

# 실시간 무인 자동차 제어를 위한 강인한 차선 검출 알고리즘

## Robust Lane Detection Algorithm for Realtime Control of an Autonomous Car

한 명 희<sup>1</sup>, 이 건 홍<sup>2</sup>, 조 성 호<sup>3</sup>

Myounghee Han<sup>1</sup>, Keonhong Lee<sup>2</sup>, Sungho Jo<sup>3</sup>

**Abstract** This paper presents a robust lane detection algorithm based on RGB color and shape information during autonomous car control in realtime. For realtime control, our algorithm increases its processing speed by employing minimal elements. Our algorithm extracts yellow and white pixels by computing the average and standard deviation values calculated from specific regions, and constructs elements based on the extracted pixels. By clustering elements, our algorithm finds the yellow center and white stop lanes on the road. Our algorithm is insensitive to the environment change and its processing speed is realtime-executable. Experimental results demonstrate the feasibility of our algorithm.

**Keywords:** Lane Detection, Yellow Center Lane, White Stop Line, RGB, Autonomous, Realtime Control

### 1. 서론

최근 무인 자동차에 대한 관심이 점점 높아지면서 국내외에서 그와 관련된 연구가 활발하게 진행되고 있다. 국외의 유명한 무인 자동차 경진대회인 DARPA를 위해 많은 연구가 진행되었고<sup>[1][4]</sup>, 2010년 10월 11일에는 Google에서 개발한 무인 자동차가 1,600km를 주행하는데 성공하였다. 또한 국내에서도 최근 현대기아차에서 “2010 미래자동차 기술공모전 (부제: 무인자율주행 자동차 경진대회)”을 개최하는 등 국내외적으로 무인 자동차 기술에 대한 관심이 높아지고 있다.)

무인 자동차가 도로를 주행할 때 영상정보의 획득은 필수적이므로 도로에서 안전하게 주행하기 위해서는 취득한 영상으로부터 무인 자동차가 주행하는 차선과 정지

선을 인식해야한다. 이러한 목적을 달성하기 위해 HSI 컬러 모델에서 관심영역을 지정하여 빛의 강도의 분포 차이와 fuzzy c-means algorithm을 사용하여 차선을 검출하는 알고리즘<sup>[5]</sup>, CIELab 컬러 모델에서 색상 클러스터링을 통하여 차선을 검출하는 알고리즘<sup>[6]</sup>, 기본적인 차선의 색상정보를 기반으로 하여 동일한 색상, 차선의 모양, 크기, 움직임 정보 등을 이용하여 실제 차선을 검출하는 알고리즘<sup>[7]</sup> 등 색상정보와 차선의 형태적 특성을 사용하는 다양한 알고리즘이 있다. 하지만 영상은 태양광의 영향을 매우 많이 받고, 카메라가 촬영하는 도로 주변의 상황이 항상 같지 않기 때문에 카메라로부터 획득된 영상을 처리하여 조도와 주변 환경으로부터 받는 영향을 줄이고, 처리된 영상으로부터 원하는 정보를 분리해 낼 수 있어야 한다. 또한 무인 자동차를 실시간으로 제어하기 위한 적절한 영상 처리 속도를 확보하여야 한다.

본 논문에서는 무인 자동차를 실시간으로 제어할 수 있는 빠른 영상 처리속도로 정해진 값에 의해서가 아닌 환경에 따라 적응적으로 영상에서 원하는 색상을 검출하고 검출된 정보를 기반으로 차선과 정지선을 인식할 수 있는 알고리즘을 제안하고자 한다.

Received : Dec. 23. 2010; Reviewed : Mar. 07. 2011; Accepted : Mar. 07. 2011

※ 이 논문은 교육과학기술부의 재원으로 한국과학재단의 지원 (No.2010-0015226) 및 지식경제부/한국산업기술진흥원 융복합형로봇전문인력 양성사업의 지원으로 수행되었음.

<sup>1</sup> KAIST 전산학과 석사과정

<sup>2</sup> KAIST 로봇학제 석사과정

<sup>3</sup> KAIST 전산학과 교수

## 2. 알고리즘 설명

그림 1은 간략하게 전체 알고리즘이 동작하는 플로차트를 보여준다. 본 알고리즘에서는 카메라를 통해 받은 하나의 3채널 RGB 영상을 3개의 1채널 R, G, B 영상으로 분할하고, 각 채널에서 노란색과 흰색의 픽셀이 도로 위에서 보이는 특징을 이용하여 픽셀을 검출한다.

그림 2는 노란색 픽셀이 도로위에서 보이는 특징을 보여준다. 영상의 처리 속도를 빠르게 하기 위해 차선 및 정지선 검출 알고리즘에 사용할 영상 처리의 최소의 단위는 그림 3과 같은  $n \times n$  크기의 픽셀의 집합이다. 이 픽셀의 집합을 차선원소라고 정의한다. 차선을 검출하기 위해 차선원소의 집합인 클러스터를 만들고, 최소자승법 등의 알고리즘을 사용하여 차선을 검출하고 판단한다.

본 논문에서 사용할 좌표계는 그림 4(a)와 같이 정의한다. 영상의 좌하단 모서리를 원점으로 하여 우측을  $x$ 축의 양의 방향, 상단을  $y$ 축의 양의 방향으로 하고, 눈금 하나의 크기는 차선원소의 한 변의 길이  $n$ 으로 정의한다.



그림 3. 차선원소 형성 예시.

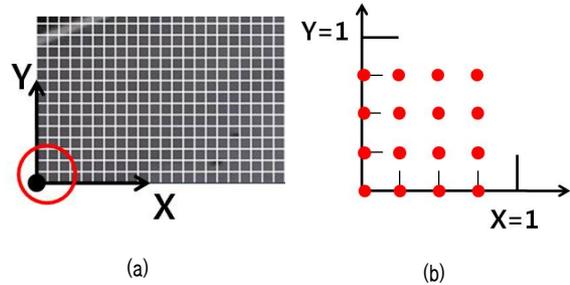


그림 4. (a) 좌표계 예시. (b)는 (a)의 등그리미 부분을 확대한 모습으로  $n=4$ 인 경우  $x < 1, y < 1$  범위에서 총 16개의 픽셀이 포함되어 있는 것을 나타냄.

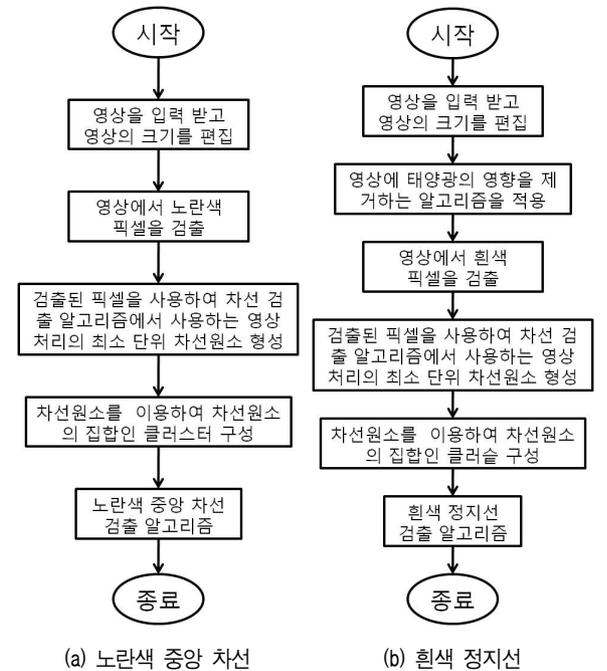


그림 1. 알고리즘 동작 과정.



그림 2. R, G, B 각 채널에서 보이는 도로위의 노란색 픽셀

### 2.1 영상 입력과 편집

카메라를 통해 받아들인 영상은 가로, 세로 길이를  $n$ 의 배수로 편집하고, 흰색의 정지선을 검출 할 때는 도로가 태양광의 영향을 많이 받기 때문에 태양광의 영향을 제거해주는 알고리즘<sup>[8]</sup>을 사용하여 영상을 처리한다.

### 2.2 픽셀 검출

영상에서  $y < 1$  범위의 픽셀에 대한 히스토그램을 분석하고 평균과 표준편차를 구하면 그림 5(a)와 같고, 동일한 범위에서 노란색 픽셀에 대한 히스토그램을 분석하면 그림 5(b)와 같다. 그렇기 때문에 동일한 범위에서 구해진 전체 픽셀의 평균과 표준편차의 합을 이용하면 그림 5(b)의 검출영역을 설정할 수 있다.

이를 일반화하면 자연수 집합  $N$ 에 대하여  $1 \leq y < i+1$ , ( $i \in N$ ) 범위 내에 있는 R, G, B 각 채널에서의 모든 픽셀의 평균값을  $E_{YR}, E_{YG}, E_{YB}$ , 표준편차를  $\sigma_{YR}, \sigma_{YG}, \sigma_{YB}$ 라 하고, 범위 내에 하나의 픽셀에 대한 R, G, B 각 채널의 픽셀 값을  $Y_R, Y_G, Y_B$ 라고 할 때, 검출되는 픽셀은 식 (1)을 만족시켜야 한다.

$$\begin{aligned} Y_R &> E_{YR} + \sigma_{YR} \\ Y_G &> E_{YG} + \sigma_{YG} \\ Y_B &< E_{YB} + \sigma_{YB} \end{aligned} \tag{1}$$

흰색 픽셀 또한 자연수 집합  $N$ 에 대해  $j \leq x < j+1$ , ( $j \in$

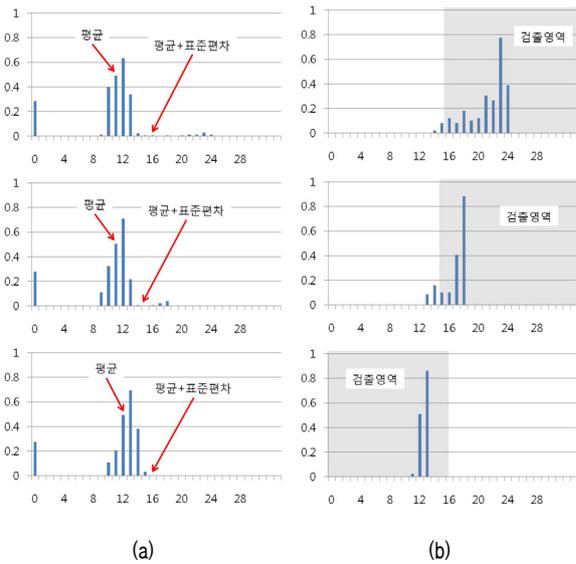


그림 5. (a)는 영상에서  $y < 1$  범위의 픽셀 값에 대한 히스토그램과 평균, 평균과 표준편차의 합을 보여줌. (b)는 (a)와 동일한 영역에서 노란색 픽셀의 픽셀 값에 대한 히스토그램과 검출되는 영역을 나타냄. 위부터 R, G, B 채널. 가로 축은 0~255의 픽셀 값을 32 구간으로 나눈 범위, 세로축은 정규화.

N) 범위 내에 있는 R, G, B 각 채널에서의 모든 픽셀의 평균값을  $E_{WR}$ ,  $E_{WG}$ ,  $E_{WB}$ , 표준편차를  $\sigma_{YR}$ ,  $\sigma_{YG}$ ,  $\sigma_{YB}$ 라 하고, 범위 내에 하나의 픽셀에 대한 R, G, B 각 채널의 픽셀 값을  $W_R$ ,  $W_G$ ,  $W_B$ 라고 할 때, 흰색 픽셀로 검출되는 픽셀은 식 (2)를 만족시켜야 한다.

$$\begin{aligned} W_R &> E_{WR} + \sigma_{WR} \\ W_G &> E_{WG} + \sigma_{WG} \\ W_B &> E_{WB} + \sigma_{WB} \end{aligned} \quad (2)$$

위와 같이 상대적으로 색상을 검출할 경우에는 주변 환경과 조도의 변화에 따라 평균값과 표준편차의 값이 달라지기 때문에 검출되는 영역 또한 주변 환경과 조도에 따라 달라진다. 따라서 주변 환경과 조도의 변화에 영향을 받지 않고 차선을 검출할 수 있다.

### 2.3 차선원소 및 클러스터 형성

차선을 검출하기 전에 픽셀 단위의 영상을 차선원소 단위로 전환한다. 한 차선원소는 포함된 픽셀의 색상에 따라 노란색, 흰색, 무색의 3가지 색상정보를 가진다. 한 차선원소에 노란색 픽셀의 포함률이 일정한 수준 이상인

경우 해당 차선원소는 노란색 색상정보를 가진다. 즉, 포함률을  $\lambda_Y$ , 검출된 노란색 픽셀의 개수를  $n_Y$ 라고 할 때, 한 차선원소에는  $n^2$ 개의 픽셀을 가지고 있으므로 식 (3)을 만족하면 해당 차선원소는 노란색 색상정보를 가진다.

$$n_Y \geq n^2 \times \lambda_Y \quad (3)$$

차선원소가 흰색의 색상정보를 가지는 것 또한 노란색 색상정보를 가지는 것과 유사하다. 포함률을  $\lambda_W$ , 검출된 흰색 픽셀의 개수를  $n_W$ 라 하면, 식 (4)를 만족할 때 해당 차선원소는 흰색 색상정보를 가지게 된다.

$$n_W \geq n^2 \times \lambda_W \quad (4)$$

식 (3), (4)의 조건에 만족하지 않는 차선원소는 무색 색상정보를 가진다.  $\lambda_Y$ 와  $\lambda_W$  매개 변수의 값을 정할 때 값이 작으면 색상정보를 갖는 차선원소의 개수가 증가하지만 잡음이 많아지고, 값이 작으면 검출되는 차선원소의 수가 적어 정확한 차선을 검출하지 못하게 되는 것을 고려한다.

위에서 형성된 차선원소를 이용하여 클러스터를 형성한다. 클러스터를 형성하는 방법은 색상정보가 있는 차선원소를 찾았을 때 인접한 동일한 색의 차선원소를 그래프와 트리 등의 검색에 사용하는 깊이 우선 탐색을 사용하여 찾아낸 뒤, 찾아낸 차선원소의 집합을 하나의 클러스터로 형성한다. 이때 인접한 차선원소는 검출된 차선원소 중심의  $x, y$  좌표를  $x_0, y_0$  찾고자 하는 차선원소 중심의  $x, y$  좌표를  $x_1, y_1$ 라고 할 때, 식 (5)의 두 식 중 하나를 만족하면 인접한 차선원소라 한다.

$$\begin{aligned} |x_0 - x_1| &= 1 & y_0 &= y_1 \\ |y_0 - y_1| &= 1 & x_0 &= x_1 \end{aligned} \quad (5)$$

클러스터를 형성한 차선원소는 무색의 차선원소로 색상정보가 변환된다.

검출된 클러스터 중 크기가 작은 클러스터는 잡음이라고 생각할 수 있으므로 영상 내에서 크기가 큰 상위 5개의 클러스터만을 다루도록 한다. 그림 6은 카메라를 통해 받아들인 영상에서 픽셀을 검출하고 차선원소와 클러스터를 형성하는 과정이다.

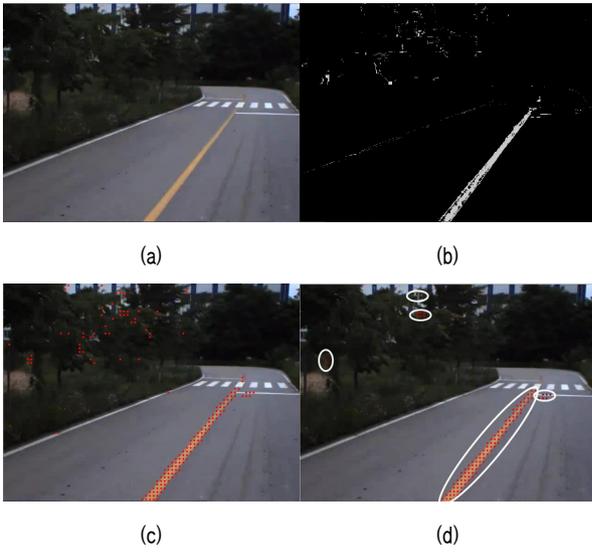


그림 6. 입력받은 영상(a)에서 노란색 픽셀을 검출(b)하고 그로부터 차선원소를 형성(c)하고 크기가 큰 5개의 클러스터를 형성(d).

### 2.4 차선 검출

검출된 클러스터 중 차선을 형성하는 직선을 알아내기 위한 처리 과정을 거쳐 적절한 클러스터를 선택한다.

그 과정의 처음 단계로 차선은 직선이기 때문에 식 (6)의 최소자승법을 사용하여 직선의 방정식을 만든다.

$$\epsilon = \sum_{i=1}^n (y_i - ax_i - b)^2 \quad (6)$$

그림 6(d)에 나타난 클러스터에 최소자승법을 사용하면 그림 7(a)와 같이 각 클러스터마다 흰색 직선이 생성되는데 이 때 생성된 직선의 기울기를  $\alpha$ 라 한다. 클러스터가 직선의 모양을 가지고 있는지 확인을 하기 위해 노란색 중앙 차선 클러스터일 경우 클러스터의 가장 오른쪽과 왼쪽에 있는 픽셀의 집합에 각각 최소자승법을 사용하여 클러스터에 추가로 2개의 직선을 생성하면 그림 7(b), (c), (d), (e), (f)와 같이 검은색 직선들이 생성된다. 가장 왼쪽과 오른쪽 픽셀의 집합에 의해 생성된 직선의 기울기를 각각  $\beta_L$ ,  $\beta_R$ 이라 할 때 식 (7)을 만족할 경우 클러스터를 제대로 된 직선이라 판단한다.

$$\begin{cases} |\alpha - \beta_R| < k_{ang} \\ |\alpha - \beta_L| < k_{ang} \end{cases} \quad (7)$$

$k_{ang}$ 는 각도에 대한 매개 변수로 값이 크면 직선이 아

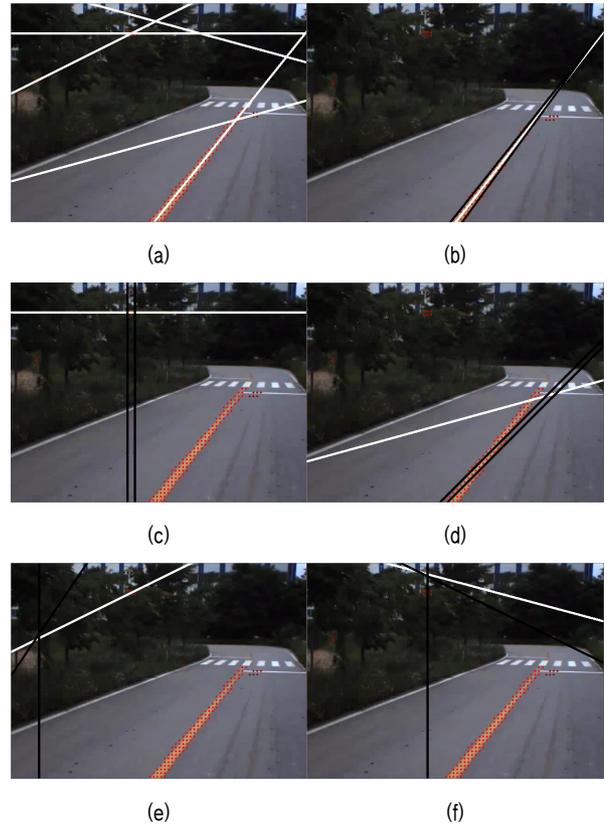


그림 7. 검출된 클러스터들이 차선을 정확히 나타내는지 흰색과 검은색 직선으로 표현한 영상.

닌 클러스터도 직선으로 판단할 수 있고, 작으면 잡음에 의해 클러스터의 모양이 조금만 왜곡이 되어도 클러스터의 모양이 직선이 아니라고 판단하는 특성을 고려해 값을 정한다.

위와 같은 과정을 거치면 그림 7에서 (b)는 식(7)을 만족하는 클러스터라는 것을 알 수 있으며 (c), (d), (e), (f)는 식(7)을 만족하지 못하는 것을 확인할 수 있다.

흰색 정지선도 노란색 중앙 차선의 직선 확인과 비슷하다. 클러스터의 가장 위쪽과 아래쪽에 있는 픽셀의 집합에 각각 최소자승법을 사용하여 직선의 기울기를  $\beta_T$ ,  $\beta_B$ 라 하면 식 (8)을 만족할 경우 클러스터를 직선이라 판단한다.

$$\begin{cases} |\alpha - \beta_T| < k_{ang} \\ |\alpha - \beta_B| < k_{ang} \end{cases} \quad (8)$$

#### 2.4.1 노란색 중앙 차선 검출

노란색 중앙 차선을 검출하는 알고리즘은 최초 직선 차선을 검출한다. 직선 모양의 클러스터 중 가장 큰 클러

스터를 노란색 중앙 차선으로 검출하고 검출된 차선이 검증된 노란색 중앙 차선인지 잡음인지를 판단하기 위해 다음과 같은 방법을 사용한다.

최소자승법을 사용하여 직선이 생성되었을 때 영상의 가장 아래쪽 직선의 좌표를 기준점이라 하자. 이전 영상에서 구한 기준점의  $x, y$  좌표를 각각  $x_{bp0}, y_{bp0}$  직선의 각도를  $\theta_0$ ,  $y$ 절편 값을  $y_{i0}$ 라 하고, 현재 영상에서 구한 BP의  $x, y$  좌표를 각각  $x_{bp}, y_{bp}$  직선의 각도를  $\theta$ , 직선의  $y$ 절편 값을  $y_i$ 라 할 때, 식 (9)를 만족하면 연속적으로 차선을 검출했다고 판단한다.

$$\begin{cases} |x_{bp} - x_{bp0}| < l_{dist} \\ |y_i - y_{i0}| < l_{dist} \\ |\theta - \theta_0| < l_{ang} \end{cases} \quad (9)$$

이때  $l_{dist}$ 는 거리에 관한 매개 변수이고  $l_{ang}$ 는 각도에 관한 매개 변수이다. 각 매개 변수의 값을 정할 때, 값이 큰 경우 잡음을 차선으로 검출할 수 있고, 값이 작은 경우에는 잡음에 의해 차선이 조금만 왜곡되어도 검출하지 못하는 특성을 고려한다.

연속적으로  $p$ 번 이상 차선을 검출한다면 검증된 노란색 직선 차선을 검출했다고 판단한다. 기존의 위치와 다른 곳에서 차선이 검출되었다면 현재 영상에서 검출된 직선을 기준으로  $p$ 번 이상 직선을 검출을 했을 경우 검증된 노란색 직선 차선을 검출했다고 판단한다. 검증된 노란색 직선 차선을 검출한 이후에는 상위 5개의 클러스터 중 하나의 클러스터가 식 (9)를 만족한다면 차선 검출에 성공했다고 판단한다. 연속적으로  $q-1$ 번 동안 노란색 직선 차선을 검출하지 못했다 하더라도  $q$ 번째에 노란색 직선 차선을 검출한 경우에는 계속 노란색 직선 차선을 검출했다고 판단한다.  $q$ 번 이상 노란색 직선 차선을 검출하는 것을 실패한다면, 검증된 노란색 직선 차선을 검출했다는 설정을 해제하고 다시 검증된 노란색 직선 차선을 찾는 알고리즘을 적용한다. 매개 변수  $p$ 의 값을 정할 때,  $p$ 가 큰 경우 정확하게 검증된 차선을 구할 수 있지만 잡음이 많은 영상에서는 검증된 차선을 구하기까지 오랜 시간이 걸리고,  $p$ 가 작은 경우 연속적으로 검출된 잡음을 검증된 차선이라고 오판할 수 있는 것을 고려한다. 매개 변수  $q$ 의 값이 크면 검증된 차선을 해제하는데 오랜 시간이 걸리고, 작을 경우 검증된 차선을 금방 해제하는 특성을 고려하여 값을 결정한다.

노란색 곡선 차선을 검출하는 알고리즘은 검증된 노란

색 직선 차선을 찾은 이후에 작동한다. 알고리즘이 노란색 직선 차선 검출을 실패했을 경우, BP의 차가  $x, y$  축에 각각  $m_{dist}$ 이내인 곳에 있는 클러스터를 노란색 곡선 차선을 이루는 클러스터라고 판단을 하고 클러스터의 가장 우측의 차선원소 집합을 곡선이라 판단한다. 매개 변수  $m_{dist}$ 의 값이 크면 잡음을 곡선으로 인식할 수 있고, 값이 작으면 잡음에 의해 왜곡된 곡선을 검출할 수 없다는 특징을 고려하여  $m_{dist}$ 의 값을 정한다.

#### 2.4.2 흰색 정지선 검출

화각이 커 전반적인 도로의 모습에 대한 영상을 얻을 수 있는 카메라에서 영상에서 나오는 정지선의  $x$ 축에 대한 길이는  $y$ 좌표의 값이 커질수록 작게 나타나고  $y$ 좌표의 값이 작아질수록 크게 나타나는데 이러한 특성을 이용해서 클러스터의  $x$ 축에 대한 길이를  $y$ 좌표에 대해 일정한 비율의 기준을 정하고 정지선을 검출한다.

화각이 좁아 도로와 정지선의 일부에 대한 영상을 얻을 수 있는 카메라에서 흰색 정지선은 영상의 왼쪽 끝과 오른쪽 끝에 붙어 나온다. 또한 차선이 가로와 평행하기 때문에 검출된 클러스터는 직사각형 모양이라는 특성을 이용하여 직사각형 안에서 검출된 흰색 픽셀의 비율이 직사각형 넓이의  $\lambda\%$  이상이 되었을 때 흰색 정지선을 검출한다.  $\lambda\%$ 의 값을 정할 때 큰 경우 정확하게 정지선을 검출할 수 있지만 차선원소의 검출률이 좋지 못할 경우 정지선을 검출 할 수 없고, 작은 경우 정지선의 검출률은 좋아이지만 잡음을 정지선으로 검출하는 특성을 고려한다.

### 3. 실험

#### 3.1 실험 개요

실험에 사용된 무인 자동차를 구성하기 위해 기아 자동차에서 생산된 쏘울 차량을 이용하였으며 항법과 센서, 그리고 카메라를 위해 3대의 컴퓨터, 차량 제어 유닛과 위성항법장치 등을 장착하였다. 차선을 검출할 때 1대의 카메라만을 사용하여 노란색 중앙 차선과 흰색 정지선을 검출 할 경우 화각이 적은 카메라는 커브를 돌 때 노란색 중앙 차선을 놓치는 경우가 발생했고, 화각이 넓은 카메라는 영상의 정확도가 떨어져 정확하게 차선과 정지선을 검출하는데 어려움이 있었다.

그런 이유로 그림 8(a)와 같이 차량 하단에는 근거리의 흰색 정지선을 검출하는 용도로 1대의 카메라를, 그림 8(b)와 같이 차량의 상단에는 노란색 중앙 차선과 흰색

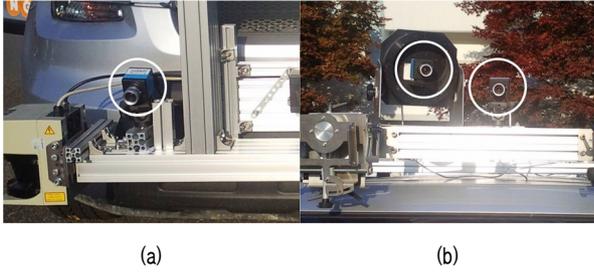


그림 8. 차량에 부착된 카메라의 위치.

정지선을 검출하는 용도로 2대의 카메라를 장착하였다. 카메라는 640×480 해상도, 60fps의 전송률, 1/4" format, 글로벌 셔터 방식을 사용하여 빠른 처리속도를 보이고 차량이 빠른 속도로 움직일 때에도 영상의 왜곡이 발생하지 않도록 하였다. 얻어진 영상을 처리하기 위해서 무인 자동차에 인텔 i5 750 2.67 GHz CPU, 마이크로소프트 윈도우 XP, DDR2 4GB RAM, 그리고 NVIDIA geforce GTS 250 그래픽카드를 가진 시스템을 탑재했다. 이 시스템에 이 논문에서 제안된 알고리즘을 적용하여 직선 구간과 곡선 구간에서 노란색 중앙 차선을 검출하면서 주행을 하고, 흰색 정지선을 검출하면 정지하고 다시 주행을 하는 실험을 실시하였다. 실험을 한 도로는 왕복 2차선의 도로로 총 1km의 거리와 한 번의 곡률이 큰 좌회전 구간으로 이루어져 있으며 도로상에는 두 개의 과속 방지턱과 한 개의 흰색 정지선이 있다.

### 3.2 매개 변수 설정

$n$ 은 차선원소의 크기를 결정,  $\lambda_Y$ 와  $\lambda_W$ 는 차선원소의 색상정보를 결정,  $k_{ang}$ 는 클러스터의 직선 모양을 판단,  $l_{ang}$ 는 이전 영상과 현재 영상에서 구한 차선의 유사도를 판단,  $m_{dist}$ 는 곡선의 검출,  $p$ 는 검증된 차선을 결정,  $q$ 는 검증된 차선의 설정을 해제,  $\lambda_{\%}$ 는 하단 카메라에서 정지선을 검출할 때 사용하는 매개 변수다.

실험적으로 확인하여서 다양한 환경에서 가장 좋은 성능을 나타내는 표 1과 같은 매개 변수를 선택하였고 정해진 값이 아니라 주변 환경에 따라 색상을 적응적으로 검출하는 특성을 가지고 있기 때문에 실제 환경에서도 조

표 1. 무인 자동차에서 사용된 매개 변수

$n$	4	$l_{ang}$	5°
$\lambda_Y$	0.125	$m_{dist}$	10
$\lambda_W$	0.125	$p$	20
$k_{ang}$	5°	$q$	10
$l_{dist}$	12.5	$\lambda_{\%}$	0.85

도가 변하는 것에 큰 영향을 받지 않고 좋은 검출률을 보여주었다.

### 3.3 실험 결과

무인 자동차가 실험 도로를 35km/h의 속도로 주행을 하면서 경로를 이탈하지 않고 완주하였고, 흰색 정지선이 있는 구간에서 잠시 정차하였다가 출발하였다. 노란색 중앙 차선의 검출 속도는 약 65fps의 처리속도를 보여주었다. 단순하게 영상에 허프 변환<sup>[11]</sup>을 통해 직선차선을 검출하는 것과 비교하여 약 2.1배 높은 처리속도를 보였다. 흰색 정지선의 검출 속도는 동일한 컴퓨터에서 태양광의 영향을 제거해주는 알고리즘을 사용하지 않았을 때에는 약 35fps의 처리속도를 보여주고 태양광의 영향을 제거해주는 알고리즘을 사용할 경우 약 16fps의 처리속도를 보여주었다.

그림 9는 상단 카메라와 하단 카메라에서 흰색 정지선을 검출한 영상, 그림 10은 무인 자동차가 주행을 하면서

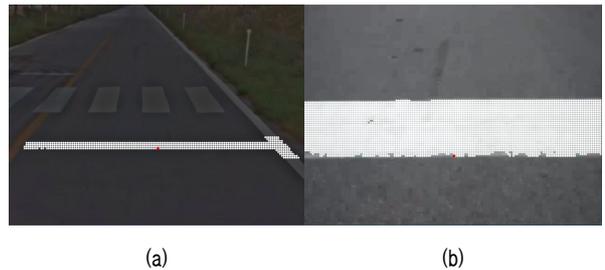


그림 9. 상단 카메라 (a)와 하단 카메라 (b)에서 검출된 정지선.

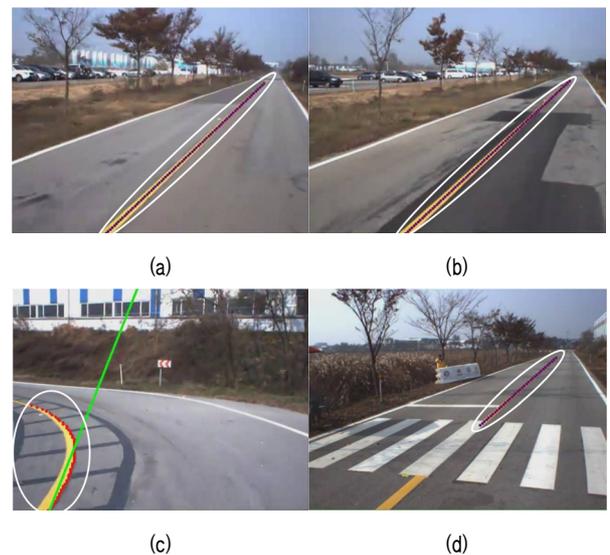


그림 10. 직선 주행(a) (b), 곡선 주행(c), 정지선에 정지해 있는 상태(d)에서 차선을 검출하는 영상.

노란색 중앙 차선과 흰색 정지선을 검출한 영상이다. 여러 번의 실험 결과 그림 11과 같이 어둡고 밝은 영상에서 조도의 변화에 관계없이 알고리즘이 동작한다.



그림 11. 다른 조도에서 검출된 동일한 도로 위의 차선 인식.

#### 4. 결론

무인 자동차에 노란색 중앙 차선과 흰색 정지선을 검출하는 알고리즘을 탑재하여 잡음이 심한 실제 야외 환경에서 주행을 한 실험결과 조도의 변화에 관계없이 약 35km/h로 속도로 실시간 제어 주행이 가능함을 보였다.

차선의 검출률과 정확성을 높이기 위해서는 도로를 검출하는 알고리즘<sup>[10]</sup>을 사용하여 영상 처리를 도로상에서 실시간으로 할 수 있다. 그리고 제안된 알고리즘에서는 현재 영상에서의 형태 정보와 색상 정보를 이용함과 동시에 기존의 누적된 영상정보를 이용하여 차선이 지워지거나 횡단보도가 나타난 후에도 차선을 강인하게 검출하는 것을 확인하였다.

#### 참고문헌

- [1] C. Crane, D. Armstrong, R. Touchton, T. Galluzzo, S. Solanki, J. Lee, D. Kent, M. Ahmed, R. Montane, S. Ridgeway, S. Velat, G. Garcia, M. Griffis, S. Gray and J. Washburn, "Team CIMAR's NaviGATOR: An Unmanned Ground Vehicle for Application to the 2005 DARPA Grand Challenge," *Journal of Field Robotics*, Vol.23, No.8, pp.599-623, 2006.
- [2] G. Seetharaman, A. Lakhota and E. Blasch, "Unmanned vehicles come of age: The DARPA grand challenge," *Special Issue of IEEE Computer*, pp.32-35, Dec., 2006.
- [3] J. Yoon and C. D. Crane, "Evaluation of terrain using LADAR data in urban environment for autonomous vehicles and its application in the DARPA urban challenge," *ICCAS-SICE*, pp.641-646, Aug., 2009.
- [4] M. Xie, H. Chen, X. F. Zhang, X. Guo, Z and P. Yu, "Development of Navigation System for Autonomous Vehicle to Meet the DARPA Urban Grand Challenge," *IEEE Intelligent Transportation Systems Conference*, pp.767-772, Sept., 2007.
- [5] T. Y. Sun, S. J. Tsai and V. Chan, "HSI Color Model Based Lane-Marking Detection," *IEEE Intelligent Transportation Systems Conference*, pp.1168-1172, Sept., 2006.
- [6] C. Ma and M. Xie, "A Method for Lane Detection Based on Color Clustering," *Knowledge Discovery and Data Mining Third International Conference*, pp.200-203 Jan., 2010.
- [7] H. Cheng, B. Jeng, P. Tseng, K and C. Fan, "Lane Detection With Moving Vehicles in the Traffic Scenes," *IEEE Intelligent Transportation Systems*, Vol.7, No.4, pp.571-582, Dec., 2006.
- [8] G. D. Finlayson, S. D. Hordley and M. S. Drew, "Removing Shadows from Images," *the 7th European Conference on Computer Vision-Part*, Vol.IV, pp.823-836, May, 2002.
- [9] O. Chutatape and L. Guo, "A modified Hough transform for line detection and its performance," *Pattern Recognition*, Volume 32, Issue 2, pp.181-192, Feb., 1999.
- [10] Y. He, H. Wang and B. Zhang, "Color-based road detection in urban traffic scenes," *IEEE Intelligent Transportation Systems*, Vol.5, No.4, pp.309-318, Dec., 2004.



#### 한명희

2011 KAIST 전산학과(공학사)  
2011~현재 KAIST 전산학과 석사과정



### 이 건 흥

2009 건국대학교 컴퓨터공학  
부(공학사)

2009~현재 KAIST 로봇학제 석  
사과정



### 조 성 호

1999 서울대학교 기계항공공  
학부(공학사)

2001 MIT 기계공학과(공학석사)  
2006 MIT 전자전산학과(공학  
박사)

2006~2007 MIT 미디어랩 박사  
후연구원

2008~현재 KAIST 전산학과 교수

관심분야: 인간-로봇 상호작용, 바이오 로보틱스, 지  
능적인 시스템 설계