

GPS의 위치 정보 시스템을 활용한 자율주행 차량의 차선 검출 구현

박 현 희*

Implementation of Lane Detection for Self-Driving Vehicles Using GPS

Hyunhee Park*

요 약

최근 영상처리 분야에서 동영상으로부터 움직이는 객체를 검출하고 추적하는 시스템에 대한 관심이 점차 높아지고 있다. 이러한 연구는 보안 분야, 관측 시스템, 군사적 분야 등 다양한 분야에서 활용되고 있으며 보다 높은 정확도와 고속처리를 위한 많은 연구가 진행되고 있다. 특히, 자율 주행에 관한 관심이 급격하게 커지면서 차선 유지 보조장치에 대한 다양한 알고리즘들이 개발되고 있다. 본 논문에서는 색 공간을 이용한 차선검출 알고리즘과 대표선을 이용한 차선검출 알고리즘의 결과들을 비교하여 차선을 인식하는 알고리즘을 제안한다. 특히, 차선 검출의 정확성을 위하여 엣지 추출과 가우시안 필터를 적용하고, 이미지에 대한 노이즈를 줄이기 위하여 Median을 적용한다. 또한 4개의 포인터를 기준으로 이미지에 대한 관심 영역을 추출함으로써 검출에 대한 정확도를 높였다. 마지막으로 허프 변환을 적용함으로써 직선 검출에 대한 정확성을 높이고자 하였으며, 그림자 영역과 어두운 곳에서의 검출률을 높이고자 대표선을 추출하는 알고리즘을 적용하였다. 특히, 차량 내 GPS를 이용하여 차량의 위치와 속도를 파악하여 저속의 시내 주행과 고속 주행의 결과를 인식하여 적응적으로 차선 검출 알고리즘을 적용하도록 한다. 본 알고리즘을 차량용 블랙박스 또는 자율주행에 이용함으로써 차선 이탈을 방지하고 사고 발생률을 감소시키는데 도움이 될 것이라 생각한다.

Key Words : Lane Detection, Self-Driving Vehicle, Machine Learning, Tensor Flow, OpenCV

ABSTRACT

As interest in Self-driving has increased rapidly, various algorithms for lane detection assistances have been developed. This paper proposes a lane detection algorithm by comparing color-based lane detection algorithms and representative line extraction. Edge extraction and Gaussian filters are applied for the lane detection and a median filter is applied for image noise reduction. The detection accuracy is improved by extracting the region of interest for the image based on four pointers. Finally, a Hough transform and representative lane detection are applied to improve the accuracy of straight line detection. In particular, GPS can be assisted to recognize the position and speed of the self-driving vehicles and adaptively applied to the lane detection algorithm. Experimental results show that the proposed algorithm can detect lanes with high accuracy. The application of this algorithm to vehicle black boxes or autonomous driving will help prevent lane departure and reduce accident rates.

* 본 연구는 한국연구재단 연구과제 (2017R1C1B5017556) 지원으로 수행되었습니다.

* 본 논문의 시뮬레이션 결과는 김환 (서울상계고등학교, rlaghks1103@gmail.com) 학생과 이용희 선생님의 참여로 진행되었음.

• First Author : (ORCID:0000-0003-3810-7367) Korean Bible University, Department of Computer Software, parkhyunhee@gamil.com, 종신회원

논문번호 : 201806-D-121-RN, Received April 26, 2018; Revised June 14, 2018; Accepted June 15, 2018

I. 서 론

IT 기술이 융합된 지능형 자동차에 대한 관심이 높아지면서 이에 대한 연구가 활발하게 진행되고 있다^[1,2]. 특히 사고의 위험을 줄여주는 차선이탈경보, 차선 유지, 보행자 충돌경보, 정방 충돌 경보 등과 같은 첨단안전자동차 (ASV: Advanced Safety Vehicle) 기술은 지능형 자동차 기술의 기초가 되는 기술로서 많은 연구와 기술개발이 진행되고 있다^[3,4]. 차선 검출 기술 개발의 경우, ASV 기술의 핵심 기술로 영상센서를 이용한 비전 기반 방법이 가장 널리 활용되고 있다. 비전 기반 차선 검출은 크게 특징 추출, 이상치 제거 및 전처리, 후처리, 차선 추적의 네 단계로 이루어진다. 각 단계에서 사용되는 알고리즘에 따라 차선 검출 알고리즘의 성능 및 특성이 결정된다^[5].

차선 검출 기술에서 사용하는 특징 추출 기법은 크게 경계선과 칼라 부분으로 나눌 수 있다. 기존의 특징 추출들은 조명 변화에 따라서 특징의 검출 여부가 크게 좌우되는 문제가 있었다. 따라서 이에 대한 해결 방안으로 고해상도 데이터 획득이 가능한 센서를 사용하거나 탐지 거리가 먼 이중의 센서를 결합하는 방법 등이 있으나, 이 또한 비용 증가와 실시간 처리의 어려움과 같은 문제가 있다^[6]. 본 논문에서는 환경에 취약한 기존 특징 검출 방법의 문제를 해결하기 위해 환경에 적응적으로 경계선을 검출하는 방법을 제안한다. 또한 본 논문에서는 다양한 차선 형태 및 도로 환경에 효율적으로 대응하고, 실용화를 목적으로 차선 검출의 성능을 개선함과 동시에 계산량을 감소시켜 실시간 처리를 가능하게 하는 새로운 차선 검출 방법을 제안한다. 본 논문의 구성은 기존의 영상 기반 차선 검출 방법의 문제점에 대하여 2장에서 언급하고, 3장에서 본 논문에서 제안하는 6단계의 차선 검출기법에 대하여 기술한다. 4장에서는 Gray 색변환, HSV 색변환, 대표선 추출 기법을 적용한 차선 검출의 실험 결과를 보여준다. 5장에서 분석 결과를 통한 결론을 맺는다.

II. 기존의 영상기반 차선 검출 기법

차선 검출 기술에서 사용하는 특징은 크게 경계선과 칼라 부분으로 나눌 수 있다. 컬러 기반 방법들은 조명 변화에 매우 취약한 성능을 보이기 때문에 아직 초기 연구에 머무르고 있다^[7]. 이러한 이유로 경계선 검출 기반 방법들이 널리 연구되고 있는 추세이다. 일반적으로 칼라 영상을 흑백영상으로 변환하여 경계선

검출을 수행한다^[8]. 이는 칼라 영상에 적용되는 경계선 검출 기법 알고리즘의 복잡도 증가에 비해 성능 향상이 적기 때문이다. 또한 이상치 제거 및 후처리 과정에서 부적절한 경계선들을 제거하고 검출되지 않는 영역에 대한 보상을 수행하는데 중점을 둔다^[9]. 그러나 경계선 검출이 잘 일어나지 않는 경우에는 전체 시스템의 성능저하가 크게 나타날 수 밖에 없다.

다른 차량의 위치와 장애물에 대한 정보를 제공하여 장애물 회피 시스템에서 차선 검출 기법이 이용될 수 있다. 이러한 차선 이탈 시 차선 감지 기술의 경우 다양한 방법이 연구되고 있다. 차량에 장착된 다수의 Rader를 통해 전방의 차선을 감지하여 인식하는 방법이 있다^[9]. 이러한 방법의 경우 주로 고비용의 센서가 장착되어 상용화 되는 단점을 가진다. 또한 블랙박스 또는 차량 전방의 카메라를 통해 차선의 경계를 인식하는 방법이 있다^[10]. 본 논문에서는 고비용의 센서를 장착하지 않아도 차량 전방을 주시할 수 있는 영상장치를 통해 바로 차선을 검출할 수 있는 방법을 제안한다. 또한 대부분의 차량 혹은 휴대폰에 내장되어 있는 GPS를 통해 이동 중인 차량의 위치와 속도를 인식하여 차선 검출 방법을 적응적으로 변화시키는 방법을 제안한다.

본 논문에서는 차선 검출의 고속화와 검출의 정확성을 위하여 6단계의 과정을 거쳐 차선 검출을 수행한다. 특히 제안하는 차선 검출 알고리즘은 GPS 기반의 위치 인식을 통하여 자율주행 차량이 고속도로에서의 고속 주행을 하는 경우 HSV 색변환 기법을 적용하여 차선 검출을 수행한다. 만일 위치 인식을 통해 자율주행이 시내 주행과 같은 저속 주행을 하게 되는 경우 주변 차량에서 발생하는 그림자 영역에 영향을 덜 받는 Gray 색 변환 기법을 사용한 차선 검출을 수행한다. 따라서 본 논문에서는, GPS를 통해 확인되는 차량의 속도와 위치를 연계하여 색 변환의 과정에서 Gray 색 변환 기법을 사용한 경우와 HSV 색 변환 기법을 사용한 경우에 대해 차선 검출의 정확성을 비교한다. 일반적인 차선의 경우 차선 위의 장애물이나 차선의 노화로 인해 선 검출이 명확하지 않은 경우가 발생하는 것을 알 수 있다. 이를 위해 차선의 위치를 예측할 수 있는 대표선 검출 알고리즘을 추가적으로 적용함으로써 차선 검출이 직선 영역과 곡선 영역에서 정확히 검출될 수 있도록 제안한다.

III. 차선 검출 기법

이미지를 출력하기 위하여 Open CV 라이브러리의

videoCapture를 사용하여 동영상을 불러온다. 동영상은 주간에 달리는 차량의 블랙박스 영상을 통해 추출하였다^[12,13]. 본 논문에서는 기본적으로 출력한 이미지의 RGB 값을 변경하기 위해 Gray color code나 HSV (Hue, Saturation, Value) color code를 고려한다. 또한 추가적으로 대표선을 추출함으로써 차선을 검출한 결과를 함께 비교한다. 이번 장에서는 제안하는 차선 검출 기법을 순서대로 기술하고 각각의 기법이 적용된 실험 결과들을 함께 제시한다.

3.1 GPS를 통한 위치 인식 기반의 color code 설정

기본적으로 차량은 시내 주행과 고속 주행을 하게 된다. 또한 차량은 곡선 구간과 직선 구간을 통과하게 된다. 이러한 차량의 동작은 차량 내 연결되어 있는 GPS의 위치 인식을 통하여 정확하게 판단할 수 있다^[14]. 기본적으로 현재 개발된 차량은 GPS가 장착되어 있고, 만일 GPS가 장착되어 있지 않은 차량이라 하더라도 휴대폰을 이용한 내비게이션 기능을 활성화하게 되면 차량을 위한 GPS가 활성화 된다. 이러한 GPS 기능이 차량에서 기본적으로 제공된다는 가정하여 본 논문은 위치 기반 및 속도 인식을 고려한다. 예를 들어, 고속 주행을 하게 되는 차량은 차량 내 구현되어 있는 GPS를 통해 일정 속도를 측정하고 주행 차량의 위치를 판단하여 고속도로를 주행하는 것을 판단할 수 있다. 또한 시내 주행을 하는 차량은 속도 감지와 GPS의 위치 파악으로 인하여 시내 주행을 한다는 것을 파악할 수 있다. 본 논문에서 차량 내 GPS를 이용하여 차량의 위치 파악을 하는 이유는 차량 속도에 따른 색 변환 기법을 적응적으로 적용하기 위해서이다. 저속의 시내 주행을 할 경우 주변 차량의 움직임과 많은 차선 변경으로 인해 진행 방향에 옆 차량으로 인한 그림자가 많이 발생하게 된다. 옆 차선에서 발생하는 그림자는 차선을 인식하는데 방해요소가 될 수 있다. 따라서 저속 주행을 그림자의 영향을 덜 받는 color code의 설정이 필요하다. 반대로 고속 주행의 경우 옆 차량에 의한 그림자가 발생하더라도 주행 방향의 속도가 빠르기 때문에 그림자를 인식하지 않고 생략한 결과로 차선을 인식하게 되더라도 문제가 되지 않음을 알 수 있었다. 이 또한 고속 주행에 대하여 적당한 color code의 설정이 필요함을 알 수 있다.

3.2 차선 검출을 위한 색 변환 기법

본 논문에서는 차선 검출을 위하여 먼저 색 변환 기법을 적용한다^[15,16]. 색 변환을 위한 기법은 Gray

color code와 HSV color code에 대한 두 가지 기법을 적용한다.

먼저 색 변환을 적용하기 위해 불러 온 차량 주행 영상을 OpenCV의 컬러 필터링을 통해 변화시킨다. OpenCV에는 150가지 이상의 색 공간 변환 방법이 있는데 본 논문에서는 영상을 실시간으로 처리하기 위하여 cv2.cvtColor을 통해 색상을 Gray로 변화시킨다. 만일 OpenCV에 적용하는 컬러 필터링의 색 공간 변환에서 Gray = TRUE라면 Gray color code를 통한 색 변환을 수행하고 난 후 이진화 작업을 수행한다. 이진화 작업은 흑백 분류를 명확히 하기 위한 작업이다. Gray color code를 사용하게 되면 노면에 생기는 그림자를 함께 추출해 내게 된다. 그림 1은 Gray color code를 적용한 직선 차선에서의 검출의 결과이고, 그림 2는 Gray color code를 적용한 곡선 차선에서의 검출의 결과이다. 그림 3을 통해 Gray color code를 적용함으로써 옆 차량에 의해 발생하는 그림자도 함께 차선 검출 영역에서 추출되는 것을 알 수 있다.

만일 OpenCV에 적용하는 컬러 필터링의 색 공간 변환에서 Gray = FALSE라면 HSV color code로 색

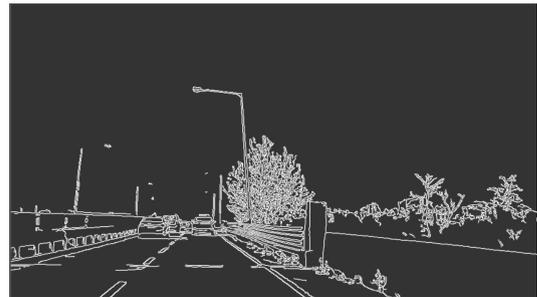


그림 1. 직선 구간에서의 에지 추출 결과
Fig. 1. Edge detection result in straight line



그림 2. 곡선 구간에서의 에지 추출 결과
Fig. 2. Edge detection result in curve line

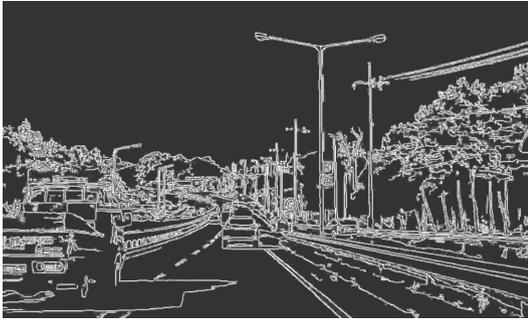


그림 3. 그림자 영역에 대한 에지 추출 결과
Fig. 3. Edge detection result in shadow area

추적을 수행한다. 이를 위해 컬러 마스크를 이용하여 흰색만을 검출한다. 차선의 특성 상 도로에는 여러 차량들이 지나갈 수 있고, 그 차량에 의한 그림자도 발생할 수 있다. 또한 노면의 노후로 인하여 지워진 차선이 발생할 수도 있다. 따라서 차선의 특성을 색을 통해 종류를 구분할 수 있다. 본 논문에서는 HSV color code의 색 공간을 이용하여 에지를 검출하였고, 이를 Canny 에지 알고리즘과 함께 적용함으로써 정확성을 높였다⁸⁾. 이를 위하여 RGB 형태로 입력 받은 이미지를 HSV color code 색 공간으로 변환한다. HSV color code 색 공간을 이용하여 색 공간 영역을 설정함으로써 흰색, 노란색, 파란색을 검출하게 된다. 이 세 가지 색상은 이진법을 적용하여 흰색으로 적용되고 나머지 색들에는 검은색으로 적용된다.

3.3 에지 검출 기법

영상을 통해 추출된 이미지에서 차선을 검출하는 가장 기본적인 방법 중 하나는 에지 검출을 수행하는 것이다. 왜냐하면 도로 노면에서 차선은 다른 부분과 경계를 이루는 것이 드러나기 때문이다. 특히 흰색으로 구분되는 차선의 경우 검정색이 기본인 도로와 확연히 구분될 수 있다. 본 논문에서는 차선의 에지 검출을 위해 Canny 에지 알고리즘을 사용한다. Canny 에지 알고리즘은 휴리스틱한 방식으로 접근하는 기존의 Roberts나 Prewitt 에지 검출 방법과 달리 최적화 관점에서 에지를 검출하는 방법으로 다음과 같이 3단계의 알고리즘으로 구성될 수 있다⁸⁾.

- 1) 화소 기울기의 강도와 방향 검출
- 2) 비 최대치 억제
- 3) 이력 임계값을 적용하여 거짓 긍정된 값을 제거

Canny 에지 알고리즘은 윤곽을 잘 찾아내면서도

영상의 회색 물질과 관련된 에지를 제거할 수 있다. 본 논문에서는 5*5의 가우시안 필터를 사용하여 노이즈를 제거하고, 이미지의 gradient 방향과 강도를 확인한다. 그리고 경계값을 결정하기 위하여 미분값을 사용하게 되는데, 주변과 색이 달라서 미분값이 급속도로 변경될 때의 값을 고려하게 된다. 그림 1과 그림 2를 보면, 직선 구간과 곡선 구간 모두 이미지의 픽셀을 전체적으로 스캔하여 에지가 아닌 픽셀을 제거함으로써 에지를 명확하게 판별하는 것을 알 수 있다.

3.4 가우시안 필터의 적용

가우시안 필터는 영상이 가지게 되는 노이즈를 제거해주는 필터이다¹⁷⁾. 가우시안 필터의 원리는 입력 배열의 각 포인트를 가우스 커널로 컨벌루션 한 후 모두 합하여 출력 배열을 만들면서 필터 작업을 진행하는 것이다. 본 논문에서는 영상을 스무딩하게 만들기 위하여 Median 필터를 추가적으로 적용한다. 가우시안 필터와 Median 필터를 순서대로 적용함으로써 이미지의 노이즈를 제거하고 영상을 스무딩하게 만들어 줄 수 있다.

그림 4는 가우시안 필터를 적용한 결과이고 그 결과에 추가적으로 Median 값을 취한 결과를 그림 5를

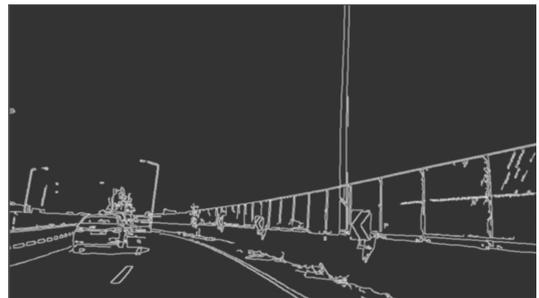


그림 4. 가우시안 필터를 적용한 결과
Fig. 4. Result of applying Gaussian filter

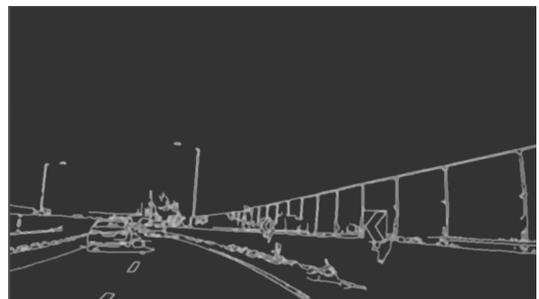


그림 5. Median 필터를 적용한 결과
Fig. 5. Result of applying Median filter

통해 볼 수 있다. 가우시안 필터를 적용한 후 Median 필터를 한번 더 적용함으로써 이미지에 블러를 입힌 것과 같이 이미지가 스무딩 하게 되는 것을 알 수 있다. 이진화 작업을 수행하면서 생긴 노이즈가 제거된 것이라 볼 수 있다.

3.5 관심영역 추출 기법

본 논문에서는 차선 검출을 위하여 관심 영역 추출 기법 (Region of Interest for Lane Boundary: ROI-LB)을 적용한다. 관심 영역을 추출하는 이유는 첫 번째로 이미지의 관심 영역만 추출하게 되면 이미지의 픽셀수가 줄어들어 연산량이 줄어들기 때문이다^[18,19]. 두 번째는 이후 허프 변환을 적용할 때도 관심 영역 내에서만 작동하기 때문에 선들이 관심 영역 내에서만 연산이 되어 다른 배경 (예를 들어, 산, 구름 등)이 인식되지 않기 때문에 정확성이 더 높아진다. 관심 영역 추출을 위하여 mask를 통해 이미지를 블러운 후 이미지의 배열을 0으로 되돌린다. 그리고 roi_corners에서 다각형의 array를 통해 4개의 점을 만든다. filloply에서 픽셀 값이 0으로 되돌린 mask와 다각형의 array를 합쳐서 하나 이상의 다각형으로 묶인 영역을 채운다. 그리고 bitwise_and 연산자를 통해 둘 다 0이 아닌 경우만 값을 통과시킴으로써 mask 이외의 영역을 모두 제거하게 된다. 즉 검정색이 아닌 경우만 통과되기 때문에 mask영역 이외는 모두 제거가 된다. ROI-LB를 사용함으로써 다양한 차선의 패턴이나 직선 또는 곡선의 도로 형상에 유연하게 대응할 수 있으며 차선이 검출영역을 벗어나는 경우를 최소화할 수 있다^[20,21]. 또한 실제 도로에서는 차선 이외에 방향 안내 표시나 제한 속도 표기 등의 다양한 주행 안내 정보, 횡단보도, 사거리 등과 같이 차선 이외에 다양한 마킹이 존재한다. 기존의 알고리즘에 의한 차선 검출은 이들의 주행 안내 정보가 차선으로 오인될 가능성이 크기 때문에 ROI-LB 기법을 통하여 차선 검출 성능을 높일 수 있게 된다. 본 논문에서는 관심영역의 설정을 위하여 4개의 특징점(key point)을 설정하였다. 영상에서 특징점이 되기 위한 조건으로는 물체의 형태나 크기, 위치가 변해도 쉽게 식별이 가능해야 하고, 카메라의 시점, 조명이 변해도 영상에서 해당 지점을 쉽게 찾아낼 수 있어야 하는 점이다. 영상에서 이러한 조건을 만족하는 가장 좋은 특징점 중에 하나는 코너점(corner point)이다. 대부분의 특징점 추출 알고리즘은 코너점 검출을 바탕으로 하고 있다. 영상에서 코너점을 찾기 위해 영상의 각 픽셀 위치에 대해 원도우를 수직, 수평, 좌대각선, 우대각선 이렇게

4개 방향으로 1 픽셀씩 이동시켰을 때의 영상 변화량을 계산하여 이를 해당 픽셀의 영상 변화량의 값으로 설정한 후 그 값이 지역적으로 극대가 되는 지점을 코너점으로 결정한다.

그림 6을 보면 4개의 점을 기준으로 관심 영역을 위한 선을 설정하는 것을 볼 수 있다. 그리고 관심 영역을 컬러 이미지로 추출한 후 코너점 계산을 통해 그림 7과 같은 코너점 추출 결과를 얻을 수 있다. 최종적으로 관심 영역에서 에지만 추출하게 되면 그림 8과 같은 결과를 얻을 수 있다. 결국 관심 영역을 추출



그림 6. 관심영역 추출을 위한 원본 이미지
Fig. 6. Original image for extracting ROI

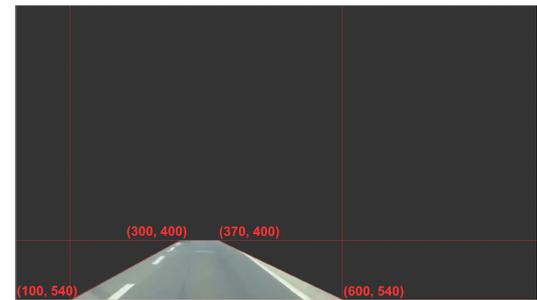


그림 7. 코너점이 적용된 컬러 이미지
Fig. 7. Color image with Corner points



그림 8. 에지 검출이 완성된 이미지
Fig. 8. Edge-detected image

하고 컬러 이미지에서 에지 정보만 추출하게 되면 차선으로 오인식되는 정보들을 제거하게 됨으로써 획기적으로 차선 검출의 성능 개선이 가능하게 되는 결과를 얻을 수 있다.

그림 7을 보면 본 논문에서 관심영역 추출을 위하여 사용한 4개의 코너점 값이 적용된 것을 확인할 수 있다. (300, 400), (370, 400), (100, 540), (600, 540)의 4개의 코너점을 이미지에 적용하여 그 점을 기준으로 생성되는 다각형의 나머지 부분은 mask 처리를 수행한다. 이렇게 함으로써 추출하고자 하는 차선에 대해 주행 방향을 고려한 차선 검출 영역을 분류하게 된다. 이렇게 차선이 있는 영역만을 관심 영역으로 추출한 후 차선을 검출하게 됨으로써 앞서 언급한 다른 영역의 연산 처리를 수행하지 않게 된다. 따라서 연산 속도가 빨라지게 되고, 다른 영역의 검출로 인하여 방해받는 요소가 적어짐으로써 전반적으로 차선 검출의 정확도가 높아지는 것을 알 수 있다.

3.6 허프 변환 기법

차선을 찾기 위해서는 우선 직선 검출을 수행해야 한다. 즉, Canny 에지 알고리즘에서 추출한 에지에서 직선을 추출하기 위하여 허프 변환을 이용하게 된다^[22]. 허프 변환은 직선의 방정식을 사용한다. 직선의 방정식으로 표현하면 $r = x \cos \theta + y \sin \theta$ 표현이 가능하다. 그리고 이것을 삼각함수로 이용하여 변형하게 되면 $y = mx + b$ 표현 할 수 있다. minLineLength로 최소 길이를 정하고 이보다 작은 값은 고려하지 않는다. maxLineGap은 선과 선 사이의 최대 허용 간격이다. 허프 변환의 경우 직교 좌표계의 표현을 극 좌표계 형식으로 나타내 준다. 극 좌표계는 원점에서 떨어진 수직 거리와 이 수직선이 x축과 이루는 각도 θ 로 직선을 표현할 수 있고, 일정한 간격으로 검출함으로써 정밀하게 검출된다. 본 논문에서는 허프 변환을 위해 아래 6단계를 거쳐 허프 변환을 수행한다.

- 1) 허프 변환 좌표계에서 행렬 구성
- 2) 영상 내 모든 화소의 직선 여부 검사
- 3) 직선인지 좌표의 허프 변환 누적 행렬 구성
- 4) 허프 누적 행렬의 지역 최대값 선정
- 5) 임계값 이상인 누적값 선별
- 6) 직선을 누적값 기반으로 내림차순 정렬

3.7 대표선 추출 기법

차선 인식을 하는 경우 직선의 구간뿐만 아니라 곡선 구간에서의 적용도 중요하다^[23,24]. 이를 위하여 직선의 기울기를 고려해야 한다. 허프 변환을 통하여 모

든 직선들의 시작점과 끝점에 대한 좌표값을 알고 있으므로 각 직선들의 기울기를 구할 수 있다. 또한 수평 기울기에 대하여 본 논문에서는 160도로 제한을 하였고, 수직 기울기에 대해서는 90도로 제한을 하였다 (이는 휴리스틱한 결과 값으로 다양한 값을 적용한 후 결정한 각도이다). 이후 필터링 된 직선을 제거하고 왼쪽과 오른쪽의 각각의 대표선을 구하도록 한다.

위 그림들은 직선 구간, 곡선 구간, 그림자 발생 구간에 대하여 대표선을 추출한 결과이다. 최종적으로 대표선에 의한 차선 검출 결과를 보면 직선 구간 (그림 9)과 그림자 구간 (그림 11)에서 정확하게 차선을 추출하는 것을 볼 수 있다. 그리고 그림 10은 곡선 구간에서의 대표선 추출 결과이다. 곡선 구간에서는 기울기 계산에 의한 대표선 추출이 실시간으로 동작 되어야 한다. 따라서 고속 주행의 곡선 구간에서는 기울기의 변화를 빠르게 적용하여 대표선을 추출해야 한다. 그림 10은 약 시속 80~100km로 곡선 주행을 하는 경우의 대표선을 추출한 결과이다. 대표선 추출은 각각 왼쪽과 오른쪽의 대표선을 기울기에 의하여 구

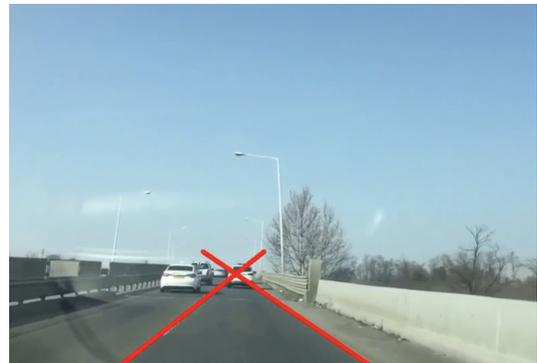


그림 9. 직선 구간의 대표선 추출 결과
Fig. 9. Representative line extraction of straight area



그림 10. 곡선 구간의 대표선 추출 결과
Fig. 10. Representative line extraction of curved area

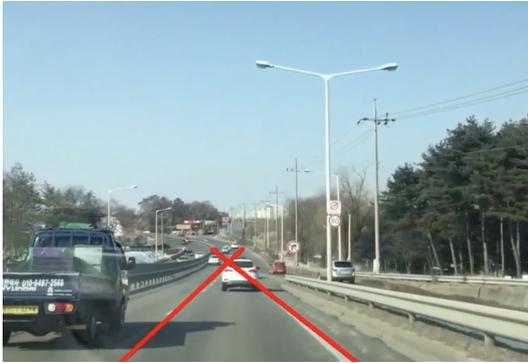


그림 11. 그림자 영역에서의 대표선 추출 결과
Fig. 11. Representative line extraction of shadow area

하게 되는데, 곡선 구간을 주행하는 이미지에 적용하는 경우 직선 구간을 주행하는 이미지에 비하여 그 정확도가 떨어지는 것을 확인할 수 있었다. 그러나 저속 주행이든 고속 주행이든 그 속도에는 영향이 없이 색 검출을 기반으로 하는 차선 검출 기법보다는 높은 정확도가 나옴을 알 수 있다. 그림 9에서 표현한 직선 구간의 경우 고속 주행 구간이 함께 적용된 결과이고, 그림 11에서 표현한 그림자 발생 구간의 경우 저속 주행 구간이 함께 적용된 결과이다. 두 결과 모두 대표선을 정확하게 추출하고 있다.

IV. 실험 결과

본 논문에서 제안한 차선 검출 방법을 검증하기 위해 시내 주행 및 고속도로에서의 실제 주행 차량을 대상으로 다양한 조건의 차선에 대해 실험을 수행하였다. 이를 위한 테스트 베드로 일반 승용차를 사용해서 고해상도의 블랙박스를 룸 미러 상단에 고정시켜 촬영한 영상을 사용했다.

기본적으로 영상 처리의 전체적인 처리량은 사용하는 프로세서의 환경이나 프로그램의 최적화 및 영상에 따라 변화되지만, 동일한 환경에서는 관심 영역을 추출한 후 히프 변환으로 직선 검출을 시행한 결과가 가장 적은 계산량으로 동작하는 것을 확인하였다. 이러한 결과는 ROI-LB 추출과 에지 방향성 정보에 의한 것으로 분석된다.

본 논문에서는 차량에 연결된 GPS를 통해 위치를 인식하여 저속의 시내 주행의 경우 Gray color code 색 변환을 적용하고, 고속 주행의 경우 HSV color code를 적용한 결과를 비교한다. 위 그림들은 Gray 색 변환에 의한 차선 검출 결과를 보여준다. 직선 구



그림 12. 직선 구간의 Gray 색 변환 결과
Fig. 12. Gray color conversion result of straight area



그림 13. 곡선 구간의 Gray 색 변환 결과
Fig. 13. Gray color conversion result of curved area



그림 14. 그림자 발생 구간의 Gray 색 변환 결과
Fig. 14. Gray color conversion result of shadow area

간 뿐만 아니라 곡선 구간까지도 Gray color code 색 변환을 적용한 차선 추출을 정확하게 하는 것을 알 수 있다. 특히 그림 14를 보면 그림자가 발생하게 되는 구간에서 Gray color code 색 변환을 적용하게 되면 그림자 영역까지도 검출을 하게 되는 것을 확인할 수 있다.

그림 15는 HSV color code 색 검출에 의한 차선 검출의 결과를 보여준다. 그림 15(c)는 그림자가 발생하는 구간에서의 HSV color code를 적용한 차선 검출의 결과이다. 결국 고속 주행에서는 옆 차선에서 발생하는 그림자의 영향에 둔감한 HSV color code를 사용하는 것이 Gray color code를 사용하는 것보다



그림 15. (a) 직선 구간의 HSV 색 검출 결과
Fig. 15. (a) HSV color conversion result of straight area



그림 16. (a) 직선 구간의 대표선 적용 결과
Fig. 16. (a) Representative line result of straight area



그림 15. (b) 곡선 구간의 HSV 색 검출 결과
Fig. 15. (b) HSV color conversion result of curved area



그림 16. (b) 곡선 구간의 대표선 적용 결과
Fig. 16. (b) Representative line result of curved area



그림 15. (c) 그림자 발생 구간의 HSV 색 검출 결과
Fig. 15. (c) HSV color conversion result of shadow area

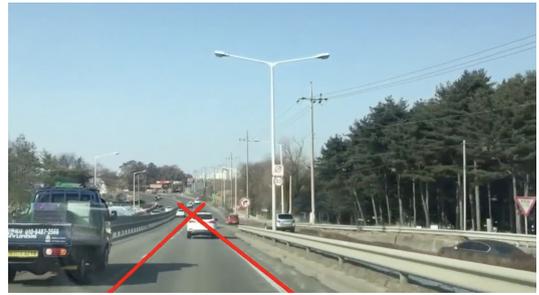


그림 16. (c) 그림자 발생 구간의 대표선 적용 결과
Fig. 16. (c) Representative line result of shadow area

차선 검출의 정확도가 더 높아진다.

추가적으로 허프 변환을 한 후 기울기 값을 계산하여 수평 기울기 값과 수직 기울기 값의 기준에 따라 대표선을 추출한 결과를 그림 16에서 볼 수 있다. 기존 허프 변환은 도로 영역을 고려하지 않고 전체 영역의 직선을 검색하여 연산하기 때문에 불필요한 연산 과정이 많다. 본 논문에서는 차선 검출 시스템에서 많은 시간이 소요되는 코너점 추출 과정과 허프 변환 과정을 GPU(Graphic Processing Unit)에서 병렬적으로 처리하게 함으로써 기존의 단순 허프 변환보다 약 6.2 배 빠른 연산 속도를 보이도록 구현하였다. 이는 실시간 처리를 위한 필수적인 과정이나, color code를 적

용한 결과에 추가적으로 적용했기 때문에 연산 속도가 빠르지 않거나 차량의 속도가 너무 빠르거나 곡선 구간의 기울기가 급격한 경우 정확도가 떨어질 수 있다.

표 1은 제안하는 기법에 따른 검출률 결과를 보여 준다. 결과 영상은 직선 구간, 곡선 구간, 저속 주행 구간과 고속 주행 구간이 포함되어 있다. OpenCV를 활용하여 960*540 픽셀 크기의 영상 1252장을 사용하여 검출률을 확인한 결과 표 1과 같이 평균 24fps의 영상에서 94.83%의 결과를 확인하였다. 미검출의 결과는 한쪽/양쪽 차선에 대하여 검출을 실패한 경우, 오검출의 경우 차선이 아닌 것을 검출한 경우로 평가

표 1. 실험결과와 검출률
Table 1. Detection rate of experiment results

	전체	검출	오류		정확도
			미검출	오검출	
Gray 색변환	418	386	22	7	93.06%
HSV 색검출	430	407	19	4	94.65%
대표선	404	391	10	3	96.78%
전체 결과	1252	1187	51	14	94.83%

하였다. 이 기준에 따라 평가 결과는 직선 차선의 구간, 곡선 차선의 구간 및 다양한 도로 환경에서도 차선 검출이 정확함을 보여준다. 특히 대표선의 경우 도로의 앞부분에 있는 꼭짓점을 기반으로 차량의 예상 진행 방향을 예측할 수 있음을 보였다. 또한 대표선의 결과는 오검출 되는 경우는 거의 발생하지 않았고 고속 주행의 곡선 구간에서의 미검출이 약간 발생함을 알 수 있었다. Gray color code의 경우 그림자 영역으로 인하여 미검출 되는 경우가 다른 기법에 비해 많이 발생함을 알 수 있었다.

V. 결 론

본 논문에서는 높은 정확성과 실시간성을 동시에 보장하는 차선 검출 알고리즘을 제안하였다. 차량 내 포함되어 있는 GPS를 통해 차량의 위치를 인식하여 저속 구간과 고속 구간에 따라 색 검출 기법을 적응적으로 변경할 수 있도록 하였다. 추가적으로 대표선을 추출할 수 있도록 하여 차선 검출에 대한 정확도를 높이고자 하였다. 제안하는 기법을 통하여 입력 영상의 환경이 변하더라도 안정적으로 차선을 찾을 수 있도록 하였다. 특히, 직선 구간과 곡선 구간, 저속 주행 구간과 고속 주행 구간을 모두 적용하여 제안하는 기법이 높은 검출률을 보이는 것을 확인하였다. 낮 시간대의 주행 결과 평균 94.83%의 높은 검출률을 보였다. 검출된 차선과 제어점을 바탕으로 차량의 차선 이탈 유무를 파악할 수 있다. 차량이 정상 주행할 경우 차선이 변함에 따라 제어점의 값도 새로운 값으로 갱신된다. 만약 제어점 값들이 새로운 값들로 갱신되지 않고 10 프레임 이상기존의 값을 계속 유지할 경우 차량이 차선을 이탈했다고 판단하고, 경고문 구를 출력하여 운전자에게 상황을 인지시킬 수 있다.

본 알고리즘은 기울기가 크게 변경되거나 빠른 고

속 구간에서 오답률이 발생하는 것을 알 수 있었다. 또한 주간의 영상으로 국한되어 있어 야간 운행과 환경 변화에 따른 연구가 추가적으로 진행될 예정이다.

References

- [1] M. Aziz, A. Prihatmanto, and H. Hindersah, "Implementation of lane detection algorithm for self-driving car on toll road cipularang using Python language," in *Proc. Int. Conf. Electric Veh. Technol.*, pp. 144-148, Paris, 2017.
- [2] H. Park, S. Park, and E. Kim, "A road condition-based routing and greedy data forwarding algorithm for VANETs," *Ad Hoc & Sensor Wireless Netw.*, vol. 33, no. 1, pp. 301-309, Jan. 2016.
- [3] M. Paula and C. Jung, "Automatic detection and classification of road lane markings using onboard vehicular cameras," *IEEE Trans. Neural Netw. and Learning Syst.*, vol. 28, no. 3, pp. 690-703, 2017.
- [4] H. Park, K. Singh, and K. Piamrat, "Density-based opportunistic broadcasting protocol for emergency situation in V2X networks," *J. Inf. Commun. Convergence Eng.*, vol. 12, no. 1, pp. 26-32, 2014.
- [5] T. Bente, S. Szaghalmy, and A. Fazekas, "Detection of lanes and traffic signs painted on road using on-board camera," in *Proc. IEEE Int. Conf. Future IoT Technol.*, pp. 1-7, Eger Hungary, 2018.
- [6] Y. Zhang, J. Wang, X. Wang, and J. Dolan, "Road-segmentation-based curb detection method for self-driving via a 3D-LiDAR sensor," *IEEE Trans. Intell. Transp. Syst.*, vol. 99, pp. 1-11, 2018.
- [7] C. Wang, S. Huang, and L. Fu, "Driver assistance system for lane detection and vehicle recognition with night vision," in *Proc. Int. Conf. Intell. Robots and Syst.*, pp. 3530-3535, Edmonton, Canada, 2005.
- [8] H. Yoo, M. Kang, and H. Sohn, "Environment adaptive canny edge detector for lane detection," *The Korean Soc. Board Eng.*, pp.

- 72-74, Jul. 2011.
- [9] J. Kim and G. Kim, "A binarization technique using histogram matching for license plate with a shadow," *J. Broadcast Eng.*, vol. 19, no. 1, pp. 56-63, 2014.
- [10] Q. Li, L. Chen, M. Li, and S. L. Shaw, "A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 520-555, Feb. 2014.
- [11] J. H. Bae and J. B. Song, "Monocular vision based lane detection using segmented regions from edge information," in *Proc. Int. Conf. URAI*, pp. 499-502, Nov. 2011.
- [12] H. Park, "Vehicle tracking system using HSV color space at nighttime," *J. Korea Inst. Inf., Electron., and Commun. Technol.*, vol. 8, no. 4, pp. 270-274, 2015.
- [13] J. Li, X. Mei, D. Prokhorov, and D. Tao, "Deep neural network for structural prediction and lane detection in traffic scene," *IEEE Trans. Neural Netw. and Learning Syst.*, vol. 28, no. 3, pp. 690-703, 2017.
- [14] M. Hanna and S. Kimmel, "Current US federal policy framework for self-driving vehicles: Opportunities and challenges," *Computer*, vol. 50, no. 12, pp. 32-40, 2017.
- [15] H. Cheng, C. Yu, C. Tseng, K. Fan, J. Hwang, and B. Jeng, "Environment classification and hierarchical lane detection for structured and unstructured roads," *IET Computer Vision*, vol. 4, no. 1, pp. 37-49, 2010.
- [16] R. Huang, B. Chang, Y. Tsai, and Y. Liang, "Mobile edge computing-based vehicular cloud of cooperative adaptive driving for platooning autonomous self driving," in *Proc. Int. Symp. Cloud and Serv. Comput.*, pp. 32-39, Kanazawa Japan, 2017.
- [17] C. Jang, S. Lee, S. Choi, and Y. Kim, "Realtime robust curved lane detection algorithm using gaussian mixture model," *J. Institute of Control, Robotics and Syst.*, vol. 22, no. 1, pp. 1-7, 2016.
- [18] U. Ozgunalp, R. Xiao, and A. Dahnoun, "Multiple lane detection algorithm based on novel dense vanishing point estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 621-632, 2017.
- [19] J. Yoo, S. Lee, S. Park, and D. Kim, "A robust lane detection method based on vanishing point estimation using the relevance of line segments," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3254-3266, 2017.
- [20] S. Chen, J. Shang, S. Zhang, and N. Zheng, "Cognitive map-based model: Toward a developmental framework for self-driving cars," in *Proc. Int. Conf. Intell. Transp. Syst.*, pp. 1-8, MIT in Cambridge, 2017.
- [21] M. Paula and C. Jung, "Automatic detection and classification of road lane markings using onboard vehicular cameras," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3160-3169, 2015.
- [22] B. Nugraha and S. Fahmizal S, "Towards self-driving car using convolutional neural network and road lane detector," in *Proc. Int. Conf. Automation, Cognitive Sci., Optics, Micro Electro-Mechanical Syst., and Inf. Technol.*, pp. 65-69, Jakarta Indonesia, 2017.
- [23] R. Satzoda, S. Sathyanarayana, T. Srikanthan, and S. Sathyanarayana, "Hierarchical additive hough transform for lane detection," *IEEE Embedded Syst. Lett.*, vol. 2, no. 2, pp. 23-26, 2010.
- [24] C. Cheong, "High speed road lane detection based on optimal extraction of ROI-LB," *J. Broadcast Eng.*, vol. 14, no. 2, pp. 253-264, 2009.

박 현 희 (Hyunhee Park)



2011년 8월 : 고려대학교 전자
컴퓨터공학과 박사 졸업

2017년 3월~현재 : 한국성서대
학교 컴퓨터소프트웨어학과
조교수

<관심분야> 전자공학, 통신공
학, 머신러닝, 빅데이터