의해 많은 시야 가림 현상이 발생하게 된다.⁸⁾ 이는 자율 주행 인지 과정을 더욱 어렵게 만드는데, 특히 횡 방향 제어에 필요한 차선인지 과정에 많은 영향을 준다.⁹⁾ 일반적인 차선 인지는 카메라 센서를 통해 들어온 이미지에서 소실점까지의 데이터를 기반으로 차선을 검출한다.¹⁰⁾ 그러나 군집 주행의 경우 대부분의 차선이 LV에 의해 가려지게 되고, 그로 인해 제한된 데이터만을 가지고 차선을 검출해 내야한다.

군집 주행이 이루어지는 고속도로의 차선은 많은 부분이 점선으로 되어 있으며, 점선 구간은 점선 길이 8 m, 점선간 간격 12 m로 이루어져 있다.¹¹⁾ 차간 거리가 12.5 m 까지 줄어드는 군집 주행 상황에서는 얻어지는 데이터에서 차선이 없는 부분이 대부분을 차지하는 상황이 계속해서 발생하게 된다. 이렇게 제한된 데이터만을 가지고 이루어지는 차선인지는 결국 정확도 저하로 이어지게 된다.¹²⁾ 군집 주행의 차선인지 과정도 일반적인 차선 인지와 같은 과정을 통해 이루어지는데, 본 연구에서는 차선인지 여러 과정 중 슬라이딩 윈도우 과정에서 발생할 수 있는 위치 설정 오류를 보정하는 방식을 통해 인지 정확도를 개선하고자 한다.

이와 함께 본 논문에서는 알고리즘 테스트를 위한 테스트 베드에 대해서도 자세히 다루고 있다. 실제 고속도로에서 여러 대의 트럭을 이용한 테스트는 여러 환경적 문제로 인해 그 한계가 있다. 그래서 다양한 환경과 좀 더 자유로운 테스트를 진행하기 위하여 1/14 스케일의 트럭과 트랙을 제작하여 테스트를 진행하였다.

본 논문의 2장에서는 슬라이딩 윈도우 기법에 관한 배경 지식 및 문제 정의에 대해 서술한다. 3장에서는 정확도 향상을 위한 슬라이딩 윈도우 이중 보정 기술을 제시한다. 그리고 4장에서는 두 가지 테스트 환경과 방법 및 실험 결과에 대해 설명하고, 마지막으로 결론 및 향후 연구에 대해 5장에서 논한다.

2. 배경 지식 및 문제 정의

2.1 슬라이딩 윈도우 기법

차선인지 알고리즘은 다음과 같은 과정을 통해 진행된다. 먼저 카메라를 통해 들어온 전체 이미지에 노이즈 제거와 처리 속도 향상을 위해 ROI(Region of Interest) 설정한다. 다음 효과적인 차선 검출을 위해 ROI를 통해 선택된 Original 이미지를 Bird's eye view 이미지로 변환한다.¹³⁾ 변환된 이미지에서 차선을 특징하기 위해 Edge

검출 및 색의 채도 차이를 비교하는 방법을 통해 Bird's eye view 이미지를 Binary 이미지로 변환한다.¹⁴⁾ 다음 슬라이딩 윈도우 기법과 Histogram을 통해 차선을 검출한다.¹⁵⁾

슬라이딩 윈도우 기법과 Histogram을 통해 차선을 검출하는 방법은 다음과 같다. Binary 이미지에서 x축을 기준으로 각 좌표의 차선 픽셀 수를 나타내주는 Histogram을 통해 좌우 차선의 기준이 될 수 있는 베이스 점을 찾는다.¹⁶⁾ 베이스 점을 중심으로 지정된 폭의 첫 번째 슬라이딩 윈도우를 설정한다. 다음 첫 번째 슬라이딩 윈도우 내의 차선 픽셀들의 x좌표 평균값을 구하고, 그 값을 중심으로 첫 번째 슬라이딩 윈도우와 같은 크기의 두 번째 슬라이딩 윈도우를 쌓는다. 이 과정을 반복하여 하나의 Binary 이미지에 있는 전체 차선을 검출한다.

2.2 문제 정의

서론에서 간략히 설명했듯이 군집 주행 차량은 좁은 거리를 유지하며 주행하기 때문에 센서를 통해 들어오는 차선 이미지에 차선이 없는 부분이 상당 부분을 차지한다. 각각의 슬라이딩 윈도우는 이전 슬라이딩 윈도우 내의 차선 데이터의 x축 평균값을 기준으로 다음 슬라이딩 윈도우를 쌓는 방식으로 진행되는데, 이때 이전 슬라이딩 윈도우의 위치가 차선이 없는 부분에 있게 되면, 슬라이딩 윈도우내에 평균을 낼 차선 데이터 자체가 없게 된다. 그럼 일반적인 알고리즘은 다음 슬라이딩 윈도우를 기존 슬라이딩 윈도우와 같은 위치에 그대로 쌓게 되는데, 이 때 빈 구간만큼의 슬라이딩 윈도우가 y축 일직선으로 쌓이게 된다.

이렇게 일직선으로 슬라이딩 윈도우가 쌓이는 알고리즘은 직선 구간에서는 크게 문제가 되지 않는다. 그러

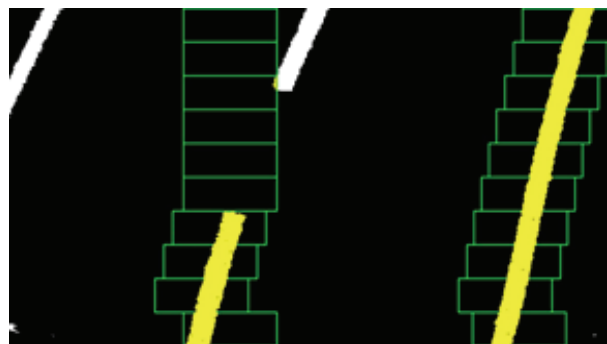


Fig. 2 Example of lane detection failure due to an error in setting the position of the sliding window in the curved dotted section

나 도로는 많은 곡선으로 이루어져 있고, 곡선 구간에서 차선은 y축 시작 부분을 기준으로 왼쪽 또는 오른쪽으로 기울어지게 된다. 곡선 구간에서 접선은 Fig. 2와 같은 차선 검출 오류를 발생시킬 수 있다. 이는 차선이 없는 구간 후에 차선이 다시 생기는 부분에서 슬라이딩 윈도우가 잘못된 위치를 설정하는 문제가 발생할 수 있다. 결국 이는 차선인지 정확도 저하로 이어질 수 있다.

3. 슬라이딩 윈도우 이중 보정

본 논문에서는 슬라이딩 윈도우 이중 보정을 통해 앞서 설명한 문제를 해결하려 한다. 슬라이딩 윈도우 보정 알고리즘은 Fig. 3의 Flow chart와 같은 순서로 진행 된다. 빈 슬라이딩 윈도우에 의해 발생 되는 위치 설정 오류를 차선이 없는 슬라이딩 윈도우의 위치에 따라 해당 프레임의 이전 슬라이딩 윈도우 내지는 이전 프레임의 정상적인 슬라이딩 윈도우 중심점 및 기울기 데이터를 통해 보정하는 2가지 방식으로 접근하였다.

3.1 1차 슬라이딩 윈도우 보정

슬라이딩 윈도우 위치 설정 오류는 빈 슬라이딩 윈도우에 의해 다음 슬라이딩 윈도우가 그대로 쌓이는 현상에 의해 발생한다. 첫 번째 보정에서는 차선 데이터가 있는 이전 슬라이딩 윈도우를 활용하여 빈 슬라이딩 윈도우가 나왔을 때 다음에 쌓이는 슬라이딩 윈도우의 위치를 보정하여 올바른 위치에 설정될 수 있게 한다.

첫 번째 슬라이딩 윈도우 보정 알고리즘은 다음과 같

다. 먼저 Flow chart의 1번 단계와 같이 보정이 필요한 슬라이딩 윈도우 여부를 판단한다. 차선 데이터가 있는 정상 슬라이딩 윈도우라고 판단이 되면 2.1장에서 설명한 Flow chart 2~6번 단계와 같은 일반적인 슬라이딩 윈도우 기법 과정을 진행한다. 아무런 차선 데이터가 없는 슬라이딩 윈도우 뿐만 아니라, 올바른 차선 위치를 판단하기에 너무 소량의 차선 데이터만 들어왔을 경우에도 빈 슬라이딩 윈도우 분류되어, 보정 알고리즘이 적용되어야 한다. 이 때 임의의 Threshold 값을 기준으로 판단하는데, Threshold 값은 본 연구를 통해 반복된 실험으로 산출하였다. 빈 슬라이딩 윈도우라고 판단되면 이전의 유효한 차선 데이터 픽셀의 x 좌표 평균을 각 슬라이딩 윈도우별로 구하고, 그 평균값으로 기울기를 구할 수 있다. 그리고 Flow chart 8번 단계와 같이 구해진 기울기를 통해 같은 위치 x 좌표가 아닌 알맞은 위치의 다음 슬라이딩 윈도우의 기준점이 구해지고, 그 기준점으로 슬라이딩 윈도우를 설정하게 된다. 빈 슬라이딩 윈도우는 차선이 없는 구간만큼 연속해서 나오는데, 차선 데이터가 들어오기 전까지 앞서 구해진 기울기를 기준으로 반복적으로 슬라이딩 윈도우를 쌓아간다. 이 과정을 반복하여 전체 슬라이딩 윈도우를 완성한다.

3.2 2차 슬라이딩 윈도우 보정

첫 번째 보정을 적용하기 위해서는 차선이 있는 2개 이상의 슬라이딩 윈도우가 쌓여 있어야만 한다. 그러나 차선이 없는 부분이 첫 슬라이딩 윈도우부터 위치하는 경우 역시 많이 있는데, 이때는 첫 번째 보정 기술을 적용하는데 한계가 있다. 그리고 첫 번째 슬라이딩 윈도우의 기준은 y축 전체 Histogram을 통해서 구해지는데, 왼쪽이나 오른쪽으로 기울어진 차선에서 아래쪽 차선이 많이 비어져 있는 경우, 치우쳐진 위쪽 차선 데이터만 들어오게 되면 그때 기준이 해당 방향으로 옮겨져서 첫 슬라이딩 윈도우부터 잘못된 위치에 설정이 되는 오류가 발생할 수 있다.

차선인지는 카메라 센서를 통해 들어온 이미지를 기반으로 이루어지고, 본 논문의 실험에서 사용하는 카메라 센서의 경우 30 fps 주기로 이미지가 들어온다. 그리고 그 이미지에서 차선을 검출 하는 알고리즘은 10 fps로 동작을 하는데, 이 말은 연속해서 처리 되는 프레임의 시간 간격이 0.1초로 크지 않고, 각 프레임간의 이미지 차이가 크지 않다는 걸 말해준다. 즉, 이전 프레임에서의 차선위치와 다음 프레임에서의 차선의 위치가 큰 차이가 없음을 알 수 있다. 그래서 본 논문에서는 처음 슬라이딩 윈도우가 쌓이는 부분의 차선 데이터가 부족하다고 판단이 되면, 이전 프레임의 슬라이딩 윈도우 데

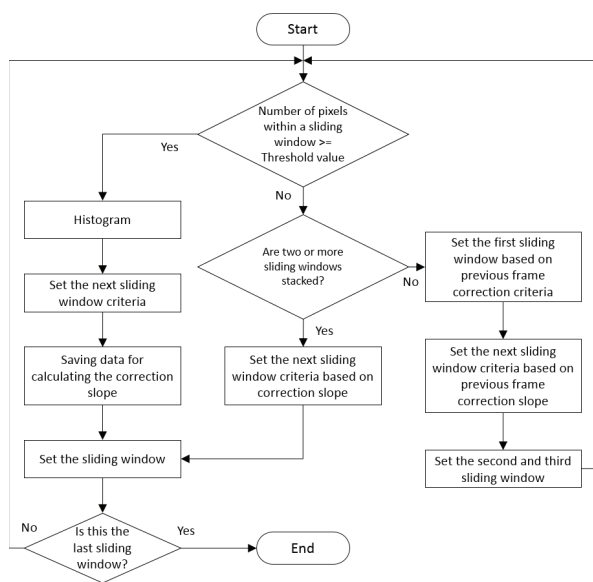


Fig. 3 Sliding window correction algorithm flow chart

이터를 활용하는 두 번째 보정을 통해 앞서 설명한 문제를 해결한다.

먼저 2차 보정의 핵심은 이전 프레임에서 첫 슬라이딩 윈도우의 기준점과 기울기를 가지고 와서 보정에 사용한다는 점이고, 이를 위해 프레임이 넘어갈 때 이 데이터를 항상 넘겨주게 된다. 2차 보정은 Flow chart 2번 단계와 같이 기울기를 구할 때 최소로 필요한 2개의 슬라이딩 윈도우를 기준으로 적용 여부를 판단한다. 첫 번째, 두 번째 슬라이딩 윈도우 중 하나라도 차선 데이터가 없는 슬라이딩 윈도우가 있으면 적용되며, Flow chart 9~11번 단계와 같이 이전 프레임의 데이터 기본 값으로 슬라이딩 윈도우의 위치와 기울기를 재설정한다. 그리고 재설정된 기울기를 바탕으로 차선 데이터가 포함된 슬라이딩 윈도우가 나올 때까지 슬라이딩 윈도우를 쌓는 과정을 반복한다. 이후 슬라이딩 윈도우가 차선을 찾으려 반복문을 초기화하고 해당 차선 데이터를 기준으로 다시 슬라이딩 윈도우 위치를 설정한다.

이 과정을 통해 앞서 설명한 시작부분 차선 데이터 부재로 인해 발생할 수 있는 문제를 해결하여, 올바른 슬라이딩 윈도우가 설정될 수 있게 된다.

4. 실험

4.1 실험 환경

본 논문을 통해 개발한 알고리즘을 실제 군집주행 환경에서만 테스트하기에는 어려움이 있다. 실제 군집주행 환경에서 센서 데이터를 획득하기 위해서는 트레일러가 달린 2대 이상의 대형 트럭을 고속도로에서 직접 주행하며 진행해야 하는데, 실험을 위한 충분한 데이터를 얻기에는 안전, 시간적 측면 등 여러 여건에서 한계가 있다.

그래서 본 연구에서는 다양한 실험을 할 수 있는, 1/14 스케일의 트럭과 트랙을 직접 제작하여, 실험을 진행하였다. 실험은 두 단계로 나뉘어 진행하였다. 스케일 차량을 기반으로 먼저 실험을 진행하였고, 다음으로 트럭 군집주행 실험을 통해 실제 환경에서도 유효함을 한 번 더 확인하였다.

4.1.1 1/14 스케일 트럭

Fig. 4와 같은 1/14 스케일의 트럭을 제작하였다. 트럭은 148 °C wide angle 카메라와 RPLidar A3 등의 센서와 Nvidia Jetson Xavier, OpenCR 보드 등으로 구성하였다. 총 3대를 제작하여 군집주행이 가능하도록 하였다. 실험에서는 군집주행 상황의 차선 데이터를 획득하는데 필요한, 2대의 스케일 트럭만을 사용하여 LV와 FV로 운행하였다.



Fig. 4 1/14 Scale autonomous truck consisting of lidar and camera sensors for platooning experiments

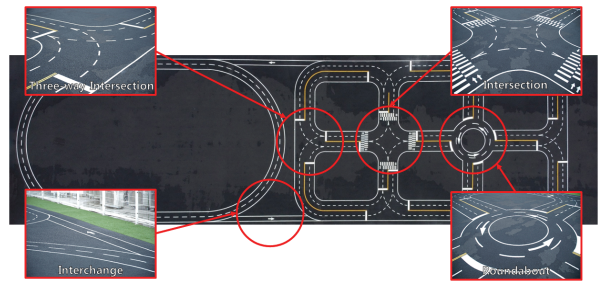


Fig. 5 A track for testing scale cars with a size of 40 m in width and 12 m in height in various environments

Fig. 5는 테스트를 진행한 트랙이다. 전체 크기는 가로 40 m, 세로 12 m이며, 차선의 폭은 33 cm로 제작하였고, 곡률이 큰 고속도로 환경 그리고 다양한 교차로가 있는 시내 도로 환경으로 구성하였다. 본 논문의 실험에서는 Fig. 5의 트랙 중 곡선 반지름 5 m인 왼쪽 고속도로 환경에서 1차 실험을 진행하였고, 2,3차 실험으로 반지름 3,4 m 트랙을 추가 하여 곡률에 따른 성능 차이도 분석하였다.

4.1.2 실제 군집 주행

실험은 실제 군집주행 환경에서도 진행하였다. 실험에 사용한 트럭은 Fig. 6에 나와 있는 2대의 트럭으로, 서론에서 언급한 국내에서 진행 중인 “V2X 기반 화물차 군집주행 운영기술 개발” 프로젝트를 위해 제작된 트럭이다. 2대의 트럭을 스케일 트럭과 같이 각각 LV와 FV로 하여 주행하였고, 21년 시험운행을 진행할 구간인 북여주 IC부터 여주 JC까지 왕복 운행하며 데이터를 획득했다. LV 매뉴얼 주행 그리고 FV는 자율주행으로 진행하였다. 안전한 실험 진행을 위해 0.5 s Time-gap 기준인 12.5 m보다는 좀 더 거리를 둔 16 m를 유지하며 주행하



Fig. 6 Platooning truck in the project “Development of Safety Driving Control Technology for Platooning Truck”

였다. 또한 안전상의 이유로 본 실험에서는 획득한 데이터를 실시간 차선 검출 및 제어에 사용하지 않고, 주행 중에는 데이터 획득만을 진행하였고, 주행 후 획득한 데이터를 통해 테스트를 하는 방식으로 나눠서 진행하였다.

4.2 실험 결과

이번 장에서는 3장에서 제시한 보정 알고리즘을 4.1장에서 설명한 스케일트릭과 실제 군집주행 트럭에 적용한 실험 결과에 대해 서술해본다. 전체 주행 영상 중에서 결과가 잘 비교 될 수 있도록 곡선 주행 영상을 사용하였고, 각각 200장씩 총 800장의 이미지로 실험을 진행하였다. Labelme를 기반으로 직접 만든 GT(Ground Truth) 이미지와 알고리즘 적용 전후의 결과 이미지를 비교하여 정확도 변화를 확인하였다. 정확도 변화는 Fig. 7과 같이 GT 이미지와 결과 이미지를 각각 Bird’s eye view 이미지로 변환 후, 그 이미지를 겹쳐 분포도 차이를 비교하여 구하였다. 분포도 차이를 통해 구해지는 수치는 절대적인 정확도 수치와는 다소 차이가 있을 수

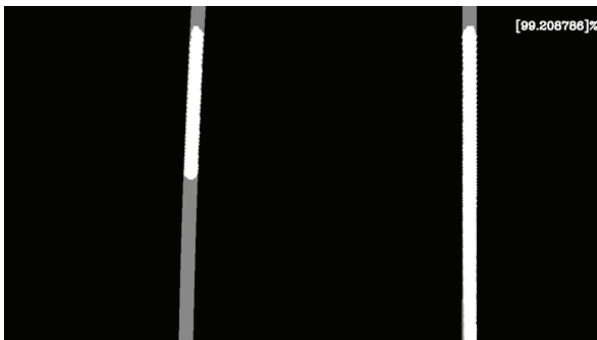


Fig. 7 Example of a comparison experiment between Ground Truth and the distribution of experimental results to verify the accuracy improvement

있지만, 이미지 간 분포도 차이의 변화를 통해 정확한 수치의 정확도 향상 정도를 확인할 수 있다. 실험은 알고리즘 적용 전과 1차 보정 적용 그리고 1,2차 보정 적용으로 총 3단계로 나눠서 진행하였다.

4.2.1 1/14 테스트 베드 결과

Fig. 9는 보정 알고리즘 적용에 의한 슬라이딩 윈도우 위치 재설정으로 알맞은 차선 검출이 이루어지는 결과를 보여주는 예시이다. 먼저 Fig. 9의 (a)~(c)는 첫 번째 보정이 적용되는 단계의 그림으로, Fig. 9(b)는 오른쪽으로 휘어져 있는 점선에서, 차선이 없는 구간 이후 오른쪽에서 나온 차선을 슬라이딩 윈도우가 따라가지 못하였고, 그로 인해 제대로 차선을 검출하지 못하는 걸 보여준다. Fig. 9(c)에는 보정 알고리즘이 적용된 것으로, 차선이 없는 구간의 슬라이딩 윈도우의 위치가 보정되어 전체 차선이 제대로 검출되는 것을 보여준다.

Fig. 9의 (d)~(f)는 두 번째 보정이 적용되는 단계를 보여주는데, Fig. 9(e)에서 세 번째 슬라이딩 윈도우 위치까지 차선 데이터가 없고 이로 인해 첫 슬라이딩 윈도우의 기준점부터 잘못된 위치에 설정되어 전체 차선 검출에 문제가 발생한 것을 보여준다. Fig. 9(f)는 이전 프레임의 슬라이딩 윈도우 기준점과 기울기를 값을 바탕으로 진행하는 두 번째 보정 알고리즘을 통해 첫 슬라이딩 윈도우 중심점 설정부터 보정된 것을 보여준다. 이로 인해 전체 차선 검출이 올바르게 진행되었고, 이는 정확도 향상으로 이어졌다.

스케일 트릭 환경의 실험은 Table 1과 같은 결과를 보

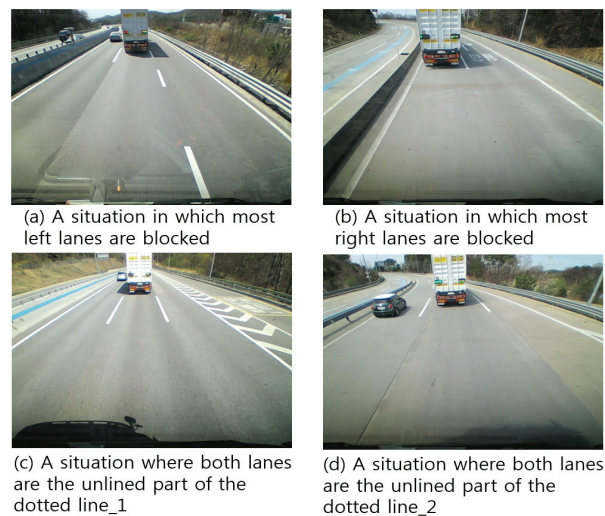


Fig. 8 Situations where it is difficult to detect lanes due to lane blocking of a truck in front caused by a narrow inter-vehicle distance on a highway

였다. 먼저 반지름 5 m 트랙에서의 1차 실험 결과에서는 알고리즘 적용 전 실험 결과와 GT와의 분포도 차이가 200장 평균 90.24 %로 나왔고, 첫 번째 보정 알고리즘만 적용했을 때의 분포도 차이는 94.94 %로 나왔다. 그리고 두 번째 보정 알고리즘까지 모두 적용했을 때는 97.75 %가 나왔다. 반지름 4 m에서 진행한 2차 실험에서는 알고리즘 적용 전이 87.28 %, 첫 번째 보정 알고리즘 적용 후 92.96 % 그리고 두 번째 보정 알고리즘 까지 적용했을 때 94.28 %의 분포도 차이를 보였다. 마지막으로 반지름 3 m에서 진행한 3차 실험에서는 알고리즘 적용 전 73.44 %, 첫 번째 보정 알고리즘 적용 후 78.96 % 그리고 두 번째 보정 알고리즘 까지 적용했을 때 84.08 %의 결과를 보였다.

이로 인해 스케일 트랙 환경의 실험에서는 알고리즘 적용으로 평균 약 8 %의 정확도가 향상되는 것을 확인하였다. 그리고 곡률 변화에 따른 정확도 비교를 통해

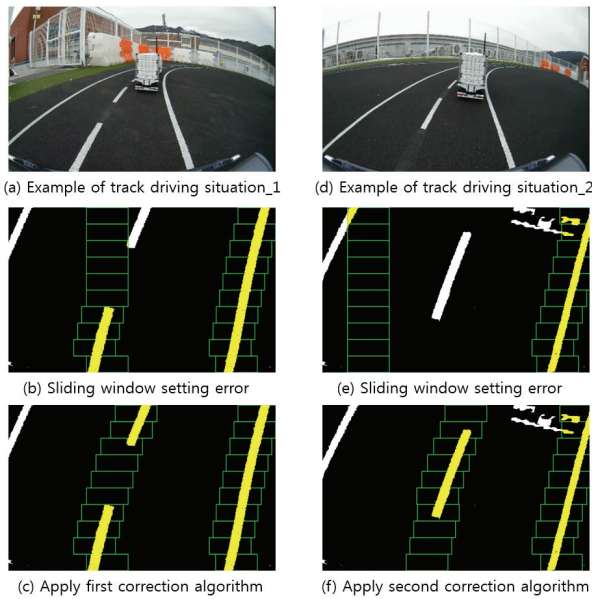


Fig. 9 Examples of proper lane detection through correction of sliding window position setting in the 1/14 scale truck experiment

Table 1 Accuracy comparison according to curvature change

Radius size (m)	5	4	3
Before applying the algorithm (%)	90.24	87.28	73.44
Apply the first correction algorithm (%)	94.94	92.96	78.96
Apply the second correction algorithm (%)	97.75	94.28	84.08

곡률이 작은 1차 실험에 비하여 곡률이 큰 3차 실험으로 갈수록 평균 정확도가 감소한 것을 확인하였고, 그리고 곡률이 커질수록 평균 정확도는 떨어지지만 알고리즘 적용을 통한 정확도 향상 폭은 더 크게 나오는 것을 확인하였다.

4.2.2 실차 데이터 결과

스케일 트랙 실험의 경우 실험을 진행한 트랙의 점선 간격이 고속도로에 비해 좁아 알고리즘 적용 전후 정확도가 대체로 모두 높게 나왔고, 그로 인해 정확도 향상 폭이 크지 않았다. 그러나 고속도로의 경우 Fig. 8과 같이 점선 간격이 확실히 넓어 점선에 의한 차선 검출 정확도가 저하가 분명하였고, 이로 인해 알고리즘 적용 전후의 차이가 좀 더 명확히 나오는 것을 확인하였다.

Fig. 10은 스케일 트랙 실험 결과 예시와 같은 실차 검증에서의 실험 결과이다. Fig. 10의 (a)~(c)는 첫 번째 보정 알고리즘의 적용으로 슬라이딩 윈도우의 위치가 보정되어 차선이 제대로 검출 되는 것을 보여주고, Fig. 10의 (d)~(f)는 첫 슬라이딩 윈도우부터 발생한 위치 설정 오류를 두 번째 보정 알고리즘까지 적용하여, 제대로 된 위치로 수정되는 결과를 보여준다.

실제 군집 주행 실험에서는 알고리즘 적용 전 GT와 실험결과의 분포도 차이는 평균 86.81 %였고, 첫 번째 보정 알고리즘 적용 후는 94.7 % 그리고 두 번째 보정 알고리즘까지 모두 적용했을 때는 97.86 %가 나왔다. 알고리즘 적용을 통해 약 11 %의 정확도 향상을 확인하

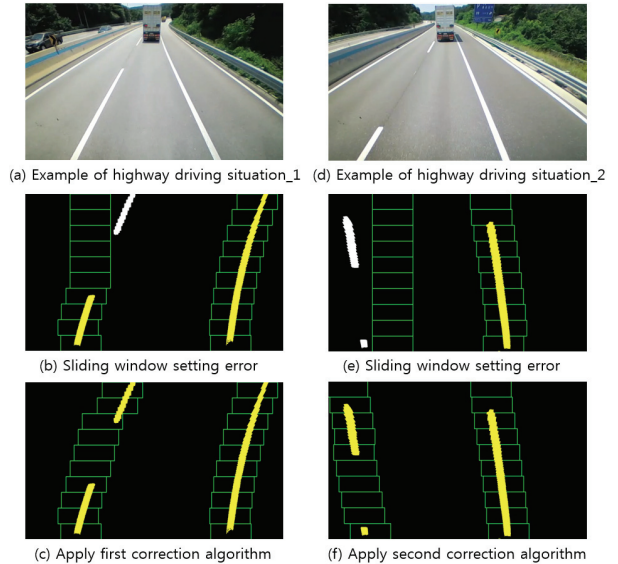


Fig. 10 Examples of proper lane detection through correction of sliding window position setting in real-world platooning experiments

였다. 실차 실험을 통해 스케일 트럭 환경뿐만 아니라 실제 군집주행 환경에서도 슬라이딩 윈도우 보정 알고리즘을 사용하여 차선인지 정확도를 향상시킬 수 있다는 것을 확인하였다.

5. 결론

본 논문은 트럭 군집주행의 특징에 의해 곡선 구간에서 발생할 수 있는 차선인지의 문제점을 해결하여 차선인지 정확도를 향상시키는 방법에 관한 연구를 진행했다. 곡선 구간에서 발생할 수 있는 슬라이딩 윈도우 위치 설정 문제를 이중 보정 알고리즘을 통해 해결하였고, 그 결과 알고리즘 적용 전에 비하여 약 10%의 정확도 향상을 확인하였다.

이후 연구에서는 본 논문에서 제시한 알고리즘을 포함한 군집주행 통합 차선인지 모듈을 개발하고, 실제 군집주행 트럭에 적용하기 위한 최적화 방법을 연구한다.

후 기

본 연구는 부분적으로 국토교통부의 ‘교통물류연구사업’으로 지원을 받아 수행된 연구결과임(21TLRP-B147674-04, V2X 기반 화물차 군집주행 운영기술 개발). 또한, 부분적으로 2021년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 BK21사업임(5199990814084).

References

- 1) V. Turri, B. Besselink and K. H. Johansson, “Cooperative Look-Ahead Control for Fuel-efficient and Safe Heavy-Duty Vehicle Platooning,” *IEEE Transactions on Control Systems Technology*, Vol.25, No.1, pp.12-28, 2016.
- 2) G. R. Janssen, J. Zwijnenberg, I. J. Blankers and J. S. Kruijff, “Truck Platooning: Driving the Future of Transportation,” *TNO Mobility and Logistics*, 2015.
- 3) C. Bergenheim, S. Shladover, E. Coelingh, C. Englund and S. Tsugawa, “Overview of Platooning Systems,” *Proceedings of the 19th ITS World Congress*, 2012.
- 4) S. Tsugawa, S. Kato and K. Aoki, “An automated Truck Platoon for Energy Saving,” *International Conference on Intelligent Robots and Systems*, pp.4109-4114, 2011.
- 5) Y. Lee, T. Ahn, C. Lee, S. Kim and K. Park, “A Novel Path Planning Algorithm for Truck Platooning Using V2V Communication,” *Multidisciplinary Digital Publishing Institute, Sensors*, Vol.20, No.24,

Paper No.7022, 2020.

- 6) Y. Lee, S. Kim, T. Ahn, M. Sho and K. Park “Development of Inter-vehicular Distance Control for Platooning of Heavy-duty Vehicle,” *KSAE Spring Conference Proceedings*, p.876, 2018.
- 7) E. van Nunen, F. Esposto, A. K. Saberi and J. P. Paardekooper, “Evaluation of Safety Indicators for Truck Platooning,” *IEEE Intelligent Vehicles Symposium (IV)*, Vol.11, No.14, pp.1013-1018, 2017.
- 8) T. W. Kim, W. S. Jang, J. Jang and J. C. Kim, “Camera and Radar-based Perception System for Truck Platooning,” *International Conference on Control, Automation and Systems (ICCAS)*, pp.950-955, 2020.
- 9) B. Kwon, H. So, Y. Lee, D. Lee, M. Soh and K. Park, “A Study on Integrated Control of Preview ARS/RWS for Improvement of Lateral Stability based on Front Road Profile,” *KSAE Spring Conference Proceedings*, p.452, 2017.
- 10) T. Kucza, M. Nättinen and P. Parviainen, “Improving Knowledge Management in Software Reuse Process,” *International Conference on Product Focused Software Process Improvement*, pp.141-152, 2001.
- 11) E. S. Jang, J. K. Suhr and H. G. Jung, “Lane-end-point Detection and Automated Performance Evaluation for Vehicle Localization on the Highway,” *Transactions of KSAE*, Vol.27, No.1, pp.47-54, 2019.
- 12) T. W. Kim and J. C. Kim, “Enhanced Lane Detection Method for Truck Platooning,” *Communications of the Korean Institute of Information Scientists and Engineers*, Vol.37, No.9, 2020.
- 13) M. Nieto, L. Salgado, F. Jaureguizar and J. Cabrea, “Stabilization of Inverse Perspective Mapping Images based on Robust Vanishing Point Estimation,” *IEEE Intelligent Vehicles Symposium*, pp.315-320, 2007.
- 14) H. Kim, M. Park, W. Son, H. Choi and S. Park, “Deep Learning based Object Detection and Distance Estimation using Mono Camera,” *Journal of Korean Institute of Intelligent Systems*, Vol.28, No.3, pp.201-209, 2018.
- 15) C. Ma, H. Zhao and X. Wang, “Wheat Rows Detection Based on Machine Vision,” *International Journal of Signal Processing, Image Processing and Pattern Recognition*, Vol.9, No.9, pp.1-10, 2016.
- 16) J. He, S. Sun, D. Zhang, G. Wang and C. Zhang, “Lane Detection for Track-following Based on Histogram Statistics,” *IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC)*, pp.1-2, 2019.