ISSN 1975-8359(Print) / ISSN 2287-4364(Online)
The Transactions of the Korean Institute of Electrical Engineers Vol. 65, No. 11, pp. 1879~1886, 2016
http://dx.doi.org/10.5370/KIEE.2016.65.11.1879

탑뷰 영상을 이용한 차선, 정지선 및 과속방지턱 인식

Recognition of Lanes, Stop Lines and Speed Bumps using Top-view Images

안 영 선* · 곽 성 우† · 양 정 민*†
(Young-Sun Ahn · Seong Woo Kwak · Jung-Min Yang)

Abstract - In this paper, we propose a real-time recognition algorithm of lanes, stop lines and speed bumps on roads for autonomous vehicles. First, we generate a top-view using the image transmitted from a camera that is installed to see the front of a vehicle. To speed up the processing, we simplify the mapping algorithm in constructing a top-view wherein the region of interest (ROI) is concerned. The features of lanes, stop lines and speed bumps, which are composed of lines, are searched in the edge image of the top-view, then followed by labeling and clustering specialized to detect straight lines. The width of lines, distances from the center of a vehicle, and curvature of each cluster are considered to select final candidates. We verify the proposed algorithm on real roads using the commercial car (KIA K7) which is converted into an autonomous vehicle.

Key Words : Autonomous vehicles, Image processing, Lane detection, Stop line detection, Speed bump detection, Top-view

1. 서 론

최근 전 세계 자동차 관련 업체들은 무인 자율주행 자동차를 상용화하기 위해 모든 역량을 집중하고 있다. 자율주행 기술은 1990년 미국 방위고등연구계획국(DARPA: Defence Advanced Research Projects Agency)이 군사적 목적으로 운용하기 위해 지상 차량에 대해서 처음 연구를 시작하였다. 이후 대학, 연구소, 일반인을 대상으로 2004년부터 2007년까지 총 4회의 경진대회를 개최하면서 자율주행 자동차에 대한 연구가 급속히 진전되었다[1],[2]. 인터넷 검색 업체인 구글(Google)이 개발한 구글 자율주행 자동차(Google self-driving car)를 시작으로 BMW, GM, 도요타, 닛산, 현대자동차 등 완성차 업체에서는 2020년까지 상용화를 목표로 적극적인 기술개발을 하고 있다.

아직도 매년 전 세계에서 120~130만 명이 교통사고로 사망한다. 중요한 점은 자동차 사고의 90% 이상이 차량자체의 결함보다는 운전자의 부주의에 의해서 발생한다는 사실이다[1]. 기술의 안정성이 검증되고 상용화되어 자율주행자동차 시대가 열린다면 운전자의 최소한의 개입으로 안전하게 목적지까지 이동할 수 있을 것이다.

현재 개발된 자율주행 자동차에는 값비싼 센서가 장착되어 주행환경을 인식한다. 그 중에서 비전 센서는 상대적으로 저렴하고 여러 가지 정보를 동시에 얻을 수 있다는 장점이 있다. 여러 정보에 대한 인식 및 판단 과정에서 많은 처리량이 필요하지만 최근 고속 컴퓨터의 개발로 이 문제는 점차 해결되고 있다[3]. 자율주행에서 차선과 정지선 등 도로 위의 환경을 정확하게 인식하는 일은 매우 중요하다. 또한 신속한 영상처리는 사고 위험에 대해 즉각 반응하여 위험으로부터 벗어날 수 있게 해준다.

차선 및 도로환경 인식을 위한 기법으로는 기하학적인 변환과 모폴로지(morphology)를 이용하는 방법, 허프 변환(Hough transform)[4]을 이용하는 방법, 히스토그램을 이용하는 방법, 스네이크(snake)를 이용하는 방법 등이 있다. 기존에 개발된 이 방법들은 영상의 넓은 영역을 탐색하고 있으며, 특징점을 찾기 위해 많은 전처리과정을 수행하여 처리 속도가 만족스럽지 않은 경우가 많다. 그리고 차선의 오염이나 주변 환경의 변화로 주요한 차선을 일부 놓쳐 잘못된 인식을 하거나 인식 자체를 하지 못하는 경우가 생긴다.

본 논문에서는 비전시스템의 장점을 극대화하기 위해 차선, 정지선 및 과속방지턱을 실시간으로 인지할 수 있도록 탑뷰(top-view)[5],[6]를 만드는 과정을 단순화하여 연산량을 줄이는 방법과, 최소한의 후보군을 찾아 실제 차선과 가장 유사한 곡선을 가진 후보를 선택하는 방법을 제시한다. 또한 차선인식 후 차선 내에 존재하는 정지선과 과속방지턱 인식 방법을 제안한다. 참고로 본 논문에서 제안된 방식은 산업통상자원부 주최 2015년 “자율주행 자동차 콘테스트”[7]에 참가한 무인자율주행 자동차에 실제 탑재되었다.

2. 픽셀 좌표와 월드 좌표의 매칭 및 탑뷰 이미지 생성

비전 센서로부터 전달된 이미지는 2차원 공간이며 3차원인 월

† Corresponding Author : Dept. of Electronic Engineering, Keimyung University, Korea.
E-mail: ksw@kmu.ac.kr
* Dept. of Electronic Engineering, Keimyung University, Korea.
†* Corresponding Author: School of Electronics Engineering, Kyungpook National University, Korea.
E-mail: jmyang@knu.ac.kr
Received : June 14, 2016; Accepted : September 30, 2016

드 좌표 공간을 원근감 있게 투사한다. 핀홀(pin hole) 카메라 모델에서 이러한 변환 관계는 수식 (1)과 같이 표현된다.

$$s\begin{bmatrix}x\\y\\1\end{bmatrix}=\begin{bmatrix}f_x & cf_x & c_x\\0 & f_y & c_y\\0 & 0 & 0\end{bmatrix}\begin{bmatrix}r_{11} & r_{12} & r_{13} & t_1\\r_{21} & r_{22} & r_{23} & t_2\\r_{31} & r_{32} & r_{33} & t_3\end{bmatrix}\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix}=A[R|t]\begin{bmatrix}X\\Y\\Z\\1\end{bmatrix} \quad (1)$$

여기서 X, Y, Z는 월드 좌표계 상의 세 점이고, x, y는 2차원(2D) 이미지의 픽셀 좌표, [R|t]는 카메라 외부 파라미터(extrinsic parameter), A는 내부 파라미터(intrinsic parameter)이다. 내부 파라미터는 초점거리(f_x, f_y), 주점(c_x, c_y), 비대칭계수(cf_x)로 카메라 자체의 특성을 의미한다. 외부 파라미터는 카메라가 설치되어 있는 위치나 기울어짐에 대한 월드 좌표계와 카메라 좌표계의 회전변환을 의미한다. s는 카메라 광학렌즈의 배율을 나타낸다. 그림 1은 핀홀 카메라 모델에서 3D 공간과 2D 이미지 평면사이의 관계를 보여준다.

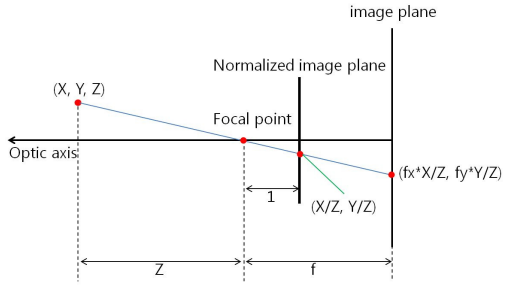


그림 1 3D와 2D의 좌표관계.
Fig. 1 Relation of 3D and 2D coordinates.

3D 공간에서의 한 점(X, Y, Z)가 초점(focal point)을 지나 초점거리가 1인 가상의 이미지 평면(normalized image plane)에 투사되는 2D 좌표는 (X/Z, Y/Z)이다. 여기서 초점거리를 곱하면 실제 이미지 픽셀 좌표를 계산할 수 있다. 그리고 실제 이미지 평면의 중심은 주점(c_x, c_y)와 광학렌즈의 배율(s)을 고려하면 수식 (2)와 같이 도출할 수 있다.

$$\begin{aligned}x&=(f_x\times X/Z\times s)+c_x\\y&=(f_y\times Y/Z\times s)+c_y\end{aligned} \quad (2)$$

카메라의 외부 파라미터는 월드 좌표 상의 카메라의 위치(X_0,Y_0,Z_0)와 카메라 좌표계와 월드 좌표의 기울어짐 차이인 팬(Pan : P), 틸트(Tilt : T), 롤(Roll : R)로 구성된다. 그림 2는 카메라 좌표계 (X_c, Y_c, Z_c)와 월드 좌표계 (X, Y, Z)와의 상관관계이다. 세 축에 대한 회전 변환 행렬 수식 (3)-(5)를 모두 곱하면 카메라의 전체 회전각에 대한 행렬식을 만들 수 있다. 회전 변환 행렬(R_{mat})과 카메라의 위치 (X_0, Y_0, Z_0)를 고려하여 외부 파라미터를 적용한 카메라 좌표 (X_c, Y_c, Z_c)를 도출한다.

팬은 카메라의 좌우 회전각이고, 틸트는 상하 회전각, 롤은 월드 좌표계의 X축과 카메라 좌표계의 X_c축 사이의 회전각을 나타

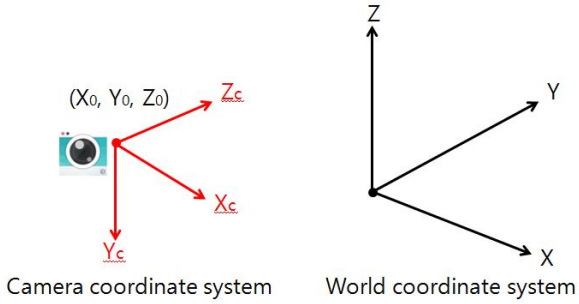
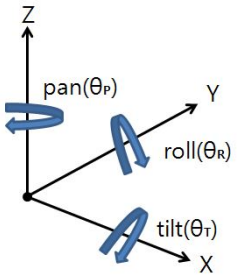


그림 2 카메라와 월드 좌표계의 상관관계.
Fig. 2 Relation of the camera coordinate and world coordinate system.

낸다.



$$R_z(\theta_P)=\begin{bmatrix}\cos\theta_P & -\sin\theta_P & 0\\\sin\theta_P & \cos\theta_P & 0\\0 & 0 & 1\end{bmatrix} \quad (3)$$

$$R_y(\theta_R)=\begin{bmatrix}\cos\theta_R & 0 & \sin\theta_R\\0 & 1 & 0\\-\sin\theta_R & 0 & \cos\theta_R\end{bmatrix} \quad (4)$$

$$R_x(\theta_T)=\begin{bmatrix}1 & 0 & 0\\0 & \cos\theta_T & -\sin\theta_T\\0 & \sin\theta_T & \cos\theta_T\end{bmatrix} \quad (5)$$

$$\begin{aligned}R_{mat}&=\begin{bmatrix}\cos(P) & 0 & \sin(P)\\0 & 1 & 0\\-\sin(P) & 0 & \cos(P)\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\0 & \cos(T) & -\sin(T)\\0 & \sin(T) & \cos(T)\end{bmatrix}\begin{bmatrix}\cos(R) & -\sin(R) & 0\\\sin(R) & \cos(R) & 0\\0 & 0 & 1\end{bmatrix}\\&=\begin{bmatrix}\cos(P) & -\sin(P)\sin(T) & -\sin(P)\cos(T)\\\sin(P) & \cos(P)\sin(T) & \cos(P)\cos(T)\\0 & -\cos(T) & \sin(T)\end{bmatrix}\begin{bmatrix}\cos(R) & -\sin(R) & 0\\\sin(R) & \cos(R) & 0\\0 & 0 & 1\end{bmatrix}\\&=\begin{bmatrix}\cos(P)\cos(R)-\sin(P)\sin(T)\sin(R) & -\cos(P)\sin(R)-\sin(P)\sin(T)\cos(R) & -\sin(P)\cos(T)\\\sin(P)\cos(R)+\cos(P)\sin(T)\sin(R) & -\sin(P)\sin(R)+\cos(P)\sin(T)\cos(R) & \cos(P)\cos(T)\\-\cos(T)\sin(R) & -\cos(T)\cos(R) & \sin(T)\end{bmatrix}\end{aligned} \quad (6)$$

수식 (6)의 R_{mat}과 카메라의 위치를 고려한 월드 좌표 ($X-X_0$, $Y-Y_0$, $Z-Z_0$)를 곱하여 회전 변환된 카메라 좌표 (X_c, Y_c, Z_c)를 식 (7)과 같이 계산한다. 이를 수식 (2)에 대입하면 해당하는 월드 좌표에 대한 이미지 픽셀 좌표 (x,y)를 알 수 있다. 픽셀 좌표는 수식 (8)과 같이 주로 (u, v)로 표현하며 수식 (2)의 (x, y)와 동일하다.

$$\begin{bmatrix}X_c\\Y_c\\Z_c\end{bmatrix}=R_{mat}\begin{bmatrix}X-X_0\\Y-Y_0\\Z-Z_0\end{bmatrix} \quad (7)$$

$$\begin{aligned}u&=(f_x\times X_c/Z_c\times s)+c_x\\v&=(f_y\times Y_c/Z_c\times s)+c_y\end{aligned} \quad (8)$$

본 논문에서는 정면을 바라보도록 설치된 카메라에서 전송되

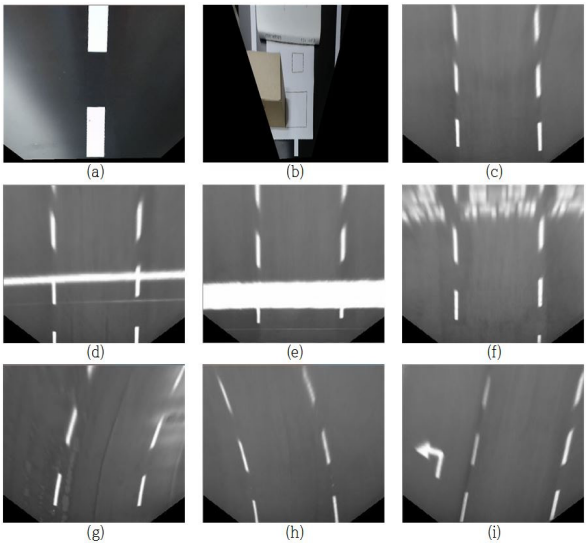


그림 3 역방향 맵핑을 이용한 탑뷰 이미지. (a)~(b): 모형 도로, (c)~(i): 실제 도로.

Fig. 3 Top-view image using backward mapping. (a)~(b): road model, (c)~(i): real road.

는 이미지를 위에서 내려다 본 탑뷰 이미지로 만들어 사용한다. 본 논문에서 사용할 탑뷰 이미지는 다음 두 가지를 만족하도록 하였다. 본 논문에서 제안하는 다음의 방식은 탑뷰를 생성하는 시간을 단축시켜 실시간 처리를 용이하게 한다.

첫째, 카메라 이미지의 가로 및 세로 한 픽셀에 대한 월드 좌표계에서의 거리 값은 같도록 한다. 둘째, 도로 환경 인지에 필요한 최소한의 영역만을 탑뷰로 변환하여 사용한다. 첫 번째 조건은 수직으로 하늘에서 바라본 이미지를 정확하게 만들기 위해서이다. 두 번째 조건은 원본이미지에서 관심영역(ROI)만을 탑뷰로 만들어 연산 속도를 올리기 위함이다. 탑뷰는 역방향 매핑(backward mapping)기법을 활용하여 생성한다. 아래 수식 (9)에 제시된 바와 같이 탑뷰 영상(Dst)의 좌표와 일대일 매칭되는 원본 영상(Src)의 좌표를 계산하여 해당하는 원본 영상의 밝기 값을 탑뷰 영상에 복사한다. 이것은 탑뷰 영상을 만드는 기존 와핑(warping)방식과 비교했을 때 생성 과정이 크게 단순화되었다는 장점이 있다. 단순화된 과정은 빠르게 탑뷰를 얻을 수 있게 한다. 여기서 (n,m)은 탑뷰 영상에서의 픽셀 좌표를 의미하고, $(u_{(n,m)}, v_{(n,m)})$은 카메라로부터 들어온 원본 영상에서의 탑뷰 영상 픽셀 (n,m)에 대한 좌표이다. width, height는 탑뷰 영상의 가로 및 세로 크기를 나타낸다.

$$
\begin{aligned}
u_{(n,m)} &= f_x \times (X_c \times n)/Z_c \\
v_{(n,m)} &= f_y \times (Y_c \times m)/Z_c \\
Dst(n,m) &= Src(u_{(n,m)}, v_{(n,m)}) \\
0 \le n < width, &\; 0 \le m < height
\end{aligned} \tag{9}
$$

역방향 매핑에서는 탑뷰 이미지의 밝기 값이 갑자기 변화하는 계단현상(jagging)이 발생한다. 이는 탑뷰 이미지가 원본 이미지에서 해당되는 거리에 대한 밝기 값을 건너뛰며 가져오기 때문이다. 선형 보간법(linear interpolation), 양선형 보간법(bilinear interpolation) 등 다양한 방법의 보간법을 사용하여 계단현상을 해결할 수 있다. 본 연구에서는 실시간 처리를 고려하여 가장 빠른 속도로 처리할 수 있는 선형 보간법을 가로 방향 1차원으로 수행한다. 아래의 그림 3은 실내 모형 도로와 실제 차량에서 촬영된 이미지에 대하여 역방향 매핑으로 만들어진 탑뷰 이미지를 보여준다.

3. 전처리

3.1 그레이 스케일(Gray Scale)

카메라를 통해 입력되는 영상데이터는 3채널 R(red), G(green), B(blue)로 구성된다. 차선이나 정지선, 과속방지턱 등 사물의 윤곽을 표현하기 위해 세 채널의 밝기를 평균한 1 채널 그레이 스케일로 변환한다.

3.2 스무딩(Smoothing)

이미지에서 잡음은 사물을 인식하는 데 어려움을 준다. 가우시안(Gaussian) 스무딩은 2차원 가우시안 분포를 가진 마스크[8]를 원본영상과 컨벌루션(convolution)하여 이미지의 고주파 잡음을 제거하는 작업이다. 이 필터를 사용하면 에지(edge)가 무뎌지는 단점이 있지만 이미지에 섞여있는 가우시안(Gaussian) 잡음을 제거하는 데 효과적이다.

3.3 소벨 연산(Sobel Operator)

사물의 윤곽을 검출하기 위해 에지 추출 연산자를 사용한다.

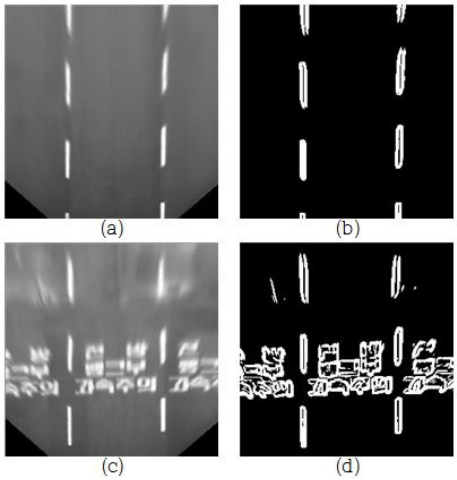


그림 4 소벨 연산 이미지. (a) 입력영상1, (b) 에지영상1, (c) 입력 영상2, (d) 에지 영상2.

Fig. 4 Image after Sobel operation: (a) Input image1, (b) Edge image1, (c) Input image2, and (d) Edge image2.

본 연구에서 사용한 1차 미분연산자인 소벨 마스크(Sobel mask)는 모든 방향의 에지를 검출할 수 있고 대각방향 에지에 민감하며, 잡음에 대체적으로 강한 특징을 갖고 있기 때문에 차선과 같은 도로 위의 표시를 찾는 데 유리하다. 수평과 수직 소벨 마스크[8]를 각각 입력 이미지와 컨벌루션(convolution)한다. 이때 만들어진 두 개의 결과를 합쳐 소벨 에지 이미지를 만든다. 이미지를 합칠 때 각각의 이미지의 밝기는 1/2로 줄인 뒤 서로 더한다. 밝기를 반으로 줄이지 않고 합치게 될 경우 이미지의 최대 밝기 값을 넘게 되거나 밝기 변화가 작은 부분도 부각될 수 있기 때문이다. 그림 4는 도로 차선 이미지에 대하여 소벨 연산자를 사용하여 영상의 에지를 추출한 것이다. 왼쪽은 입력영상이고 오른쪽은 소벨 연산 후 영상을 보여준다.

4. 차선 및 도로환경 인지 알고리듬

본 논문에서는 도로환경을 정확하고 빠르게 인지하기 위하여 탑뷰 기반 에지 이미지에서 차선과 유사한 데이터만을 추출하여 최소한의 후보군을 만든 후 근접한 후보군을 각각의 레이블(label)로 클러스터링(clustering)한다. 이러한 과정을 거친 클러스터들을 2차 곡선으로 방정식화 하여 최적의 차선을 인식한다. 그 후 차량이 주행하는 차선 안의 정지선이나 과속방지턱 등 노면표시를 인지한다.

4.1 차선 후보 검출

차선은 약 15 cm의 폭으로 노면에 그려져 있다. 에지 이미지를 일정한 임계값으로 이진화하면 하나의 차선에서 그림 5과 같이 두 개의 펄스가 만들어진다. 첫 번째 펄스와 두 번째 펄스의 사이 간격이 약 15 cm로 차선의 폭과 유사하다면 이것을 차선 후보로 저장한다. 이 방법으로 에지가 추출된 이미지 전체에서 가로 방향으로 차선과 유사한 폭을 가진 특징점들을 모두 저장하고 후보로 관리한다.

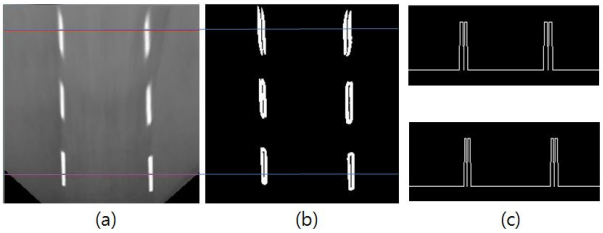


그림 5 차선에 의한 펄스. (a) 입력 영상, (b) 에지 영상, (c) 펄스 영상.

Fig. 5 Pulse of lane: (a) Input image, (b) Edge image, and (c)Pulse Image.

그림 6은 위의 방법을 이미지 전체에 수행하여 모든 차선 후보를 검출한 것이다. 그림의 왼쪽 영상은 에지 추출 영상이고 오른쪽 영상에 파란색으로 표현된 부분은 차선 후보를 나타낸다.

4.2 차선 후보 클러스터링(Clustering)

그림 6의 (a), (b)와 같이 차선의 후보는 서로 연결되어 있는 특징을 보인다. 이러한 관계를 가진 포인트를 각각의 묶음으로 표현하기 위해 다음과 같은 과정을 수행한다.

알고리듬 1: 클러스터링 알고리듬

(i) 모든 차선 후보에 대하여 레이블(label) "0"을 부여한다(그림 7(a)).

(ii) 영상의 가장 아래쪽에 있는 후보로부터 레이블 "1"을 부여한 뒤 픽셀 좌표의 위쪽에 있으며 픽셀 거리가 2보다 작은 거리에 있는 후보가 있으면 같은 레이블 번호를 부여한다(그림 7(b), (c)).

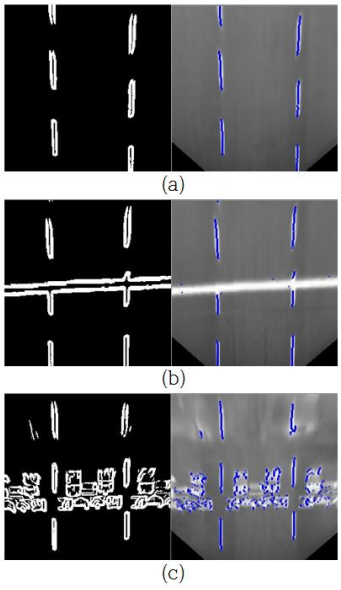


그림 6 차선 후보 검출

Fig. 6 Candidates of the lane

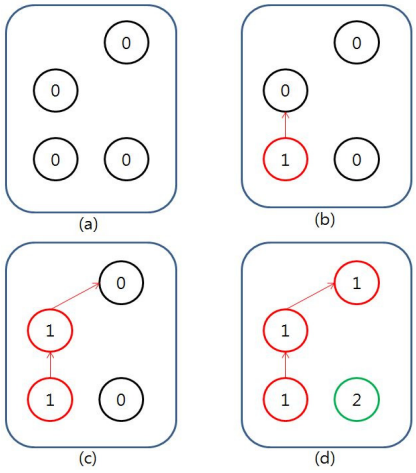


그림 7 클러스터링 과정.

Fig. 7 Clustering process.

(iii) 만약 픽셀 거리가 2보다 작은 거리에 후보가 없다면 레이블 번호를 하나 증가시키고 (ii)의 픽셀 조건을 만족하는 후보를 찾아 새로운 군집을 만든다(그림 7(d)).
(iv) 모든 후보의 레이블 번호가 "0"이 아닐 때까지 (iii)의 방법을 반복한다.
(v) 일정 개수 이하의 레이블 번호를 가진 묶음은 모두 삭제한다.
(vi) 최종적으로 남겨진 묶음을 차선 최종 후보군으로 관리한다.

일반적으로 사용되는 클러스터링 방법은 이미지 내의 픽셀 덩어리를 모두 찾아내지만 본 연구에서의 클러스터링은 가로 방향으로 이웃한 픽셀은 묶지 않기 때문에 차선일 가능성이 매우 큰 클러스터를 분류할 수 있는 장점이 있다. 그림 8은 차선 후보 픽셀(a)들에 대한 클러스터링 결과(b)이다. 각각 다른 색상으로 표현된 6개의 차선 후보 클러스터가 생성된 것을 볼 수 있다.

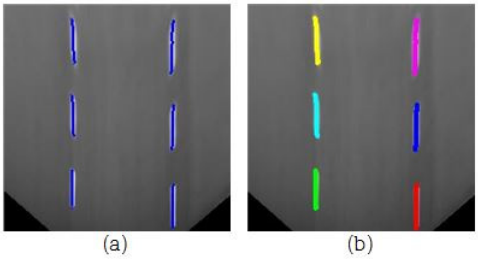


그림 8 클러스터링 결과. (a) 후보 포인터, (b) 클러스터.
Fig. 8 Results of clustering: (a) candidate points and (b) cluster.

위의 그림 8(b)와 같이 하나의 차선이지만 서로 떨어져 있거나 지워져 있는 차선으로 인해 서로 다른 클러스터로 나눠질 경우가 발생한다. 이 문제를 해결하기 위해 같은 성분의 클러스터를 서로 묶어주는 다음과 같은 알고리듬을 사용한다.

알고리듬 2: 클러스터 연결 알고리듬
(i) 그림 9의 (a)와 같이 하나의 차선 성분을 구성하는 서로 다른 클러스터는 서로 멀지 않고 각각의 연장선상에 있다. 그러므로 가장 아래쪽 클러스터를 기준으로 처음과 끝 두 점을 사용하여 직선방정식 y=ax+b를 만든다.
(ii) 자신보다 위쪽에 있는 클러스터의 시작점 세로 좌표를 (i)의 직선방정식에 대입하여 연장선상의 가로 예상 위치를 계산한다(그림 9(b)).
(iii) 예상 위치와 클러스터 초기점이 유사하다면 시작한 클러스터와 같은 레이블 번호를 부여하여 하나로 묶는다(그림 9(c)).

그림 9(a)의 6개의 클러스터들은 위 알고리듬에 의해 그림 9(d)와 같이 2개의 클러스터로 묶인다. 본 논문에서 제안된 클러스터링 알고리듬은 여러 클러스터들을 서로 관련된 하나의 클러스터로 묶기 때문에 가장 차선과 유사한 각각의 클러스터들로 나

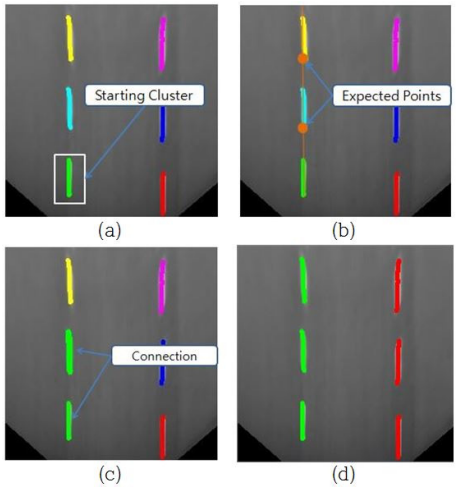


그림 9 클러스터 연결.
Fig. 9 Cluster connection.

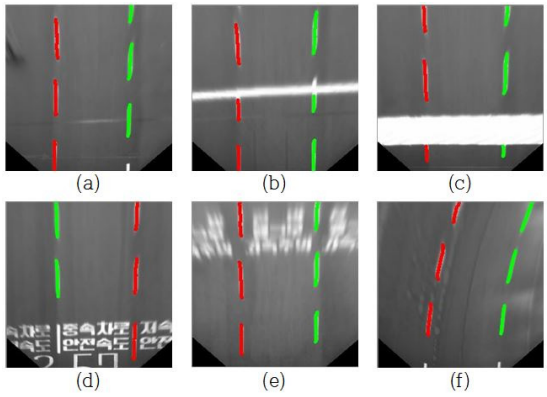


그림 10 실제 도로 영상에서 차선 클러스터 추출.
Fig. 10 Clusters of lane on real road images.

눌 수 있다. 노면 위의 글자나 차선 이외의 사물에서 생성된 클러스터는 잘못된 차선을 인식할 수 있게 한다. 이를 막기 위해서 최종적으로 묶여진 클러스터의 시작점과 끝점의 유클리디안 거리 (Euclidean Distance, $D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$)가 탑뷰 이미지에서 만들어질 수 있는 최대 클러스터의 80% 이상의 길이를 가진 클러스터만을 분류하여 잘못 인식할 가능성을 최대한 줄였다. 그림 10(a)~(f)는 실제도로에서 최종적으로 차선에 대한 클러스터를 추출한 결과이다. 각 영상에서 빨간색과 초록색의 2개의 차선이 검출되었음을 보여준다.

4.3 차선 인식(Lane Detection)

클러스터링 작업을 완료했을 때 그림 10과 같이 두 개의 클러스터가 발생하는 것이 가장 좋지만 그렇지 않은 경우가 많다. 두 개 이상의 차선 후보 클러스터가 있을 경우 차량의 중심과 가장 가깝고, 실제 차선 간격(2.7m~3.7m)과 유사하며 서로 비슷한 곡률을 가지는 한 쌍의 클러스터를 선택한다.

$$\begin{aligned} y_1 &= Ax_1^2 + Bx_1 + C \\ y_2 &= Ax_2^2 + Bx_2 + C \\ y_3 &= Ax_3^2 + Bx_3 + C \end{aligned} \tag{10}$$

곡률을 고려하기 위해 각 클러스터의 처음, 중간, 끝의 세 점 (x_1, y_1), (x_2, y_2), (x_3, y_3)을 수식 (10)에 대입하여 미지수 A, B, C를 구한다. 다른 클러스터의 미지수 값들과 비교하여 곡률이 유사하고, 간격이 실제 차선 간격과 비슷하다면 차선으로 선택한다. 한 쌍 이상의 차선이 선택된다면 시작점의 중심이 영상의 중심(차량의 중심)과 가장 가까운 차선을 선택한다. 이 과정을 거치면 차량이 현재 주행하고 있는 차선을 인식할 수 있다. 더불어 최종적으로 선택된 두 차선 중 오른쪽 차선의 색상을 분석하여 주황색 계열이라면 차량이 도로의 중앙선을 이탈했다고 판단할 수 있다.

4.4 정지선 및 과속방지턱 인식

차선을 정확히 인지하면 현재 주행 중인 차선 내부로 ROI를 한정시킨다. 이렇게 하면 차선 내에 존재하는 정지선 및 과속방지턱과 같은 노면표시를 인식하기 위한 연산량을 줄일 수 있다. 정지선과 과속방지턱은 보통 차선과 수직하며 정해진 폭으로 그려져 있기 때문에 그 특징을 활용하여 차선을 찾을 때와 유사한 방법으로 인식할 수 있다. 그림 11은 차선을 인식한 후 정지선과 과속방지턱을 인지하기 위한 ROI를 보여준다. 차선이 검출된 후 만들어진 ROI는 전체 이미지의 절반 이하로 줄어든다. 따라서 차선 인식 후 정지선 및 과속방지턱 인식 속도는 차선 인식 전에 비해 크게 증가한다.

일반적으로 정지선은 약 1 m의 폭으로 그려져 있고, 과속방지턱은 약 2 m의 폭으로 그려져 있다. 이러한 물리적인 특징을 이용하여 알고리듬 3과 같은 과정을 통해 정지선 및 과속 방지턱을 인지한다.

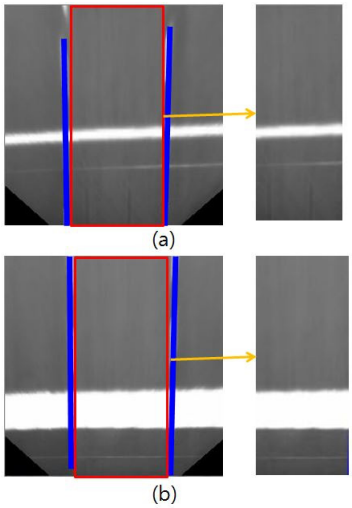


그림 11 정지선 및 과속방지턱을 위한 관심영역. (a) 정지선 ROI, (b) 과속방지턱 ROI.

Fig. 11 ROI for stop lines and speed bumps: (a) ROI of stop line and (b) ROI of speed bump.

알고리듬 3: 정지선 및 과속방지턱 인지 알고리듬

(i) 그림 11의 (a) 정지선 ROI와 (b) 과속방지턱 ROI에서 차선을 찾을 때와 같이 그레이 스케일, 스무딩, 소벨 연산을 수행하여 전처리를 진행한다.

(ii) 전처리된 이미지에서 가로 방향이 아닌 세로 방향으로 정지선과 과속방지턱의 물리적 특징과 같은 펄스를 찾아 그 중심점을 후보 포인트로 각각 관리한다.

(iii) 알고리듬 1의 클러스터링 알고리즘을 통해 수평방향 클러스터링을 실시하고 ROI의 이미지 가로폭의 90 % 이상의 클러스터를 정지선 및 과속방지턱으로 인지한다.

5. 실험 결과

본 연구에서는 다음과 같은 환경 하에서 실험을 수행하였다.

- 카메라 : (주)비전에스티 Real-Time Stereo Camera (VSTC-V200GLRD)
- 차량 : 기아 K7
- PC : Intel core i5-2500(3.3Ghz), 8G RAM
- Software Tool : Visual C++, MFC, OpenCV

시험용 차량에 스테레오 카메라를 설치하여 ITS 지능형 자동차부품 주행시험장[10]에서 자체적으로 취득한 영상에 대해 본 연구에서 제시된 알고리듬을 이용하여 차선, 정지선 및 과속방지턱을 인지하도록 하였다. ITS 주행시험장의 고속 주행코스는 편도 3차선으로 점선 차선이 그려져 있다. 교차로와 횡단보도는 없으며, 노면 표시와 같은 차선인식 방해 요소가 있다. 카메라로부터 전달받는 영상의 크기는 640 × 480 픽셀이고, 탑뷰 이미지는 250 × 250으로 만들어 진다. 초당 약 30프레임의 영상이 카메라로부터 전송되므로 각 프레임마다 약 33ms의 지연시간을 갖는다. 본 논문에서 제시한 차선, 정지선, 과속방지턱 인식 알고리듬을 모두 수행하는 데 약 18ms가 소요되어 모든 프레임을 문제없이 처리할 수 있었다.

총 8회에 걸쳐 주행시험을 실시하였으며 오전, 오후, 맑은 날, 흐린 날 등 다양한 환경에서 실험을 진행하였다. 실험을 실시한 시험장의 고속 주행코스는 교차로, 횡단보도와 같은 환경은 포함되어 있지 않았지만 노면에 적혀있는 글자와 정지선 같은 방해 요소에도 불구하고 오검출이 발생하지 않았다. 그림 12와 그림 13은 주행코스에서 차선, 정지선 및 과속 방지턱의 인식결과를 보여주는 화면이다. 파란색은 차선, 빨간색은 정지선, 분홍색은 과속방지턱을 나타낸다.

표 1과 2는 전체 실험에서 획득한 차선의 인식률과 정지선 및 과속방지턱의 최초 인식거리를 보여준다. 시험장에서 취득한 영상데이터는 모두 차선이 존재하며 차량이 차선을 벗어나지 않게 수집했다. 차선인식 프레임은 탑뷰 영상에서 차선이 명확히 보일 때, 양쪽 차선이 모두 인식되었을 경우를 카운트 했다. 차선의 경우 95% 이상의 인식률을 보였으며, 정지선과 과속방지턱은 평균적으로 17m 이상의 거리에서 최초 인식되었다. 이미 개발되어 있는 기존 차선 및 정지선 인지 알고리듬[11]과 비교했을 때 인

표 1 차선인식 실험 결과.
Table 1 Experiment results of lane detection.

번호	시간/날씨	총 프레임	차선 인식 프레임	인식률 (%)
1	오전/맑음	9750	9428	96.6
2	오전/맑음	8849	8478	95.8
3	오전/흐림	10447	9960	95.3
4	오전/구름	8725	8424	96.5
5	오후/비	9237	8928	95.7
6	오후/구름	9492	9087	95.7
7	오후/맑음	7845	7483	95.3
8	오후/맑음	9132	8884	97.2

표 2 정지선 및 과속방지턱의 최초인지 거리.
Table 2 Distance of first detection to a stop line and speed bump.

번호	시간/날씨	정지선(m)	과속 방지턱(m)	평균 인지 거리(m)
1	오전/맑음	20.72	17.44	19.08
2	오전/맑음	18.94	16.28	17.61
3	오전/흐림	19.43	17.48	18.45
4	오전/구름	17.96	15.50	16.73
5	오후/비	18.77	16.88	17.82
6	오후/구름	19.46	18.21	18.83
7	오후/맑음	20.12	18.36	19.24
8	오후/맑음	20.94	18.82	19.88

표 3 알고리듬 처리속도 및 인식율 비교
Table 3 Comparison of processing speed and recognition rate

구분	기존 알고리듬	본 알고리듬
처리속도	약 38ms	약 18ms
프레임/초	26 프레임	55 프레임
차선 인식율	90%	96%
정지선 인식율	97.8%	97.8%

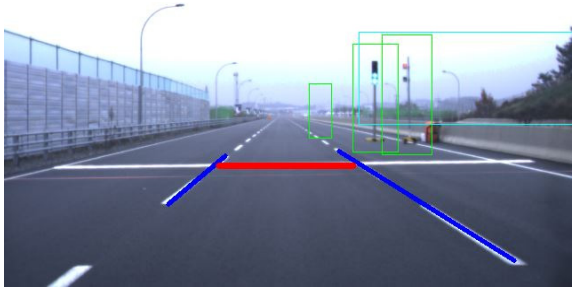


그림 12 차선 및 정지선 인식.
Fig. 12 Detection of lane and stop line.

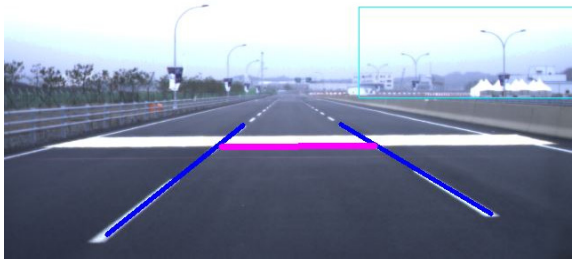
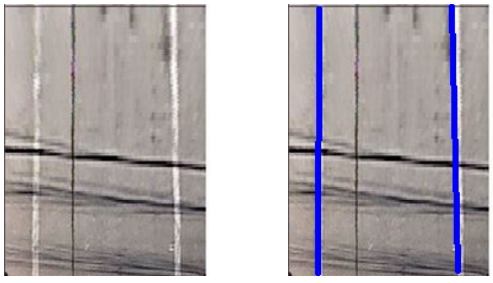


그림 13 차선 및 과속방지턱 인식.
Fig. 13 Detection of lane and speed bump.



(a) (b)

그림 14 흐려진 차선 인식. (a) 인식 전, (b) 인식 후.
Fig. 14 Blurred lane recognition: (a) Before recognition, (b) After recognition.

식률은 비슷하지만 표 3과 같이 처리 속도가 더욱 우수했다. 또한 그림 14에서와 같이 도로의 노후화로 차선이 흐려진 경우에도 명확하게 차선을 인식할 수 있었다.

6. 결 론

본 논문에서는 자율주행 자동차에서 차선, 정지선 및 과속 방지턱을 실시간으로 빠르고 정확하게 인식하기 위한 알고리듬을 제안하였다. 먼저 월드 좌표와 픽셀 좌표를 일대일 매칭하여 탑뷰를 만들고, 구현된 탑뷰를 이용하여 차선과 가장 유사한 특징점만을 후보군으로 저장하여 연산처리속도를 증가시켰다. 서로 연관된 클러스터를 연결하여 차선이 일부 지워지거나 흐려진 곳도 정확하게 인식하여 자율주행차량의 위치보정에 도움을 줄 수 있도록 하였다. 기존의 이미지 와핑(warping) 방식을 이용하여 탑뷰를 제작할 수 있지만, 카메라의 정확한 캘리브레이션(calibration)으로 탑뷰 생성에 필요한 최소한의 연산만을 사용하였다. 또한 도로 위 노면표시의 물리적 특징을 최대한 활용하고 불필요한 과정을 최소화 하여 실시간 처리를 했다는 것에 차별성이 있다. 실험결과 약 5~6분 동안의 주행 중 95% 이상의 인식률을 보였으며 정지선 및 과속방지턱과 같은 노면표시는 18m 이상의 거리에서 최초 인지되었다. 이것은 차량이 30~50km/h의 속도로 움직일 때 충분히 차량을 제어하여 대처할 수 있는 거리이다.

감사의 글

이 논문은 2016년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2016R1D1A1B02012959). 이 논문은 2015년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2015R1A2A1A15054026). 이 논문은 2015년 교육부와 한국연구재단의 지역혁신창의인력양성사업의 지원을 받아 수행된 연구임(No. 2015H1C1A1035914).

References

[1] M. Kang, S. Hur, I. Park, and Y. Park, "Map building based on sensor fusion for autonomous vehicle," The Transactions of the Korean Society of Automotive Engineers, vol. 22, no. 6, pp. 14-22, Sep. 2014.

[2] S. Thrun, et al., "Stanley: the robot that won the DARPA grand challenge," Journal of Field Robotics, vol. 23, no. 9, pp. 661-692, Sep. 2006.

[3] S. Ahn and M. Han, "Research of the lane recognition for an advanced vehicle system," Journal of Korea Institute of Information Technology, vo1. 5, no. 1, pp. 136-142, Mar. 2007.

[4] J. Ahn, T. Oh, and I. Kim, "Development of lane and obstacle detection system using vision sensor," in Proceeding of Information and Control Systems (CICS) Conference, pp. 276-277, Oct. 2008.

[5] J. Yeo, K. Koo and E. Cha, "A lane tracking algorithm using IPM and Kalman filter," Journal of Korea Institute of Information and Communication Engineering, vol. 17, no. 11, pp. 2492-2498, Nov. 2013.

[6] M. Bertozzi, "Stereo inverse perspective mapping: theory and applications," Journal of Image and Vision Computing, vol. 8, pp. 585-590, 1998.

[7] http://autonomous.ksae.org/

[8] R. C. Gonzalez and R. E. Woods, Digital Image Processing (3rd edition), Prentice Hall, 2009.

[9] M. Aly, "Real time detection of lane markers in urban streets," in Proceeding of the Intelligent Vehicles Symposium, Eindhoven, pp. 7-12, 2008.

[10] http://www.kiapi.or.kr/

[11] T. Park and T. Cho, "A crosswalk and stop line recognition system for autonomous vehicles," Journal of Korean Institute of Intelligent Systems, vol. 22, no. 2, pp. 154-160, Apr. 2012.

저 자 소 개

안 영 선 (Young-Sun Ahn)

1990년 8월 20일생. 2015년 2월 계명대학교 전자공학과 졸업(학사). 2015년 3월~현재 계명대학교 전기전자공학과(석사과정). 주관심분야: 자율주행자동차, 비전 영상처리, 임베디드 HW/SW.

E-mail : younsunan@naver.com

곽 성 우 (Seong Woo Kwak)

1970년 3월 10일생. 1993년 2월 한국과학기술원 전기 및 전자공학과 졸업(학사). 1995년 2월 한국과학기술원 전기 및 전자공학과 졸업(석사). 2000년 8월 한국과학기술원 전기 및 전자공학과 졸업(공박). 2000년~2002년 인공위성연구센터 선임연구원, 연구교수. 2003년~현재 계명대학교 전자공학과 교수. 주관심분야: 실시간시스템, 내고장성 기법, 자율주행자동차, 비동기 시스템, 위성 시스템 등.

Tel : 053-580-5926, Fax : 053-580-5165

E-mail : ksw@kmu.ac.kr

양 정 민 (Jung-Min Yang)

1971년 3월 31일생. 1993년 2월 한국과학기술원 전기 및 전자공학과 졸업(학사). 1995년 2월 한국과학기술원 전기 및 전자공학과 졸업(석사). 1999년 2월 한국과학기술원 전기 및 전자공학과 졸업(공박). 1999년 3월~2001년 2월 한국전자통신연구원 컴퓨터·소프트웨어연구소 선임연구원. 2001년 3월~2013년 8월 대구가톨릭대학교 전자공학과 교수. 2013년 9월~현재 경북대학교 전자공학부 교수. 주관심분야: 비동기 순차 머신 제어, 실시간 시스템 등.

Te l: 053-950-7235, Fax : 053-950-5505

E-mail : jmyang@knu..ac.kr