

무인 자율 주행 자동차를 위한 횡단보도 및 정지선 인식 시스템

A Crosswalk and Stop Line Recognition System for Autonomous Vehicles

박태준·조태훈[†]

Taejun Park and Tai-Hoon Cho[†]

한국기술교육대학교 컴퓨터공학부

요 약

최근 무인 자율 주행 자동차를 실현하기 위한 기술 개발이 활발히 이루어지고 있는 추세이다. 본 논문에서는 무인 자율 주행 자동차의 핵심 기술인 컴퓨터 비전을 이용한 무인 자율 주행 자동차를 위한 횡단보도 및 정지선 인식 시스템을 제안한다. 본 논문의 컴퓨터 비전 시스템은 먼저 무인주행을 위하여 반드시 필요로 하는 차선을 RANSAC 알고리즘과 Kalman 필터를 이용하여 인식하고 인식된 차선이 실제로는 평행하다는 점을 이용하여 원근 시점인 입력 영상을 평면 시점으로 변환하여 횡단보도의 크기가 일정하게 만든다. 그런 후, 변환된 영상에서 횡단보도의 기하학적 특징을 이용하여 횡단보도를 인식하고 횡단보도 앞의 영역을 관심 영역으로 설정한 후 설정된 관심 영역에서 정지선을 추출한다. 구현된 알고리즘을 다양하게 실험한 결과 차선, 횡단보도, 정지선에 대하여 높은 인식률을 보였다.

키워드 : 컴퓨터 비전, 차선, 횡단보도, 인식

Abstract

Recently, development of technologies for autonomous vehicles has been actively carried out. This paper proposes a computer vision system to recognize lanes, crosswalks, and stop lines for autonomous vehicles. This vision system first recognizes lanes required for autonomous driving using the RANSAC algorithm and the Kalman filter, and changes the viewpoint from the perspective-angle view of the street to the top-view using the fact that the lanes are parallel. Then in the reconstructed top-view image this system recognizes a crosswalk based on its geometrical characteristics and searches for a stop line within a region of interest in front of the recognized crosswalk. Experimental results show excellent performance of the proposed vision system in recognizing lanes, crosswalks, and stop lines.

Key Words : Computer Vision, Lane, Crosswalk, Recognition

1. 서 론

1970년대부터 미국, 일본, 유럽 등 선진국에서는 지능형 차량과 도로 시스템에 관심과 투자가 점점 증대되었다. 지능형 자동차의 연구 개발은 이제 선택이 아닌 의무로 변해가는 추세이며 국내 자동차 업체들은 관련 기술의 연구개발에 주력하고 있다. 해외 주요 업체들은 이미 적응순항제어 시스템, 차선 유지 시스템, 자동 주차 시스템 등을 상용화하며 시장을 선도하고 있다.

본 논문은 무인 자율 주행 자동차에서 사용 가능한 차선

인식, 횡단보도 인식, 정지선 인식을 할 수 있는 컴퓨터 비전 시스템을 제안한다. 본 논문의 컴퓨터 비전 시스템은 차선을 인식하고 인식된 차선을 이용하여 원근 시점(Perspective view)인 입력 영상을 평면 시점(Top view)으로 변환한다. 변환된 영상을 이용하여 횡단보도를 인식하여 횡단보도 앞의 정지선을 인식한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에 관한 관련 연구를 이야기하고 3장에서는 비전 시스템을 구성하고 있는 알고리즘에 대한 설명을 하며 4장에서는 실험 결과에 대하여 이야기한다. 마지막으로 5장에서는 결론 및 향후 연구 방향에 대하여 논한다.

접수일자: 2010년 12월 28일

심사(수정)일자: 2012년 3월 30일

게재확정일자 : 2012년 4월 2일

†교신저자

이 논문은 한국기술교육대학교 교육연구진흥비지원 프로그램의 지원에 의하여 수행되었음.

2. 관련 연구

2.1 차선 인식

차선 인식 알고리즘은 이미 오래전부터 연구된 분야이다.

차선인식은 크게 도로 모델링, 특징점 추출, 후처리, 차선 인식과 같은 순서를 따른다.

2.1.1 도로 모델링

도로 모델링(Modeling)은 모델링 방법에 따라서 알고리즘의 성능을 크게 좌우한다. 도로 모델링은 역원근 워핑 영상(Inverse-perspective-warped image)에서 평행한 직선을 도로로 모델링하는 방법[1], 평면에서 차선의 폭을 이용하여 모델링하는 방법[2], 평면에서 포물선으로 모델링 하는 방법[3] 등 여러 가지 방법들이 있다.

2.1.2 특징점 추출

특징점 추출은 차선 인식에 있어서 제일 중요한 부분이다. 특징점 추출 방법으로는 에지 기반의 추출[1][11], 컬러기반 추출[4], 템플릿을 이용한 추출[5], 이산 코사인(Discrete Cosine Transform) 변환[6]을 이용한 추출 등 여러 가지 방법들이 있다. 도로와 차선의 특징점은 한가지의 특징점 추출 방법으로는 추출하기가 어렵다. 따라서 일반적으로는 특징 추출 방법을 한 가지 방법만 사용하지 않고 여러 가지 방법을 사용하여 특징점을 추출한다.

2.1.3 후처리

후처리(post-processing) 과정은 추출된 특징점에서 이미 알고 있는 지식을 이용하여 차선을 추론해 내는 과정이다. 차선을 추론하는 방법으로는 허프 변환(Hough transform)[7-8]을 이용하는 방법, 신경망(Neural network)[9-10]을 이용하는 방법 등이 있다.

2.1.4 차선 인식

후처리 과정을 거친 차선의 후보군들 중 실제 차선만을 선택하는 부분이다. 영상에서 차선은 현재 차량이 속해있는 차선만 있는 것이 아니라 다른 차선이 존재할 수 있다. 이러한 차선들 중 필요한 차선만을 필터링(Filtering)하는 과정이 필요하다.

2.2 횡단보도 인식

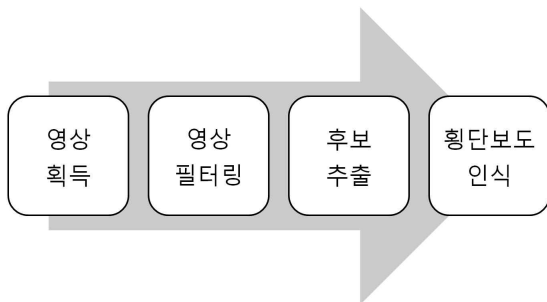


그림 1. 일반적인 횡단보도 인식 알고리즘의 순서
Fig. 1. General diagram of crosswalk recognition

횡단보도 인식은 일반적으로 작은 커널을 이용한 에지 추출 방법과 미리 알고 있는 사전 지식(Prior knowledge)[12-15]을 이용한다. [12]에서는 Figure-ground 추출 방법을 사용한 것에 반하여 [13]에는 추출을 하기 위하여 요소 그래프(Factor graphs)를 이용하였다. [14]는 검출된 에지들을 Homographic framework로 융합하였다. 또한 [15]는 스테레오 영상을 횡단보도 인식에 사용하였다. 이러

한 방법들은 필터링하는 방법의 차이이다. 필터링은 횡단보도 영역을 좀 더 쉽게 추출할 수 있도록 하는 역할을 하며 횡단보도 인식 속도에 영향을 준다. 본 논문은 횡단보도의 특징점을 구하기 위하여 에지의 페어를 이용하여 필터링 하였다. 횡단보도 인식은 보통 4단계의 순서를 가지며 그 순서는 그림 1과 같다.

3. 알고리즘

본 비전 시스템의 전체적인 작동순서는 그림 2와 같다.

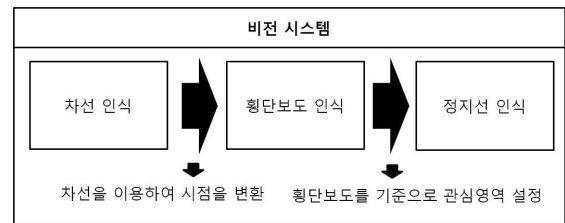


그림 2. 비전 시스템의 작동 순서
Fig. 2. The diagram of the vision system

3.1 차선 인식

3.1.1 차선 모델링

그림 3에서 보는 것과 같이 1차 함수로 정의하였다. 이러한 단순한 모델링의 장점과 단점은 다음과 같다. 차선을 $y = ax + b$ 로 정의하게 되면 식 자체가 간단하기 때문에 연산량이 많지 않게 된다. 이는 곧 실시간으로 처리하기 보다 더 쉬워진다는 것을 의미한다. 하지만 실제 도로는 직선으로만 이루어져있지 않아 곡선으로 이루어진 차선은 정확하게 표현되지 않는 단점이 생긴다.

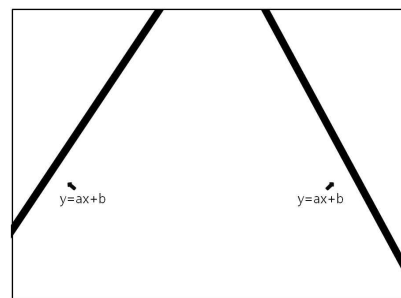


그림 3. 차선 모델링
Fig. 3. Lane modeling

3.1.2 특징점 추출

카메라에서 영상을 획득하게 되면 제일 먼저 차선의 특징점들을 추출하여야 한다. 차선의 특징점들을 추출하려면 차선의 특징을 우선 알아야 한다. 본 비전 시스템에서 사용하는 차선의 특징은 다음과 같다.

- ◆ 차선은 도로면보다 밝다.
- ◆ 차선은 일정한 굵기를 가지고 있다.

실제로 차선의 두께는 15cm의 굵기를 가지고 있다. 또한

차선은 아스팔트 혹은 시멘트로 만들어진 도로보다 밝은 흰색, 파란색, 주황색이다. 이러한 특징을 이용하여 차선의 특징점을 추출한다. 특징점 추출 알고리즘의 순서는 그림 4와 같다. 입력 영상의 잡영(Noise)을 줄이기 위해서 가우시안 블러(Gaussian blur)를 적용한다. 이때 커널(Kernel)의 크기는 5이다. 또한 차선의 특징은 저해상도 영상에서도 추출이 가능하기 때문에 연산속도를 위하여 640(W)x480(H)의 영상을 320(W)x240(H) 해상도로 크기를 줄인 다음 에지를 추출한다. 차선은 차량에 대하여 거의 수직으로 있기 때문에 수직 성분만을 탐지할 수 있는 연산자인 식 (1)의 1차원 에지 추출 연산자(1D Edge Extraction Operator)를 사용한다.

$$G_x = [-1 -2 0 +2 +1] \quad (1)$$

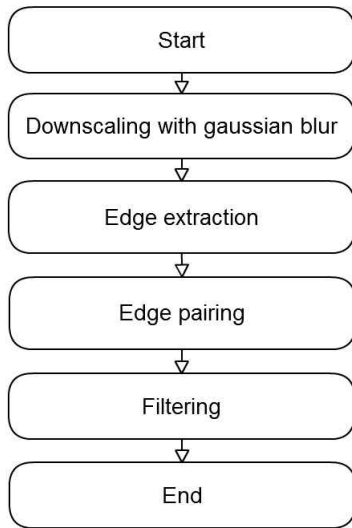


그림 4. 특징점 추출 순서도
Fig. 4. Feature extraction diagram

식 1의 G_x 를 영상에 회선(Convolution)을 시키면 각 픽셀에 대한 에지 강도(Edge intensity) 값을 얻을 수 있다. 1차원 에지 추출 연산자를 적용하면 에지는 상승 에지(Positive edge)와 하강 에지(Negative edge)로 나뉘게 된다. 상승 에지는 어두운 곳에서 밝은 곳으로 영상의 밝기가 변화할 때 나오며 하강 에지는 밝은 곳에서 어두운 곳으로 영상의 밝기가 변화할 때 나타난다. 이러한 특징을 이용하여 에지들을 페어링(Pairing)한다. 에지 페어링을 하고 나면 각 페어에 대하여 필터링을 한다. 페어링을 하면 페어링된 에지 사이의 거리를 구할 수 있다. 이 거리는 곧 차선의 굵기이다. 실제로 좌표계 기준으로 차선의 굵기는 15cm로 일정하다. 이러한 특징을 이용하여 너무 작거나 너무 큰 거리의 페어들을 제거한다. 페어들을 제거한 후 각 페어별로 특징점이 하나만 되도록 한다. 특징점의 위치는 각 페어의 상승 에지와 하강 에지의 중앙으로 하였다. 그림 5는 특징점을 단순화 시킨 영상이다.

3.1.3 차선 추출

차선은 왼쪽 차선과 오른쪽 차선으로 나뉘어져 있다. 따라서 차선 추출을 위해서 처리 영역을 중앙을 기준으로 좌우로 나누어 독립적으로 차선 추출을 한다.

좌우로 나뉜 특징점들을 RANSAC[16] 직선 근사(line fitting)를 이용하여 차선을 추출한다. 차선을 추출 후 해당 차선의 기울기를 비교하여 차선여부를 다시 한 번 판별한다. 판별 결과 차선이라고 확정되면 칼만 필터를 적용하고 좌우 차선중 하나라도 차선이라고 인식되지 않으면 가장 최근에 추출한 차선을 선택한다. 그림 6은 차선 추출 알고리즘 순서를 나타내고 그림 7은 차선을 추출한 영상이다.

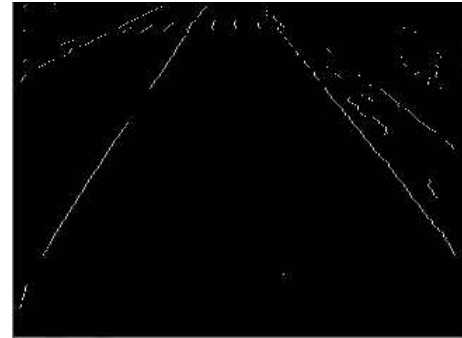


그림 5. 차선 특징점 추출 영상

Fig. 5. An image of lane features extracted

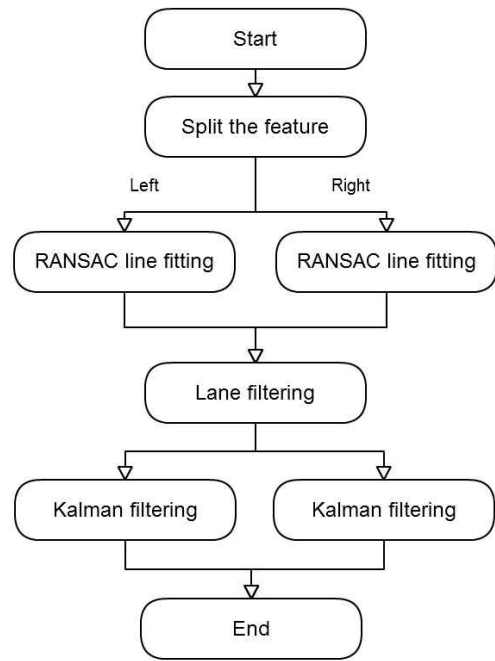


그림 6. 차선 추출 알고리즘 순서도

Fig. 6. A diagram of lane extraction algorithm

3.2 횡단보도 인식

횡단보도는 ‘도로교통법 시행규칙 제11조(횡단보도의 설치기준) <별표 6> 5. 노면표시’에서 제시한 형식인 그림 8과 같은 규격으로 노면에 표시되어야 한다. 단위는 센티미터(cm)이다. 그림 8에 나와 있는 규격을 바탕으로 횡단보도의 특징점을 파악하여 횡단보도 인식에 사용한다.

3.2.1 시점 변환

카메라에서 획득한 영상은 원근 투영(Perspective pro-

jection)된 영상이다. 원근 투영 영상에서는 횡단보도가 카메라의 위치에 따라 크기가 변화하기 때문에 인식하기가 어려워진다. 이러한 영상에서 횡단보도의 크기가 항상 같을 수 있도록 평면 시점(Top view) 영상으로 시점(View point) 변환한다. 시점 변환을 위해서는 원근 투영된 영상에서 2개의 평행한 직선을 검출해야 변환할 수 있다. 차선은 항상 평행하기 때문에 3.1에서 추출한 차선을 이용하여 시점 변환을 한다. 그림 9는 추출된 차선을 이용하여 시점을 변환한 영상이다.

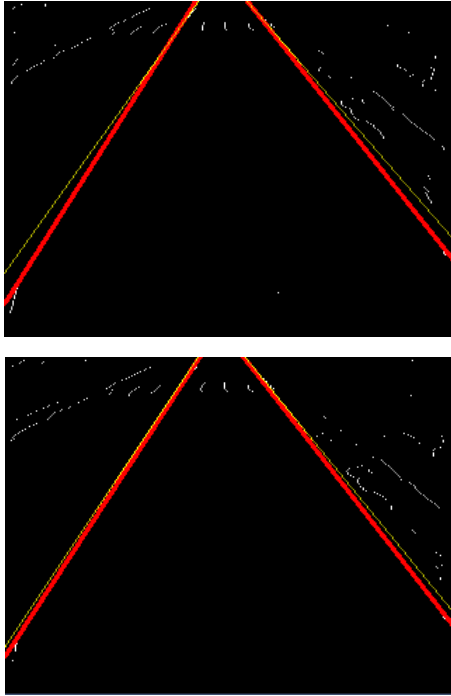


그림 7. 차선 추출 영상
Fig. 7. Lane extraction images

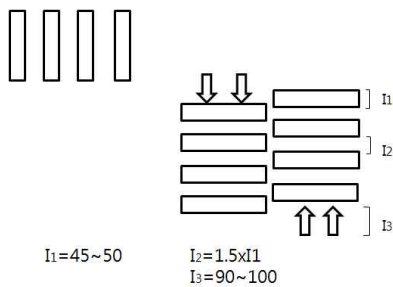


그림 8. 횡단보도 규격
Fig. 8. Crosswalk specification

3.2.2 특징점 추출

횡단보도의 특징점 추출 방법은 차선의 특징점 추출 방법과 유사하다. 횡단보도의 특징점 추출은 차선 특징점 추출의 단계 중 필터링 단계만 다르고 나머지 부분은 같다. 따라서 필터링 부분만 설명한다.

필터링은 차선을 추출할 때와 같이 에지간의 거리를 기반으로 필터링한다. 그림 8에서 알 수 있듯이 횡단보도의 굵기는 0.45~0.5m이다. 이 두께를 바탕으로 에지의 거리를

필터링 한다. 필터링 한 결과는 그림 10과 같다.



그림 9. 평면으로 시점을 변환한 영상
Fig. 9. The image from the view point changed to top view

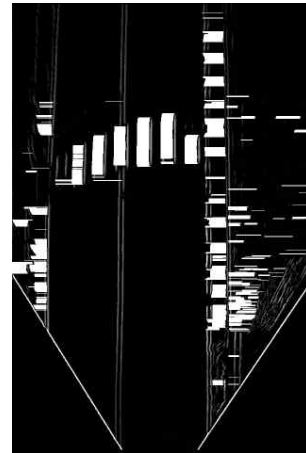


그림 10. 횡단보도 특징점 결과 영상
Fig. 10. A crosswalk feature extracted image

3.2.3 횡단보도 영역 추출

횡단보도의 영역을 추출하기 위해서 가로 방향의 라인별로 페어의 개수를 센다. 그림 10에서 볼 수 있듯이 횡단보도 영역의 라인은 특징점의 개수가 많다. 이러한 특징을 이용하여 횡단보도 영역을 추출한다. 그림 11의 검은색 사각형은 추출한 횡단보도 영역을 나타낸다.

3.3 거리 측정

이 절에서는 간단하게 거리를 측정할 수 있는 방법에 대하여 설명한다.

3.3.1 카메라 캘리브레이션

영상 좌표계에서 실세계 좌표계로 변환하려면 카메라 캘리브레이션(Camera calibration)이 필요하다. 카메라 캘리브레이션이란 내부 파라미터(Intrinsic parameter)와 외부 파라미터(Extrinsic parameter)를 구하는 것이다. 본 논문은 정밀하게 카메라 캘리브레이션을 하지 않고 대략적인 거리만을 구하기 위하여 5m 거리마다 막대

를 놓아 촬영하여 Catmull-Rom spline으로 추정하지 않은 빈곳을 보정하였다. 이렇게 함으로써 대략적인 위치를 알 수 있게 되었다.

3.3.2 Catmull-Rom spline

Catmull-Rom spline은 좌표 4개를 가지고 곡선을 만들어 내는 보간법 중의 하나이다. Catmull-Rom spline은 식 2와 같다.

$$CINT_x(p_{-1}, p_0, p_1, p_2) = \frac{1}{2} \begin{pmatrix} x((2-x)x-1) \\ x^2(3x-5)+2 \\ x((4-3x)x+1) \\ (x-1)x^2 \end{pmatrix} \cdot \begin{pmatrix} p_{-1} \\ p_0 \\ p_1 \\ p_2 \end{pmatrix} \quad (2)$$

식 2의 식에 좌표 4개(p_{-1}, p_0, p_1, p_2)를 넣으면 p_0, p_1 좌표 사이의 공간이 곡선으로 보간된다. x 는 $x \in [0, 1]$ 의 범위의 값으로 p_0, p_1 사이의 비율을 나타낸다.

3.4 정지선 인식

3.4.1 관심 영역 설정

차량이 횡단보도를 인식하여 올바르게 서기 위해서는 정지선의 위치를 인식하여야 한다. 정지선 인식은 횡단보도 인식을 사용한 평면(Top view) 영상을 이용한다. 연산속도 향상과 오인식을 방지하기 위하여 관심영역을 횡단보도 아래쪽 1~6미터 영역으로 설정한다. 그림 11의 흰색 사각형은 관심영역을 나타낸다.



그림 11. 정지선 인식을 위한 관심영역

Fig. 11. A region of interest for stop line recognition

3.4.2 에지 추출

정지선의 특징점 추출 역시 차선이나 횡단보도 특징점 추출 방법과 비슷하지만 식(1)의 G_x 가 아닌 식(3)의 G_y 를 사용한다.

$$G_y = \begin{bmatrix} -1 \\ -2 \\ 0 \\ +2 \\ +1 \end{bmatrix} \quad (3)$$

식 (3)의 G_y 는 수직 방향의 에지를 추출해내는 1D 에지

검출기이다. 차선이나 횡단보도는 영상에서 세로로 나타나는 반면에 정지선은 가로로 나타나기 때문에 G_x 를 사용하지 않고 G_y 를 사용한다.

3.4.3 에지 페어링

횡에지 페어링 단계는 차선 인식에서 페어링 한 것과 대부분 비슷하다. 다른 점은 페어링하는 방향이 가로방향에서 세로 방향으로 바뀌었다.

3.4.4 필터링

에지 페어링을 하면 정지선이 아닌 페어들도 검출된다. 이러한 페어를 제외시키기 위하여 영상에서 나타나는 정지선의 굵기를 측정하여 필터링 한다. 그림 12는 페어의 에지 간의 거리를 이용하여 필터링한 영상이다.



그림 12. 필터링 결과 영상

Fig. 12. The result image of filtering

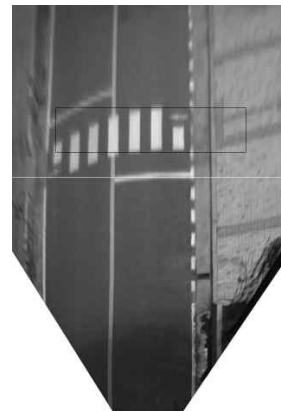


그림 13. 정지선 인식 결과 영상

Fig. 13. The result image of stop line recognition

3.4.5 정지선 영역 추출

정지선 영역의 추출은 횡단보도 영역 추출 방법과 대부분 같다. 정지선의 특징들은 가로로 균집되어 있기 때문에 가로 방향의 라인별로 페어들의 개수를 세어 구분한다. 정지선은 일정한 두께의 가로로 긴 선이다. 영상에서 보이는 정지선의 두께와 길이를 문턱값으로 설정하여 정지선을 추출한다. 또한 검출된 차선의 위치 중 차량과 제일 가까운 정지선의 아래쪽을 정지선으로 설정한다. 그림 13은 정지선 인식 결과 영상이다. 검정색 사각형은 인식한 횡단보도를 나타내고 흰색 가로선은 정지선의 위치를 나타낸다.

4. 실험 결과

실험을 위하여 교내의 횡단보도를 촬영하여 이용하였다. 실험 영상은 640(W)x480(H)의 흑백 영상으로 15

FPS(Frames per second)로 촬영되었다. 촬영에 사용된 카메라는 PointGrey사의 Chameleon 카메라(USB 연결방식)를 사용하였다.

실험 방법은 다음과 같다. 실험 영상을 이용하여 비전 시스템을 동작 시킨다. 영상 내에 차선, 횡단보도 그리고 정지선이 존재하는 이미지에서 알고리즘이 차선, 횡단보도 그리고 정지선을 인식하는지 알아본다. 또한 칼만 필터의 동작 여부를 알아본다.

4.1 알고리즘 테스트

4.1.1 차선 인식

실험 세트에 대하여 차선의 인식률은 표 1과 같은 결과가 나타났다.

표 1. 차선 인식률
Table 1. Lane recognition rate

세트	영상 개수	인식 개수	인식률
학교 내	1026	1003	97.8 %

영상 중 차선이 한쪽이라도 나타나지 않은 영상과 차선이 영상 내에 있지만 차량이 완전히 한 차선 안에 있지 않은 경우 인식률 테스트를 위한 테스트에서 제외하였다.

4.1.2 횡단보도 인식

횡단보도를 테스트한 결과는 표 2와 같다. 횡단보도 인식률은 횡단보도가 영상에 존재하는 영상 중 횡단보도가 이미지 높이의 140픽셀 이상 아래로 내려왔을 때부터 횡단보도의 정지선 끝의 높이가 400픽셀 밑으로 내려올 때까지의 영상을 테스트 세트로 하였다.

4.1.3 정지선 인식

정지 신호를 보낼 때의 영상에서 정지선 위치에 대한 분석 결과는 표 3과 같다.

표 2. 횡단보도 인식률
Table 2. Crosswalk recognition rate

세트	영상 개수	인식 개수	인식률
학교 내	180	180	100 %

표 3. 정지선 위치
Table 3. Stop line position

	평균	표준편차
정지선	408.13	8.21

표 3은 정지선의 위치가 400픽셀 이상일 때의 평균과 표준편차를 나타낸다. 비전 시스템의 특성상 400픽셀 이하로는 나오지 않는다. 3.3절을 이용하여 실제계 좌표계로 변환을 하면 408픽셀 기준으로 1픽셀에 0.01m가 된다. 따라서 표준편차인 8.21은 0.082m가 된다.

4.2 칼만 필터

그림 14는 칼만 필터의 동작 여부를 알기 위하여 차선 모델링 식인 $ax + b = 0$ 에서 인자 a에 대한 그래프를 나타낸 것이다. 그림 14에서 세로 축은 인자 a의 값이고 가로 축은 프레임의 수이다. 실제 차선은 영상에서 연속적으로 나오게 된다. 따라서 칼만 필터를 적용한 인자 a를 보면 적용전의 급격한 변화를 보이는 인자 a의 값이 적용 후 좀 더 부드럽게 변하는 것으로 보아 바르게 동작하는 것으로 유추할 수 있다.

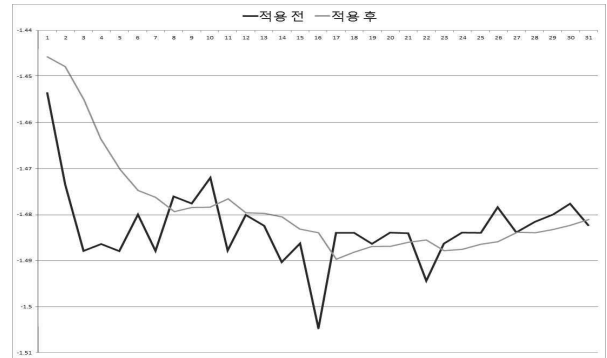


그림 14. 인자 a에 대한 칼만 필터 적용 전 과 후 그래프

Fig. 14. The graph of parameter a using Kalman filter

4.3 처리 속도

본 비전 시스템의 처리 속도는 차선과 횡단보도의 특징점이 화면에 얼마나 많이 있는지에 따라 결정된다. 총 280개의 영상을 가지고 처리 속도를 측정하였다. 측정된 결과 평균 처리속도는 38ms이다. 이는 초당 26프레임을 처리할 수 있는 연산속도이다.

5. 결론 및 향후 연구 방향

5.1 결론

본 논문에서는 무인 자율 주행 자동차의 핵심 기술인 컴퓨터 비전을 이용한 무인 자율 주행 자동차를 위한 횡단보도 및 정지선 인식 시스템을 제안하였다. 무인 자율 주행 자동차를 위하여 꼭 필요한 차선 인식, 횡단보도 인식, 정지선 인식을 구현하고 실험해본 결과 횡단보도 인식과 정지선 인식을 올바르게 수행하는 것을 볼 수 있다. 본 논문의 차선 인식과 횡단보도 인식 사이에는 종속성이 존재한다. 즉 정지선을 인식하기 위해서는 횡단보도가 먼저 인식되어야 한다. 이러한 단점을 줄이기 위해서 차선 필터링 과정을 통해 잘못 추출된 차선을 필터링하여 올바르게 인식한 가장 최근의 차선을 다시 사용한다. 그 결과 본 시스템을 실행시킨 후 차선을 성공적으로 한번이라도 인식한 경우에는 차선의 인식에 실패하더라도 이전 차선 데이터를 이용하여 성공적으로 횡단보도를 인식하는 결과를 볼 수 있었다.

5.2 향후 연구 방향

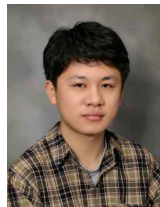
본 비전 시스템은 비전 시스템의 무결성을 위하여 횡단보도를 인식한 후 진짜 횡단보도인지를 판단을 해야 할 필

요성이 있다. 또한 2개의 카메라를 사용하여 깊이 정보를 얻어 차선이나 횡단보도가 아닌 특징점에 대해서 필터링해야 할 필요성이 있다. 향후 연구방향은 앞서 언급한 문제들을 보완하여 좀 더 지능적이고 강인하며 빠른 시스템을 개발하는 것이다.

참 고 문 헌

- [1] E. D. Dickmanns and B. D. Mysliwetz, "Recursive 3-D road and relative ego-state recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 199 - 213, Feb. 1992.
- [2] M. Bertozzi and A. Broggi, "GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Trans. Image Process.*, vol. 7, no. 1, pp. 62 - 81, Jan. 1998.
- [3] D. Pomerleau and T. Jochem, "Rapidly adapting machine vision for automated vehicle steering," *IEEE Expert-Special Issue on Intelligent System and Their Applications*, vol. 11, no. 2, pp. 19 - 27, Apr. 1996.
- [4] K. Kluge and C. Thorpe, "The YARF system for vision-based road following," *Math. Comput. Model.*, vol. 22, no. 4 - 7, pp. 213 - 233, Aug. 1995.
- [5] C. Taylor, J. Košecká, R. Blasi, and J. Malik, "A comparative study of vision-based lateral control strategies for autonomous highway driving," *Int. J. Robot. Res.*, vol. 18, no. 5, pp. 442 - 453, May 1999.
- [6] C. Kreucher and S. Lakshmanan, "LANA: A lane extraction algorithm that uses frequency domain features," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 343 - 350, Apr. 1999.
- [7] Q. Li, N. Zheng, and H. Cheng, "Springrobot: A prototype autonomous vehicle and its algorithms for lane detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 300 - 308, Dec. 2004.
- [8] J. B. McDonald, "Detecting and tracking road markings using the Hough transform," in *Proc. Irish Machine Vision and Image Processing Conf.*, Maynooth, Ireland, pp. 1 - 9, 2001.
- [9] D. Pomerleau, "Neural network vision for robot driving," in *The Handbook of Brain Theory and Neural Networks*, M.Arbib,Ed. Cambridge, MA: MIT Press, 1995.
- [10] S. Baluja, "Evolution of an artificial neural network based autonomous land vehicle controller," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 26, no. 3, pp. 450 - 463, Jun. 1996.
- [11] 림청, 한영준, 한현수, "복잡한 환경에서 Grid기반 모폴로지와 방향성 에지 연결을 이용한 차선 검출 기법", *한국지능시스템학회 논문지*, vol. 20, no. 6, pp. 786-792, 2010.
- [12] J. M. Coughlan and H. Shen, "A fast algorithm for finding crosswalks using figure-ground segmentation," in *Proc. 2nd Workshop on Applications of Computer Vision*, in conjunction with ECCV, p. 2, 2006.
- [13] V. Ivanchenko, J. Coughlan, and H. Shen, "Detecting and locating crosswalks using a camera phone", *Computer Vision and Pattern Recognition Workshops*, 2008.
- [14] S. Se, "Zebra-crossing detection for the partially sighted," in *Proc. IEEE Conf on Computer Vision and Pattern Recognition*, vol. 2, pp. 211 - 217, 2000.
- [15] B. Soheilian, N. Paparoditis, D. Boldo, and J. Rudant, "3d zebra crossing reconstruction from stereo rig images of a ground-based mobile mapping system", *IEVM06*, 2006.
- [16] Martin A. Fischler and Robert C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography", *Communication of the ACM*, vol. 24, June. 1981.

저 자 소 개



박태준(Taejun Park)

2008년: 한국기술교육대학교 컴퓨터공학과 학사
 2008년~현재: 한국기술교육대학교 전
 기전자공학과 석사과정

Phone : 010-3487-5700
 E-mail : crazyvirus@gmail.com



조태훈(Tai-Hoon Cho)

1981년: 서울대학교 전자공학과 학사
 1983년: 한국과학기술원 전기 및 전자공학과 석사
 1991년: Virginia Polytechnic Institute & State University 박사
 1992년~1998년: LG산전 연구소 책임/수석 연구원
 1998년~현재: 한국기술교육대학교 컴퓨터공학부 교수

Phone : 041-560-1351
 Fax : 041-560-1462
 E-mail : thcho@kut.ac.kr