

Bird Eye View 시스템을 이용한 지능형 자동차의 주행차선 및 주차선 인식

Recognition of Driving and Parking Lines for Intelligent Vehicles Using Bird Eye View System

배현수¹, 이석규^{1,#}
Hyun Soo Bae¹ and Suk Gyu Lee^{1,#}

¹ 영남대학교 전기공학과 (Department of Electrical Engineering, Yeungnam University)
Corresponding Author / E-mail: sglee@ynu.ac.kr, TEL: +82-53-810-2487
ORCID: 0000-0001-7237-9062

KEYWORDS: Intelligent vehicle (지능형 자동차), Lane recognition (차선 인식), Top view (평면 시점), Auto parking (자동 주차)

In this paper, we propose a noble line-recognition algorithm of driving parking for intelligent vehicles by using images obtained from a bird's eye view system. To achieve safe driving and parking of unmanned vehicles, we need to obtain noiseless and effective images around vehicles. In addition, fast image processing is a fundamental requirement for the real time recognition of lanes and obstacles to ensure safety. In fact, the number of sensors equipped with conventional unmanned vehicles is reluctant to their commercialization. To solve this problem, we propose a noble method to detect straight lines and turning curves in the images obtained from a bird's eye view system. For conventional vehicles equipped with this bird's eye view system, straight lines and turning curves are detected by using a Hough and trigonometric function. Since parking lines have the form of a rectangle or parallelogram, detection of their vertices makes it possible to determinate parking areas. In the case of a parking space without parking lines, the parking space is detected by using a stereo algorithm after calculating the area for parking. The experimental results using the proposed algorithm show that, without additional sensors, the lines and the surrounding environment are detected for unmanned driving and auto-parking.

Manuscript received: May 24, 2017 / Revised: December 11, 2017 / Accepted: December 18

NOMENCLATURE

fc = Focal length
 cc = Principal point
 kc_1, kc_2, kc_3 = Radial distortion
 kc_4, kc_5 = Tangential distortion
 h_{13}, h_{23} = Translation
 h_{31}, h_{32} = Perspective change
 E = Image variation
 d = Distance

1. 서론

지능형 차량은 운전자의 도움 없이 스스로 주위환경을 인식하면서 지정된 도로를 따라 안전하게 주행하는 자동차를 의미한다. 지능형 차량은 주행 중에 주변의 장애물을 인지하여 속도 조절과 제동이 가능하여야 하며 자신의 현재 위치를 확인하여 목표 위치까지 경로도 확인할 수 있어야 한다.¹ 특히 교통량이 증가함에 따라서 교통사고 유발 요인이 많아지게 되어, 운전자가 안전하게 운전할 수 있도록 차량 주변의 정보를 자동차에 제공해주는 것이 필요하다.²⁻³ 예를 들면 차량 주변의 장애물이나 차량간의 거리를 알려주거나 차선 이탈에 대한 정보 등은 운전자가 안전하고 편리하게 운전할 수 있도록 해준다.⁴ 지능형 자동차와 관련된 기술적인

문제들이 점차 해결되고, 자동차의 안전과 편리성에 대한 관심이 높아짐에 따라서 국내외에서 지능형 차량에 대한 관심이 높아지고 있으며, 국내외의 자동차회사뿐 아니라 IT 업계에서도 자체적인 지능형 차량을 개발하고 있다. 특히 미국에서는 2015년까지 미국의 수송차량의 1/3을 무인화하려는 계획을 수립하였다.⁵ 또한 국내 자동차 업계, IT 업계, 대학, 연구소 등에서 미래 성장 동력 산업의 하나인 지능형 차량에 대하여 많은 투자와 연구를 진행하고 있다. 지능형 차량은 차선을 기준으로 목표 지점까지 차량에 설치된 레이저 센서, 초음파 센서, 카메라 등을 이용하여 획득한 차량 주변의 차선, 장애물 등에 관한 정보를 바탕으로 주변의 상황에 따라 능동적인 대처를 하면서 자동 주행이 되는 시스템이다.⁶ 지능형 차량에서 가장 중요한 기술은 정확한 주변 정보 획득 기술이며, 이를 근거로 차량의 주변정보를 추출하여 차량이 안정적인 주행을 하도록 한다.⁷ 일반적으로 차량의 주변에 관련된 가장 많은 데이터를 추출할 수 있는 정보인 영상정보는 환경의 영향을 쉽게 받게 되어 정확한 정보를 획득할 수 없는 경우가 발생할 수 있어서 차량에 추가적인 센서를 부착하여 이를 보완하는 경우가 많다.^{8,9} 현재 차량 생산 업체와 IT 업체들간의 협업으로 지능형 차량 또는 무인 차량의 개발을 주도하고 있다. 많은 지능형 차량들은 LiDAR (Light Detection and Ranging), 가속도, GPS (Global Positioning System), 지자기 센서 등과 복수개의 카메라와의 융합하여 전방의 장애물과 차선 등의 차량 주행에 필요한 정보를 획득하고 처리한다. 이러한 방법들은 전체 센서의 가격 문제로 인하여 현재 상용화에 있어 가장 큰 문제점으로 알려져 있다. 본 논문에서는 차량에 카메라만을 부착하여 차량 주변의 정보를 사각없이 획득하는 Bird Eye View를 사용하여 차량 주변의 장애물이나 차선을 확인하여 주행 시 차량이 장애물을 효율적으로 회피하고 차선을 기준으로 정확하게 주행하는지 판별할 수 있게 한다. 또한 주차시에는 Bird Eye View를 이용하여 주차선, 주차 가능한 공간, 주차구역 내 장애물을 확인하여 차량이 자동 주차할 수 있도록 도와준다.¹⁰

2. Bird Eye View

최근 여러장의 영상을 합성하여 한 장의 영상으로 만들어 내는 파노라마 기술에 관한 연구가 활발히 진행되고 있다. 인공위성을 통하여 얻은 영상 정보를 이용하여 한 장의 지도를 만들거나 일반 카메라의 사진 촬영에도 파노라마를 많이 사용하고 있다. 이러한 기술을 차량 주행에도 이용하여 차량 주변의 상황을 사각없이 획득하기 위해 사용되고 있다. 즉 차량 사방에 설치된 카메라를 이용하여 차량 주변의 영상을 정합하여 주변 환경을 Top View 형식으로 볼 수 있게 해준다.^{12,13}

카메라를 이용하여 획득되는 영상을 사용하기 위하여 먼저 카메라의 특징을 카메라 Calibration을 이용해 확인하여야 한다.

$$k = \begin{bmatrix} fc_1 \text{ skew } cc_1 \\ 0 & fc_2 & cc_2 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

여기서, fc 는 Focal Distance, fc_1 과 fc_2 는 각각 Focal Length, cc_1 과 cc_2 는 Principal Point를 skew는 비대칭계수를 나타낸다. CCD 센서를 사용하는 경우에는 오차가 존재하여 fc 는 일반적으로 fc_1 과 fc_2 가 비슷한 값을 갖지만 CCD Pixel이 직사각형일 경우 fc_2/fc_1 이 1이 아닌 값을 가진다. 일반적으로 카메라의 화각이 넓을수록 왜곡이 심하게 나타나며, 차량의 전방향 영상을 획득하기 위해서 일반적으로 약 180°의 화각을 가지는 카메라를 사용한다. 이러한 왜곡을 보정하기 위해 Radial과 Tangential 왜곡 모델을 이용한다.¹¹ 일반적으로 방사 왜곡은 kc_1 , kc_2 , kc_3 3의 파라미터를 이용하여 반지름 r 로부터 떨어진 정도를 보정하지만, 이것보다는 접선 왜곡은 kc_4 , kc_5 두 개의 파라미터를 이용하여 이동량을 보정한다. 카메라의 왜곡을 보정하기 위한 영상 왜곡 보정 파라미터를 구해야 하는데 식(2)와 같이 왜곡 보정 파라미터를 픽셀에 적용시켜 보정된 좌표를 계산한다.^{14,15}

$$x_d = ((1 + kc_1 r^2 + kc_2 r^4 + kc_3 r^6) \cdot x_n + d_x) \quad (2)$$

여기서,

$$d_x = \begin{bmatrix} 2kc_4 + xy + kc_5(r^2 + 2x^2) \\ kc_1(r^2 + 2y^2) + 2kc_5xy \end{bmatrix} \quad (3)$$

카메라를 사용하기 위해서는 내부 파라미터와 외부 파라미터도 구하여야 한다. 카메라와 실제 지면의 좌표계 사이의 회전과 이동은 외부 파라미터와 앞에서 구한 내부 파라미터를 이용하여 각 카메라 영상마다 대응되는 실제 좌표 사이의 호모그래피 행렬을 이용하여 구할 수 있으며, 모든 카메라를 같은 점으로 맞추어 주어야 영상을 바르게 정합할 수 있다.

우선 Fig. 1과 같이 오차 보정을 진행하기 위하여 차량을 미리 크기를 정해 놓은 체스보드 위에서 카메라를 보정을 해준다. 주행

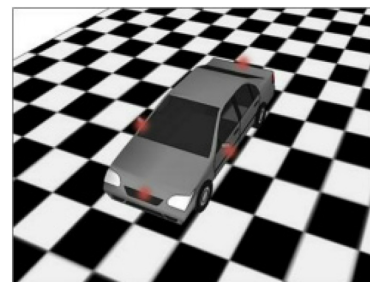


Fig. 1 Chess board for the camera location and calibration for Bird eye view

중에 항상 특징점을 이용하여 영상을 정합할 수 없기 때문에 미리 체스보드위에서 카메라의 임의의 특징점을 잡아둔다.

일반적으로 카메라를 보정할 때 체스보드를 이용하는 이유는 영상을 이용하여 정합하기 위해 정합점을 정확하게 찾을 수 있고, 이에 따라 호모그래피 행렬을 구하여 정확하게 호모그래피 값을 얻을 수 있기 때문이다. 식(4)는 호모그래피 변환 행렬로서 w 는 Homogeneous 좌표에서의 Homography 스케일, h_{11} , h_{12} , h_{13} , h_{14} 는 회전, 스케일, Shearing, Reflection을 h_{13} , h_{23} 는 평행이동을 h_{31} , h_{32} 는 원근 변화를 각각 나타내는 파라미터이다.

$$w = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4)$$

여기서, 각 영상에서 구해지는 호모그래피 파라미터를 이용하여 각 영상을 보정하여 정합을 해주어야 하는데 각 다른 카메라에서 획득되는 영상을 보정이 된 영상으로 변환 후 Fig. 2에서와 같이 각 변환된 영상에서 체스보드 기준의 특징 점을 추출하여 각 영상 사이의 특징점 기준으로 영상을 정합을 해준다.¹⁶ 이와 같이 카메라에서 입력되는 영상을 이용하여 정합을 하는 것이 아니라 기존의 체스보드의 특징점을 이용하여 빠르게 정합 한다. Fig. 3은 실제 카메라에서 입력되는 4개의 영상을 보여주며 각 영상의 빨간 부분은 체스보드를 이용하여 미리 획득한 특징점 부분을 의미한다.

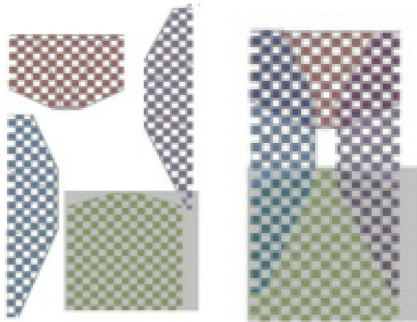


Fig. 2 Image matching depending on location

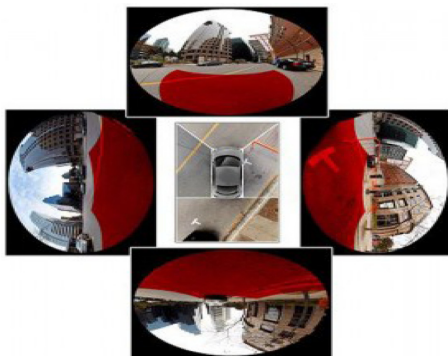


Fig. 3 The image matched with the image acquired from the vehicle

3. 실선 및 점선 차선 인식

일반적인 도로의 차선은 다음과 같은 일반적인 특징을 가지고 있다고 가정한다. 즉, 평평한 도로라는 가정하에 차선은 흰색, 노란색 등 원색 계열이다. 또한 좁은 영역에서는 직선으로 볼 수 있으며 도로에는 두 개 이상의 차선이 존재하는 등의 특징을 가지고 있고, 차선은 직선이며 평행할 때 하나의 소실점에서 만난다. Fig. 4는 차선의 특징을 나타내는 그림이며 영상에서 Edge를 검출하여 차선이 있는 후보지역을 Mask로 영역을 지정한 뒤 차선 후보점을 평가한다.¹⁶

입력된 영상에서 차선을 검출하기 위해서 차선이 나오는 부분부터 소실점까지 영상의 ROI 추출을 해준다. ROI로 추출된 영상을 Gray 영상으로 변환해주고 Gray 영상에서 노이즈 제거를 위하여 Gaussian Kernel을 컨벌루션하여 영상을 부드럽게 해준다. 영상에서 차선을 추출하기 위해 Edge를 추출하게 되는데 Edge과정에서 찾은 점들 중 y 축 값을 가지는 두 점 $a = (a_x, a_y)$, $b = (b_x, b_y)$ 를 골라 두 점간의 거리가 차선의 폭(w)을 만족하는지 검사한다. 여기서 두 점간의 거리는 식(5)로 계산이 가능하며, 그리고 차선의 폭은 y 값에 따라 식(6)으로 결정 된다. 두 점에 대한 Edge의 강도와 방향을 비교하여 서로 Edge의 강도가 유사하고 방향이 반대일 때 차선의 확률이 높으므로 이 두 점에 대하여 중점을 구해 차선의 후보점으로 저장한다.¹⁷

$$w_{ab} = b_x - a_x \quad (5)$$

$$w = w_{\max} \times (y - a_y) / (y - b_y) \quad (6)$$

$$(c_x, c_y) = \left(\frac{a_x + b_x}{2}, a_y \right) \quad (7)$$

식(7)는 차선에서 x 축 방향으로 가지는 값과 y 축 방향으로 가지는 값이 존재하는데 이때 x 축은 차선의 a_x 는 후보점의 a 와 b 의 합성으로 방향성분을 계산한다. 이 후보점들을 이용하여 Hough 변환 후 RANSAC알고리즘으로 소실점으로 향하는 선분들만 추출하여 차선을 확인한다. 추출된 차선에서 주행을 위해 실선과 점

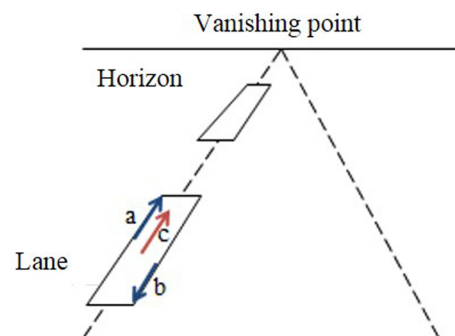


Fig. 4 Direction of the driving lane, vanishing point and candidate point

선을 구분해 차선 이 변경 가능한지 확인한다. Hough 변환된 결과가 30픽셀 이상 이어져 있으면 실선으로 판단하고 그렇지 않으면 점선으로 결과를 확인한다.

4. Bird Eye View를 이용한 주차선 및 주차 공간인식

4.1 주차선인식

일반적인 주차선도 도로의 차선과 같이 일반적인 특징을 가지고 있다고 가정하였다. 즉, 일반적으로 주차선은 흰색으로 이루어져 있으며, 직사각형과 평행사변형의 형태를 가지고 있다. 이러한 특징을 이용하여 위의 차선을 검출하는 방법을 이용하여 주차선을 검출한다. 주차선은 일정한 간격을 가지고 있으므로 모서리 점들을 모아 직사각형의 특징인 각 모서리 마다 각이 90°를 가지는 후보군을 검색하고 평행사변형의 특징을 가지는 일정한 각도 범위 내에 있는 모서리를 검색하여 주차구역 후보군을 검출하여 차량에서 가깝고 주차선 후보 위에 장애물이 없는 주차구역 후보군을 선택하여 주차구역으로 선택한다.¹⁸ 주차구역을 인식하기 위해서는 좌, 우와 후방의 정보가 중요하다. 이러한 정보를 처리하기 위해서 Fig. 5와 같이 후방의 인식을 위해 차량의 앞쪽을 제외한 뒤, 좌, 우, 3개의 카메라를 이용한 영상 정합을 보여주며 기존의 정합방법에서 전면 영상만 제외하고 진행된다.

Fig. 6은 일반적으로 많이 사용되는 주차선 중 직각, 평행, 사각 주차선이다. 주차선에서 차량이 주차가 되어야 하는 공간은 흰색 선 안에 위치에 주차를 해야 하므로 흰색선의 안쪽 모서리들을 확인하여 공간을 확인한다.

Fig. 7과 같이 주차선의 4개의 꼭지점을 검출하여 주차선 내 주차가 가능한 공간을 확인한다. 이때 코너 검출은 Harris Corner를 이용하여 검출한다. 1픽셀씩 이동시킬 때 영상변화 량 E의 최소 값을 해당 픽셀의 변화량 값으로 설정하고, 설정 된 값이 지역적으로 극대가 되는 지점이 코너이다. 식(8)을 이용하여 픽셀을 이동시켰을 때 영상변화량 E를 계산한다.

$$E(\Delta x, \Delta y) = \sum_w [I(x_i + \Delta x, y_i + \Delta y) - I(x_i, y_i)]^2 \quad (8)$$

이때 이동량이 매우 작다고 가정하고 I를 미분하여 선형적으로 근사하면 식(9)과 같이 나타낼 수 있다.

$$I(x_i + \Delta x, y_i + \Delta y) \approx I(x_i, y_i) + [I_x(x_i, y_i) I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (9)$$

식(8)과 식(9)을 이용하여 식(10)을 유도하고

$$E(\Delta x, \Delta y) \approx \sum_w [I(x_i, y_i) + [I_x(x_i, y_i) I_y(x_i, y_i)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} - I(x_i, y_i)]^2 \quad (10)$$

식(10)을 이용하여 식(11)을 유도한다. 식(11)의 유도된 2 X 2

행렬의 고유값을 λ_1, λ_2 라고 하면 영상 변화량은 λ_1 의 고유벡터 방향으로 이동할 때 최대가 되고 λ_2 의 고유벡터 방향으로 이동할 때 최소가 된다. 두 고유값을 구했을 때 두 값이 모두 크게 된다면 그 부분이 코너라 판단할 수 있다.

$$E(\Delta x, \Delta y) \approx [\Delta x, \Delta y] \begin{bmatrix} \sum_w I_x(x_i, y_i)^2 & \sum_w I_x(x_i, y_i) I_y(x_i, y_i) \\ \sum_w I_x(x_i, y_i) I_y(x_i, y_i) & \sum_w I_y(x_i, y_i)^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (11)$$

검출된 코너점들을 이용하여 사각형을 만들어 주며, 가로는 한 코너점에서 양쪽에 생성되어 있는 코너점의 픽셀 값을 계산해서 가장 먼 코너점과의 직선을 만들고 세로로는 같은 코너점에서의 위와 아래의 코너점을 비교하여 가장 가까운 코너점과 직선을 만들어 주차공간의 사각형을 생성한다.

4.2 주차선이 없는 주차구역 인식

주차할 때 항상 주차선이 있는 경우에 만 주차하는 것이 아니라 주차선이 없는 구역에 주차를 하는 경우도 생기게 된다. 이를 위해서 먼저 Bird eye view를 위해서 4대의 카메라에서 획득되는

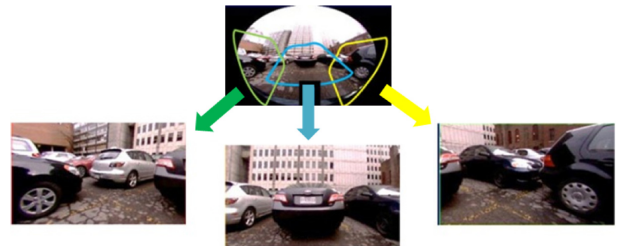


Fig. 5 Camera images to measure the obstacle behind the vehicle

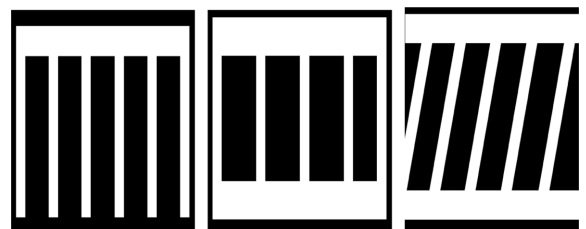


Fig. 6 Parking lines for right, parallel, and 45° parking

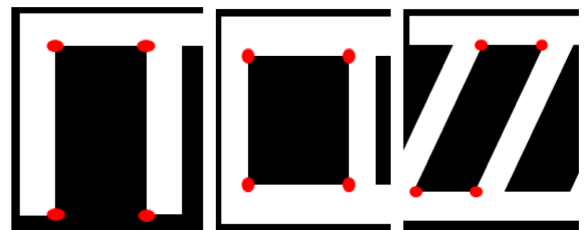


Fig. 7 Vertex inside the parking line

영상을 정합하기 위해서 정해놓은 매칭점들을 이용하여 Stereo 알고리즘을 통해 장애물과 차량의 거리를 측정하여 주차선이 없어도 안전하게 주차를 할 수 있도록 해준다. 주차 가능한 공간을 찾기 위해 장애물이 없는 환경 중에서 차량의 공간 이상의 빈 공간을 검색해야 하는데 이는 Stereo 알고리즘을 사용한다. 매칭되는 특징점들을 이용하여 Fig. 8과 같이 두개의 카메라를 이용한 삼각측량법을 사용하여 카메라와 물체사이의 거리를 다음과 같이 구할 수 있다.

삼각함수에 의해 식(12)를 유도하고

$$F^2 = p_1^2 + \alpha_1^2 - 2p_1\alpha_1 \cos \theta_1 \quad (12)$$

피타고라스 정리와 식(12)를 이용하여 θ_1 에 대해 다음과 같이 정리하고 θ_2 도 같은 방법으로 구할 수 있다.

$$\theta_1 = \cos^{-1} \left(\frac{p_1}{\sqrt{p_1^2 + F^2}} \right) \quad (13)$$

각 θ_1, θ_2 에 관하여 정리하면 식(14), 식(15)와 같다.

$$\tan \theta_1 = \frac{d}{l} \quad (14)$$

$$\tan \theta_2 = \frac{d}{l-1} \quad (15)$$

식(14), 식(15)를 이용하여 d 에 관해 정리하면 다음과 같이 식(16)을 유도한다

$$d = l \times \frac{\tan \theta_1 \tan \theta_2}{\tan \theta_1 + \tan \theta_2} \quad (16)$$

차량이 주차하기 위한 필요한 공간을 최초로 차량 크기 파라미터로 정해놓은 뒤 Bird Eye View 영상 변환을 위해 미리 체크보드에서 추출한 특징점들을 사용해 Stereo 알고리즘으로 영상에서 입력되는 모든 공간을 3차원으로 검색한다. 차량이 주행하는 동안 주차를 할 수 있는 공간을 계속해서 검색하여 주차가 가능한 공간이 발견이 되면 운전자에게 알려주게 된다.¹⁹⁾

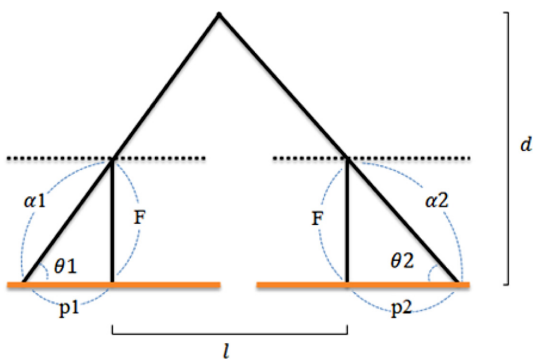


Fig. 8 Stereo triangulation

5. 시뮬레이션

제안된 알고리즘은 Fig. 9와 같이 주행 차선과 주차선들을 실제 차량에 부착되는 카메라의 스펙을 사용하여 시뮬레이션 하였다. 즉, 카메라는 1/3.6" CMOS, 화각 185°, 해상도 640 X 480을 파라미터를 가지며, 차량은 일반적인 가솔린 엔진을 사용하는 국내 준중형 차량의 파라미터를 사용하였다. 시뮬레이션에 사용되는 차선은 도로교통법 기준으로 폭 15 cm, 주차선은 건축법에 따라 가로 5 m, 세로 2.3 m로 구축하여 진행하였다.

5.1 차선인식 시뮬레이션

Fig. 10은 시뮬레이션상에서 전방 카메라에 획득되는 영상을 보여준다.

알고리즘을 통하여 획득한 영상을 이진화 후 차량 소실점을 제거하고, 차선일 확률이 높은 부분을 Fig. 11과 같이 빨간 선으로

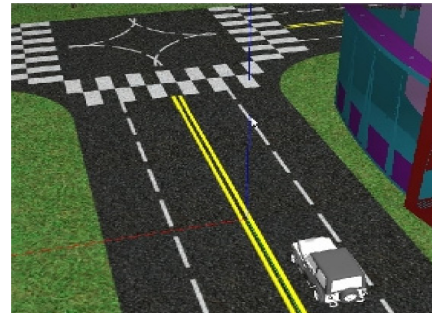


Fig. 9 Road environment for simulation

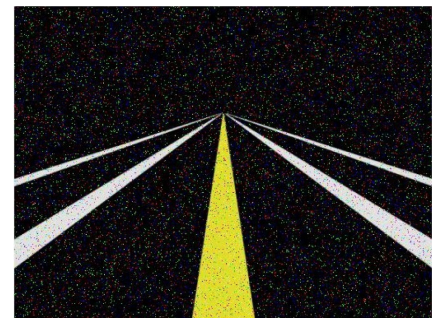


Fig. 10 Straight line on the road for simulation

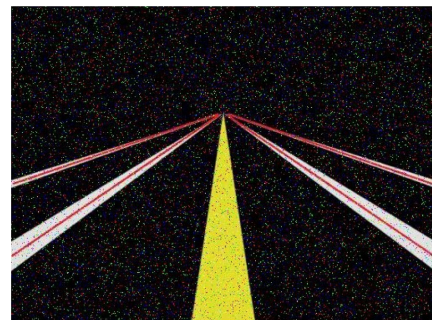


Fig. 11 Straight road for simulation

표시하였다. 실제의 차선을 모델링하여 차선을 검출되는 것을 확인하기 위해서 앞의 차선 검출 알고리즘을 이용하면 차선에 따라 차선이 검출되는 것을 확인할 수 있다.

직진 차선을 시뮬레이션 한 결과 운전자가 보는 시야에서 시작되는 차선에서 소실점까지 빨간색으로 표시되는 것을 확인하였다.

5.2 주차선 인식 시뮬레이션

Fig. 12 주차선 및 주차공간 인식을 위해 시뮬레이션상의 카메라에서 취득 되는 영상을 보여준다.

주차선은 차선과 마찬가지로 직선을 이루고 있다는 공통점을 가지지만 주차선 만이 가지는 고유한 특성인 사각형들이 모여 직사각형과 평행사변형을 만들 수 있다는 특성을 이용하여 알고리즘을 시뮬레이션하였다. Fig. 13에서는 모델링된 주차선을 이용하여 알고리즘을 이용하여 주차선이 정확하게 추출되는 시뮬레이션 결과를 확인할 수 있다.

위의 시뮬레이션 결과를 보면 초록색 주차공간에 맞추어 주차선에 따라 빨간 선이 표시되는 것을 시뮬레이션 상으로 확인 했으며, 이를 이용하여 주차공간까지 파악할 수 있다.

6. 실험

실험은 시뮬레이션에서 사용한 것과 동일한 스펙을 가지는 카메라를 4대를 사용하였고 실제자동차를 대신하여 RC카(L490 XW380XH280, 모터 BLDC 540급)을 사용하였다. 실험 환경은 일반 도로와 주차장에서 진행되었다. Fig. 14는 실험을 진행하기

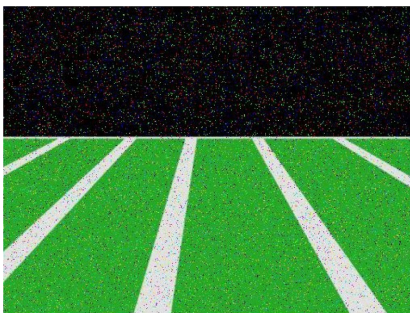


Fig. 12 Parking lane environment for simulation

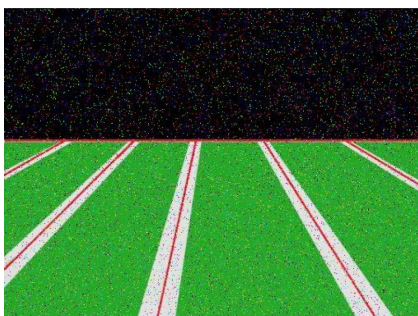


Fig. 13 Parking lane for simulation

위해 영상을 정합한 이미지와 UI를 보여준다.

Fig. 15는 실험에 사용된 RC카 및 장비이다.

6.1 차선 인식

Fig. 16은 실제도로에서 카메라를 이용하여 추출된 영상을 Bird Eye View영상으로 정합을 한 뒤에 차선추출 알고리즘을 통하여 차선을 추출하였다. 알고리즘의 반응속도는 약 60 km/h로 주행하는 차량에서도 무리 없이 차선을 인식을 하는 것으로 보였으며 곡선 차선이 나누어 지는 분기점에서도 차선이 정확하게 인식되었다.

Fig. 17은 실제 차선에서의 차선의 확률을 낮추고 인식률이 안 좋은 상태에서 실험을 추가 하기 위해 입력되는 영상에서 랜덤한 흰색 선과 Speckle 노이즈를 추가하여 실험하였다.

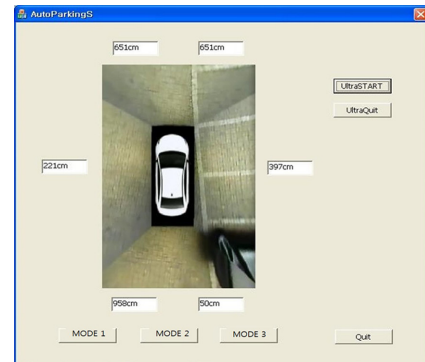


Fig. 14 Program implementation for around view

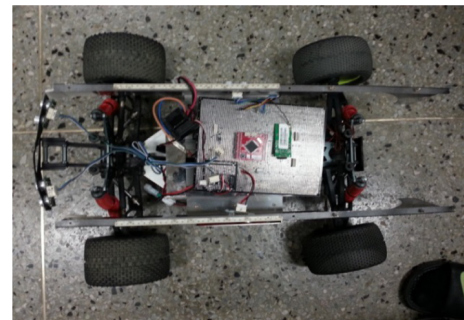


Fig. 15 Intelligent vehicle mounted with a camera

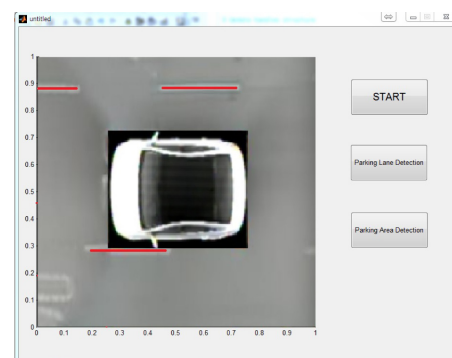


Fig. 16 Lane recognition of the actual lane

6.2 주차선 인식

위의 실험과 마찬가지로 Bird Eye View 영상에서 주변의 주차선을 인식하여 Fig. 18과 같이 빨간 선으로 주차구역을 표시를 해주었다. 10 km/h의 속도로 진행을 하는 차량에서 차량 주변의 주차선을 인식하기 위해 주차선 검출 알고리즘을 사용하여 차량 주변의 주차선을 검출하여 표시한다.

6.3 주차선이 없는 주차구역 인식

주차선이 없는 주차구역 인식은 주차선을 인식하지 않고 위에 설명한 알고리즘을 이용하여 주차구역을 Fig. 19와 같이 빨간 선으로 표시를 해주도록 하였다. 차량과 가장 가까이 있는 주차 가능한 공간을 판단하여 화면에 선으로 표시를 하여 사용자가 주차 가능한 공간을 인지 할 수 있도록 하였고 실제로 앞에 보이는 공간이 주차가 가능한지 아니면 주차를 하기에 불가능한 공간인지 판단할 수 있다.

6.4 야간 환경에서의 차선 인식

야간에도 차선을 인식하기 위해서 차량 주변에 LED를 부착하여 카메라에서 필요한 조도를 최소한으로 확보할 수 있도록 하였다.

Fig. 20은 야간에 가로등이 없는 환경에서 차량에 부착되어 있

는 LED를 이용하여 시야를 확보하고 카메라에서 획득되는 영상을 이용하여 차선을 확인하였다. 주간에 비해 정확하게 차선이 인식되지는 않지만 차량 진행 방향으로 차선이 인식되는 것을 확인하였다.

Fig. 21는 야간에 가로등이 있는 환경에서 차선인식 결과를 보여주고 있다. Fig. 21의 가로등이 없는 환경과 비교하면 차량 주



Fig. 19 Recognition of the parking space without parking lane

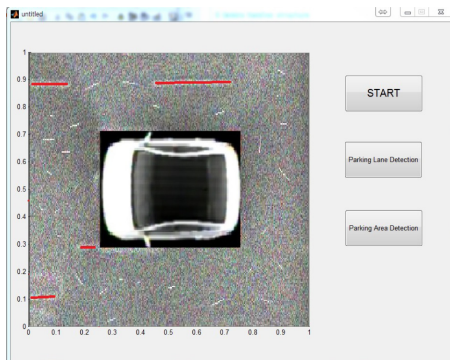


Fig. 17 Lane recognition of the actual lane added noise

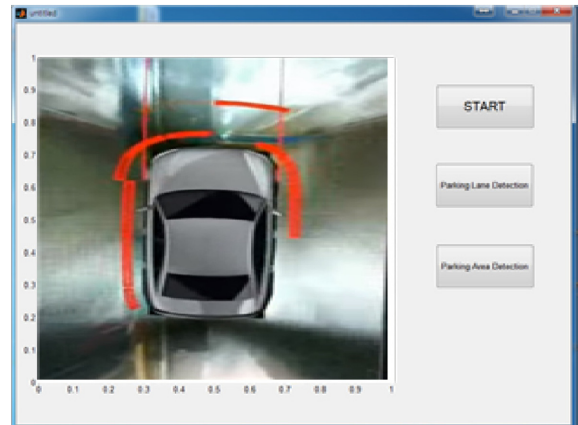


Fig. 20 Recognition of lane without streetlights

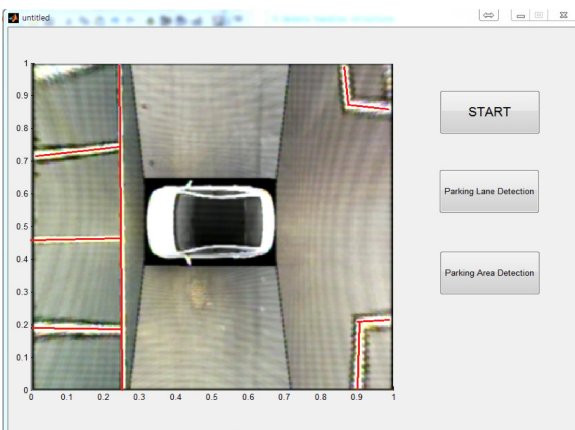


Fig. 18 Lane recognition in the parking lot.

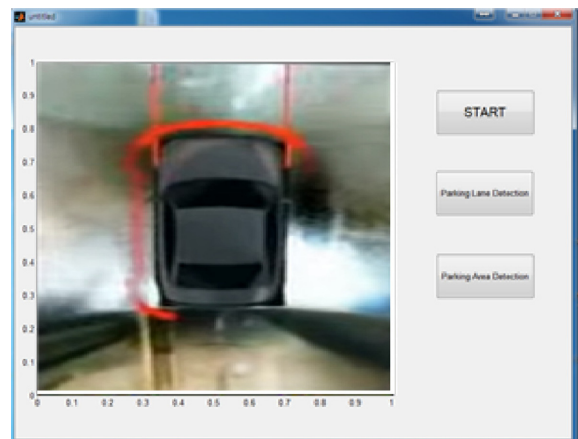


Fig. 21 Line recognition with streetlights

변의 노이즈가 많이 사라지고 차량 진행 방향의 정확한 라인이 생성되는 것을 확인하였다. 따라서 제안된 알고리즘으로 야간에도 차선 인식이 가능하며 가로등이 없어도 차량 주변의 약간의 빛으로도 차선이 인식이 가능한 것을 확인하였다.

7. 결론

본 논문에서는 시뮬레이션과 실험을 통하여 지능형 차량에서 추가적인 센서없이 카메라만 이용하여 제안된 알고리즘을 사용하여 차선인식, 주차선과 주차공간을 인식할 수 있다는 것을 보여주고 있다. 지능형 차량에서 최소한의 센서를 사용하여 자율주행과 자동주차가 가능하다는 것을 확인하였으며 이에 따라 지능형차량 제작 비용을 줄여 상용화를 앞당길 수 있을 것으로 판단된다. 또한 차량의 헤드라이트를 통하여도 어두운 환경에서 충분히 차선을 검출할 수 있기 때문에 카메라가 문제가 생겨 정상적인 동작을 하지 않는 경우를 위해 일시적으로 사용할 센서만 부착하면 된다는 것을 확인할 수 있었다. 카메라의 화각이 겹치는 부분에서는 장애물과의 거리 값이 전부 계산되지만 계산상의 문제로 영상이 겹쳐지는 부분만 장애물과의 거리 값을 계산하고 나머지는 장애물과의 거리를 계산하지 않는다. 추후 렌즈 화각이 겹치는 부분의 모든 영역의 파라미터들을 저장하여 차량이 주행하는 동안 많은 영역에 대해 장애물과의 거리를 계산하여 장애물이 많은 상황에서 주행과 주차를 안정적으로 해야 하며, 직각과 대각 주차선에 대해서만 주차선이 인식이 되는 부분도 더욱 다양한 주차선에 대해 인식이 되도록 해야 할 예정이다.

ACKNOWLEDGEMENT

이 연구는 2016년도 영남대학교 학술연구조성비에 의한 것임.

REFERENCES

1. Sharp, R. and Peng, H., "Vehicle Dynamics Applications of Optimal Control Theory," *Vehicle System Dynamics*, Vol. 49, No. 7, pp. 1073-1111, 2011.
2. Bender, J. G., "An Overview of Systems Studies of Automated Highway Systems," *IEEE Transactions on Vehicular Technology*, Vol. 40, No. 1, pp. 82-99, 1991.
3. Choi, Y. W., Kim, K. D., Choi, J. W., and Lee, S. G., "Laser Image SLAM Based on Image Matching for Navigation of a Mobile Robot," *Journal of the Korean Society for Precision Engineering*, Vol. 30, No. 2, pp. 177-184, 2013.
4. Hedrick, J., Tomizuka, M., and Varaiya, P., "Control Issues in Automated Highway Systems," *IEEE Control Systems*, Vol. 14, No. 6, pp. 21-32, 1994.
5. Thorpe, C., Hebert, M. H., Kanade, T., and Shafer, S. A., "Vision and Navigation for the Carnegie-Mellon Navlab," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 3, pp. 362-373, 1988.
6. Kim, S. H., Lee, S. I., Lee, K. S., Cho, S. I., Park, J. H., et al., "Understanding Lane Number for Video-Based Car Navigation System," *Korea Spatial Information Society*, Vol. 1, No. 1, pp. 104-111, 2009.
7. Sunwoo, M. H. and Jeong, D. H., "The Concept and Technologies of Intelligent Vehicle," *The Korean Society of Automotive Engineers*, Vol. 27, No. 6, pp. 30-33, 2005.
8. Suh, S. and Kang, Y., "A Robust Lane Recognition Technique for Vision-Based Navigation with a Multiple Clue-Based Filtration Algorithm," *International Journal of Control and Automation and Systems*, Vol. 9, No. 2, pp. 348-357, 2011.
9. Ahn, S.-J. and Han, M.-H., "Research of The Lane Recognition for an Advanced Vehicle System," *Journal of Korean Institute of Information Technology*, Vol. 5, No. 1, pp. 136-142, 2007.
10. Zhu, C. H., Hirahara, K., and Ikeuchi, K., "Street-Parking Vehicle Detection Using Line Scan Camera," *Proc. of IEEE Intelligent Vehicles Symposium*, pp. 575-580, 2003.
11. Duane, C. B., "Close-Range Camera Calibration," *Photogrammetric Engineering*, Vol. 37, No. 8, pp. 855-866, 1971.
12. Lin, C.-C. and Wang, M.-S., "A Vision Based Top-View Transformation Model for a Vehicle Parking Assistant," *Sensors*, Vol. 12, No. 4, pp. 4431-4446, 2012.
13. Baek, S., Kim, H., and Boo, K., "A Method for Rear-Side Vehicle Detection and Tracking with Vision System," *Journal of the Korean Society for Precision Engineering*, Vol. 31, No. 3, pp. 233-241, 2014.
14. Kim, H., Ryu, M., and Lee, S., "Lane Detection System for Self-Driving Car," *Human Computer Interaction*, pp. 205-209, 2008.
15. Kluge, K. and Lakshmanan, S., "A Deformable-Template Approach to Lane Detection," *Proc. of the Intelligent Vehicles' 95 Symposium*, pp. 54-59, 1995.
16. Rhee, S.-A., Lee, T.-Y., Kim, T.-J., and Sung, J.-G., "Development of a Lane Detect Algorithm from Road-Facing Cameras on a Vehicle," *Journal of Korean Society for Geospatial Information System*, Vol. 13, No. 3, pp. 87-94, 2005.
17. You, F., Zhang, R., Zhong, L., Wang, H., and Xu, J., "Lane Detection Algorithm for Night-Time Digital Image Based on Distribution Feature of Boundary Pixels," *Journal of the Optical Society of Korea*, Vol. 17, No. 2, pp. 188-199, 2013.
18. Kamiyama, H., "Parking Space Detection Using Side-Camera Image," *Proc. of 15th World Congress on Intelligent Transport Systems and ITS America's 2008 Annual Meeting*, 2008.
19. Suh, S., and Kang, Y., "A Robust Lane Recognition Technique for Vision-Based Navigation with a Multiple Clue-Based Filtration Algorithm," *International Journal of Control, Automation and Systems*, Vol. 9, No. 2, Paper No. 348, 2011.