

논문 2014-51-1-16

Accumulator cells를 최적화한 안드로이드 기반의 차선 검출 시스템 개발

(Lane Detection System Development based on Android using Optimized Accumulator Cells)

척트바타르 엘덴토야*, 장 영 민*, 조 재 현*, 조 상 복**

(ErdenetuyaTsogtbaatar, Young-Min Jang, Jae-Hyun Cho, and Sang-Bock Cho[©])

요 약

지능형 교통 시스템(ITS) 및 지능형 자동차의 운전자 보조 시스템에서 차선의 경계를 검출하기 위한 허프 변환 방법이 많이 연구되고 있다. 이 방법의 경우 차선을 효과적으로 인식하지만 차선 이외의 영역의 직선들도 인식할 수 있기 때문에 인식이 떨어질 수 있고 연산속도가 늦어진다. 본 논문에서는 이러한 문제를 해결하기 위해 Hough space에 Accumulator cells를 최적화한 방법을 이용해서 차선 경계를 인식하는 알고리즘을 제안하였다. 이를 바탕으로 H/W 검증을 통해 안드로이드용 어플리케이션을 개발하였다. 스마트 기기의 사용자라면 언제 어디서든 운전자의 주행안전을 위한 차선검출 및 차선이탈 경보시스템을 사용 할 수 있도록 하였다. 소프트웨어 검증은 OpenCV를 사용하여 93.1%의 높은 차선인식률을 보였으며, 하드웨어 실시간 검증은 안드로이드용 휴대폰을 사용하여 68.89%의 차선인식률을 보였다.

Abstract

In the Advanced Driver Assistance Systems (ADAS) of smart vehicle and Intelligent Transportation System (ITS) for to detect the boundary of lane is being studied a lot of Hough Transform. This method detects correctly recognition the lane. But recognition rate can fall due to detecting straight lines outside of the lane. In order to solve this problems, this paper proposed an algorithm to recognize the lane boundaries and the accumulator cells in Hough space. Based on proposed algorithm, we develop application for Android was developed by H/W verification. Users of smart phone devices could use lane detection and lane departure warning systems for driver's safety whenever and wherever. Software verification using the OpenCV showed efficiency recognition correct rate of 93.8% and hardware real-time verification for an application development in the Android phone showed recognition correct rate of 70%.

Keywords : Lane detection; Hough transform; Edge detection; Accumulator cells; Android

I. 서 론

2012년 교통사고 분석에 따르면 223,441건의 교통사

고가 발생했으며, 5,363명이 사망하고, 341,437명이 부상을 입은 것으로 나타났다. 특히 운전자의 부주의로 인한 사건 사고들이 점점 증가되어 가고 있는 추세이다.

차선 검출은 ITS(지능형 교통 시스템)의 구성 요소 중에서 핵심적인 역할을 하고 있으며 차선 이탈 경고 시스템에 적용되어 교통사고를 줄이는데 도움을 줄 수 있는 중요한 역할을 하고 있다.

차선 검출과 관련하여 차선 이탈 경보 시스템을 뜻하

* 학생회원, ** 평생회원, 울산대학교 전기공학부
(University of Ulsan, School of Electrical Eng.)

© Corresponding Author (E-mail: sbcho@ulsan.ac.kr)

※ 본 논문은 2012년 울산대학교 교비연구비의 지원에 의해 작성되었음.

접수일자: 2013년8월30일, 수정완료일: 2014년1월2일

는 LDWS(Lane Detection Warning System)의 중요성이 커지면서 다양한 분야에서 관련된 기술을 개발 중이다. 보쉬의 경우 다목적카메라의 하드웨어 플랫폼을 기반으로 하여 제품 사양별 영상 소프트웨어 컴포넌트를 가변적으로 선택, 설계할 수 있는 ‘통합 영상 인식 소프트웨어 및 주행 지원 시스템’을 개발 중이며, 현대 모비스의 경우 LDWS를 현대차에 양산 및 공급을 하고 있으며 대향 차량 및 선행 차량 인식 등의 기술에 대해서는 선행 개발 및 개념 설계 단계에 있다.

대부분의 자율이동차량은 차량에 설치된 카메라를 통해 얻은 영상을 통해 길을 찾고 주행 제어를 하고 있으며, 그러기 위해서는 카메라 영상으로부터 차량의 위치, 차선 그리고 차량의 방향과 같은 의미 있는 정보를 얻어야 한다. 카메라 영상에서 추출하는 정보들 중에서도 영상으로부터 차선을 추출하는 것에 많은 관심이 모아지고 있으며, 카메라를 통해 얻은 도로 영상은 많은 잡음을 가지고 있기 때문에 이러한 잡음에 강한 많은 차선 검출 알고리즘들이 연구되고 있다.

차선 감지를 위한 여러 가지 방법들이 연구가 되고 있으며 이와 관련해 여러 방법들이 사용되었다. 영역을 세분화하여 차선을 감지하는 알고리즘^[1]의 경우 주변광에 매우 민감하고 효과를 극대화하기 위해 추가적인 처리 과정이 필요로 하며, B-snake를 이용한 차선 감지 및 추적알고리즘^[2], RANSAC를 이용한 알고리즘^[3, 4]의 방법도 연구 되어 있다. 기존의 일반적인 허프 변환 기법^[9,10]의 경우 허프 변환 알고리즘 적용 뒤 morphology 직선 인식률이 높아 차선 검출을 위한 알고리즘으로 널리 사용 되지만 차선 경계뿐만 아니라 영상 내의 다른 직선들도 인식하는 문제가 생긴다. 또한 블랙 박스 등의 차량용 영상기기와 같이 차량 내에서 차도를 찍은 영상이 아닌 차도 외부에서 찍은 이미지에 알고리즘이 적용 되어 시뮬레이션을 진행하였다.

따라서 본 논문에서는 기존의 알고리즘 및 일반적인 허프 변환의 차선인식 문제를 보완하기 위해 허프 영역에 accumulator cells를 최적화한 알고리즘을 이용하여 높은 차선 인식 효율 및 실시간 처리 시간을 줄일 수 있는 방법을 제안한다.

제안한 방법을 차량 내에서 촬영한 영상에 적용하여 주행자를 위한 차선 인식이 되는지 시뮬레이션을 진행하였다.

또한 이를 바탕으로 안드로이드용 어플리케이션을

개발하여 향후 스마트 기기의 사용자라면 언제 어디서든 운전자의 주행안전을 위한 차선검출 및 차선이탈 경보시스템을 사용할 수 있도록 H/W 검증은 하였다.

본 논문은 II장에서 전체 차선 검출 시스템 및 Accumulator cells를 최적화한 허프 변환 기법을 제안하며, III장에서는 차선 검출 시스템의 S/W, H/W 검증 및 개발에 대해 소개한다. IV장에서는 실험 결과를 통해 차선 검출에 대한 성능을 비교 분석하고, V장에서 결론을 맺는다.

II. 제안하는 차선검출 시스템

그림 1은 제안하는 차선 경계 검출 시스템의 블록다이어그램을 보여준다.

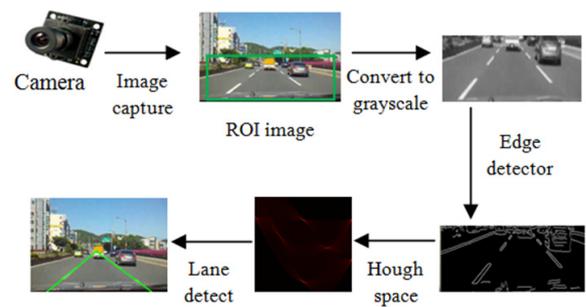


그림 1. 차선인식 시스템의 블록다이어그램
Fig. 1. Block diagram of lane detection.

먼저 ROI(Region of interest) 기법을 이용하여 입력 영상의 하위 영역만을 알고리즘 적용을 위한 부분으로 설정하는데 이는 처리 속도를 증가시키기 위해서 사용했으며, ROI를 통해 얻은 서브 이미지는 흑백 영상으로 변환된다. 두 번째는 일반적인 흑백 이미지에서 엣지를 검출할 경우 노이즈가 생기기 때문에 이를 가우시안 필터를 이용하여 노이즈를 감소시키며 그 뒤 케니 엣지 알고리즘을 이용하여 엣지를 검출하게 된다. 마지막으로 허프 영역에 accumulator cells를 최적화하는 기법을 사용하여 특정 구간에서의 평균적인 각도 및 거리를 계산하여 차선의 경계를 검출하게 된다.

1. 입력 도로의 영상

첫 번째 단계는 그림 2처럼 차량의 내부에 장착된 블랙박스 또는 안드로이드 기기의 카메라를 통해서 차선

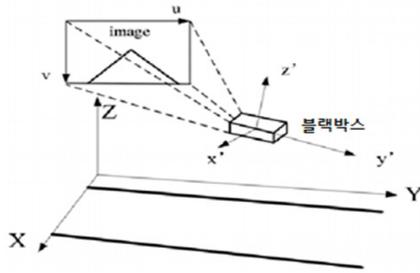


그림 2. 입력영상의 차선 모델
Fig. 2. Load model of input image.

의 영상을 받아들이는 단계이다.

차선 모델에서 X-Y-Z는 도로의 좌표이며, x'-y'-z'는 입력 이미지의 좌표이다. 그리고 영상에서 v는 높이, u는 넓이로 받아들이며, 차선은 실선 또는 점선으로 표시된다.

2. ROI(Region of Interest)

차선을 검출하는 알고리즘은 나무, 건물 등에 의해 생기는 그림자에 영향을 받지 말아야 한다. 따라서 전체적인 도로의 영상은 기타 사물(하늘, 건물, 나무 등)을 검출하지 않고 연산 시간을 줄이며 알고리즘의 성능을 향상시키기 위해 ROI 검출 기법을 사용해서 차선 경계 검출에 필요한 부분만 남기고 차선 위쪽의 필요하지 않은 요소들은 잘라버린다. 전체 입력 영상의 좌표 I(v, u)는 ROI를 통해서 ROI(x, y)의 좌표로 변환되며 그림 3과 같이 서브 이미지를 만든다. 서브 이미지 내에 L1은 도로의 왼쪽 차선 경계이며 L2는 오른쪽 차선이 경계이다.

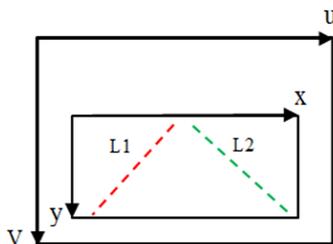


그림 3. ROI 방법을 이용한 Sub-image
Fig. 3. Sub-image using the ROI method.

3. 엣지 검출

세 번째 단계는 차선의 경계를 인식 하기 위한 엣지를 검출하는 단계이다. 먼저 엣지를 검출하기 전에

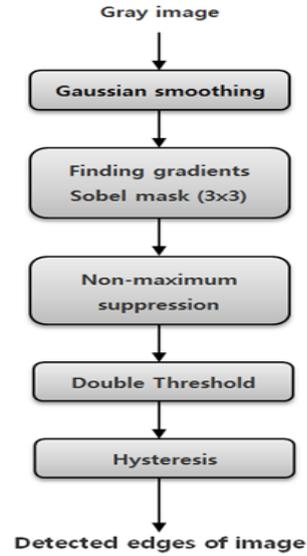


그림 4. 케니 엣지 검출 알고리즘의 블록 다이어그램
Fig. 4. Block diagram of Canny edge detection algorithm.

RGB 이미지가 흑백 이미지로 변환된다.

엣지는 영상의 밝기가 낮은 값에서 높은 값으로, 또는 이와 반대로 변하는 지점에 존재하는 부분을 나타낸다. 엣지를 검출하는 다양한 검출 기법들이 있는데 본 논문에서는 차선의 경계를 검출하기 위해서 케니 엣지 검출을 이용한다.

일반적으로 엣지 검출기는 노이즈에 매우 민감한 특성을 가지고 있다. 이러한 특성에 따라 잘못된 엣지의 검출을 방지하기 위해 개발 된 알고리즘이 케니 엣지 검출기이다. 이 알고리즘은 일반적으로 5단계로 진행이

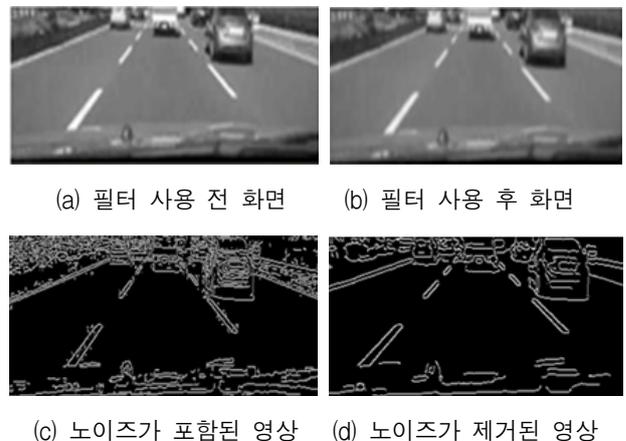


그림 5. 가우시안 필터를 사용한 케니 엣지 검출
Fig. 5. Canny edge detection Using Gaussian smoothing.

되며 블록다이어그램은 그림 4와 같다.

영상에서 노이즈의 영향을 최소화하기 위한 방법으로 가우시안 smoothing 필터를 수행한다.

그림 5(a)는 노이즈를 가지고 있는 입력 영상을 보여주며 이 영상을 가우시안 필터를 사용하지 않고 일반적인 엣지 검출과정을 수행하게 되면 그림 5(c)와 같은 노이즈를 포함한 차선을 나타낸다. 그림 5(b)는 입력 영상에 가우시안 필터를 사용한 후의 화면이고, 이 결과 값에 엣지 검출 과정을 수행하게 되면 그림 5(d)에서처럼 노이즈가 제거된 차선을 보여 준다.

4. 허프 변환 방법

허프 변환은 영상내의 선 기울기나 교점 등의 매개 변수로 표현될 수 있는 경우에 그것을 검출 해 내는데 좋은 효과를 보이는 방법이다.

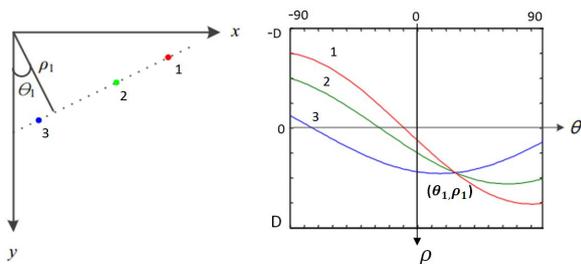
허프 변환 알고리즘은 가능한 모든 각도인($\pi/180$ 단 계별 크기로)와 해당(반올림) 값을 계산해 반복하는 과정에서 해당 점을 지나는 모든 선을 식별할 수 있으며, 지나는 직선을 식(1)과 같은 형식의 방정식을 사용해 직선을 표현한 것이다.

$$x\cos\theta + y\sin\theta = \rho \quad (1)$$

허프 변환은 θ, ρ 의 파라미터를 생성하고 허프 영역에서 정현과 곡선으로 나타내어지며 영상 좌표 (x, y) 엣지 지점에 의해 변환된다.

그림 6 및 식(1)에서 사용된 변수들의 기하학적 해석은 다음과 같다. ρ 는 원점에서 직선까지의 길이이며, θ 는 ρ 와 x 축이 이루는 각이다.

x와 y는 영상에서의 가로와 세로축을 의미한다. 수평 방향의 선은 $\theta = 0^\circ$ 이고 ρ 는 양의 x절편과 같다. 수직 방향의 선은 $\theta = 90^\circ$ 이고 ρ 는 양의 y절편, 혹은



(a) image space (b) Hough space

그림 6. 직선의 일반적인 표현방법

Fig. 6. Regular representation of a straight line.

$\theta = -90^\circ$ 이고 ρ 는 음의 y절편과 같다.

이 함수를 통해서 영상의 x, y 평면에서 한 점은 허프 영역 (ρ, θ 평면)에서 곡선으로 표시된다. 따라서 이를 통해 직선을 찾기 위해서는 허프 영역에서 선이 모이는 점을 찾으며 그 점을 accumulator cells (array)라고 부른다.

일반적으로 모든 θ 값을 계산하여 영상의 xy평면에 있는 모든 직선들을 인식 하기 때문에 차선의 경계 및 다른 직선들도 검출하게 되어 결과는 악화된 상태로 나타난다. 따라서 이러한 문제를 결정하기 위해 허프 영역에 Accumulator cells를 최적화한 알고리즘을 제안했다.

5. Accumulator cells를 최적화한 제안하는 알고리즘

본 논문에서는 허프 변환을 이용하여 차선의 경계들을 왼쪽과 오른쪽으로 두 부분에 나누어서 검출한다. 다음 그림 8에 제안하는 알고리즘의 순서도를 나타내었다. 이 알고리즘의 기본 개념은 허프 영역에 accumulator cells를 최적화한 방법이다. 먼저 차선의 경계를 그림 7처럼 xy평면에 적용한다.

그림 7은 도로의 입력 영상의 2D 평면에서 왼쪽 차선 경계(left lane boundary) 및 오른쪽 차선 경계(right lane boundary)를 통해 두 개의 차선으로 나타냈다. ρ_1 은 원점과 왼쪽 차선 경계 간의 수직 거리이며 θ_1 는 y 축과 왼쪽 차선 경계 간의 각도이며, 이는 차량의 정면에 해당하는 y축과 차량 간의 간격이다. 비슷하게 ρ_2 은 오른쪽 차선과의 수직 거리이며 θ_2 는 y축과 수직으

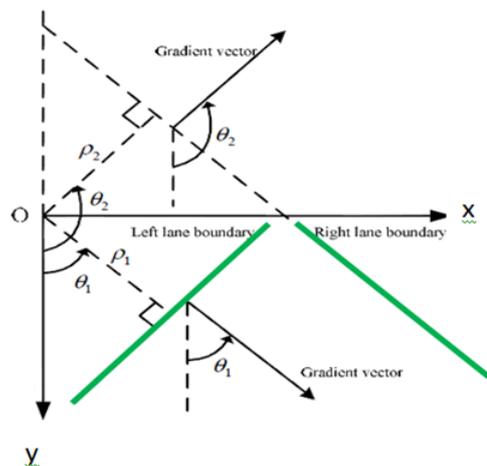


그림 7. 영상 좌표에서의 차선 위치

Fig. 7. Lane position on image coordinates.

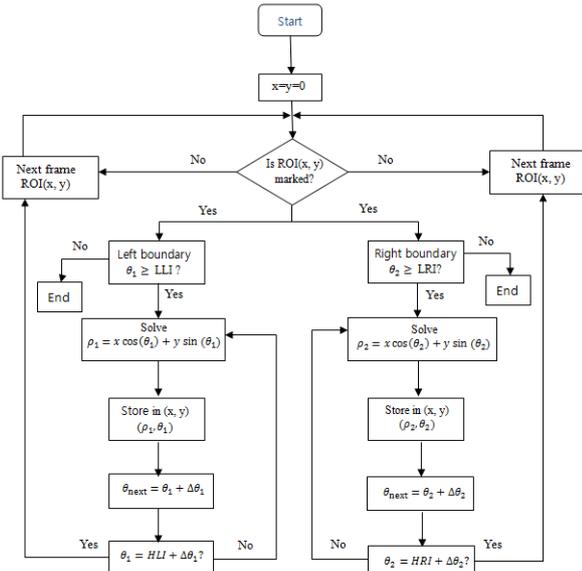


그림 8. 제안하는 알고리즘에 대한 허프 변환의 순서도
 Fig. 8. Flow chart of proposed Hough Transform algorithm.

로 만든 오른쪽 차선 간의 각도이다. 여기서 Gradient vector은 항상 출력 될 차선과 90°의 각도를 유지 해야 한다.

그림 8은 제안하는 허프 변환 알고리즘에 대한 순서도를 나타낸 것이다. 먼저 영상의 한 프레임이 입력으로 들어와 ROI에 이미지가 읽힐 경우 ROI에서 각각 왼쪽 차선과 오른쪽 차선을 검출하는 알고리즘이 병렬적으로 이루어진다. 왼쪽 차선 검출의 경우 먼저 왼쪽 영역에서 θ_1 이 LLI보다 클 경우 식 (1)을 통해 ρ_1 을 계산한다. 그 뒤 ρ_1 과 θ_1 (x,y)영역에 저장을 한 뒤 해당하는 검출 차선을 출력한다. 또한 다음 프레임의 theta 값인 θ_{next} 는 θ_1 에서 $\Delta\theta$ 만큼 변화한 값인데 여기서 θ_1 가 HLI+ $\Delta\theta$ 과 같을 경우 다음 프레임이 알고리즘에 적용이 되며, 같지 않을 경우 ρ_1 을 연산하는 부분으로 feedback이 된다. 같은 알고리즘으로 오른쪽 영역에서 우측 차선이 검출이 되며 검출 된 양 차선이 한 프레임에 동시에 출력이 된다.

그림 9는 허프 영역 평면의 accumulator cells이다. Hough space에 (ρ_{min}, ρ_{max}) 와 $(\theta_{min}, \theta_{max})$ 는 매개변수 값들의 예상 범위이다. 대개 이들 값들의 범위는 $-90^\circ \leq \theta \leq 90^\circ$ 와 $-D \leq \rho \leq D$ 이며, 여기서 D는 x-y 공간에서 원점을 기준으로 맞은편 꼭지점까지의 거리를 나타낸다.

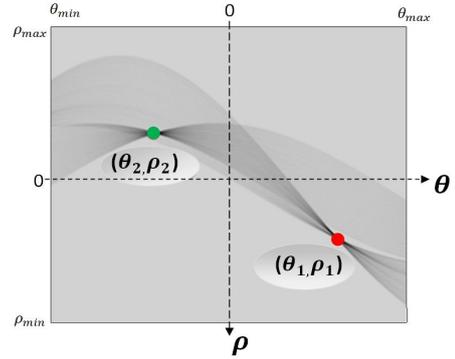


그림 9. 허프 영역 평면의 accumulator cells
 Fig. 9. Accumulator cells in the Hough space.

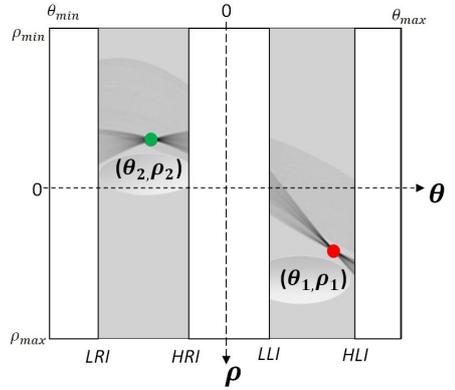
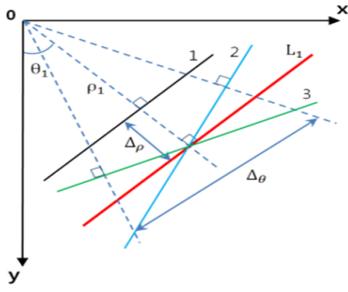


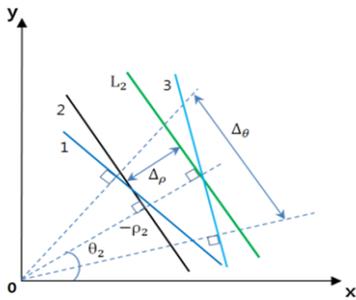
그림 10. Accumulator cells를 최적화한 제안하는 알고리즘
 Fig. 10. An optimized Accumulator cells algorithm proposed in this paper.

제안하는 알고리즘은 그림 10과 같이 왼쪽 차선 경계의 경우 LLI(왼쪽 하단 지점)부터 HLI(왼쪽 상단 지점)까지 있는 theta 값을 계산한다. 그 부분에 accumulator cells만 추출된 직선들을 선택하였다.

그리고 오른쪽 차선의 경계도 같은 LRI(오른쪽 하단 지점)부터 HRI(오른쪽 상단 지점)까지 있는 부분에 추출된 직선들을 선택하였다. 그림 11의 θ, ρ 평면을 accumulator cells에서 왼쪽의 $LLI \leq \theta_1 \leq HLI$ 은 $\Delta\theta$ 로 추출하면 왼쪽 차선에 n개의 직선들이 인식되게 된다[그림 11(a)]. 또한 $LRI \leq \theta_2 \leq HRI$ 은 $\Delta\theta_2$ 로 추출하면 다른 오른쪽 차선에 m개의 직선들이 인식되게 된다[그림 11(b)]. 기존의 Hough Transform의 경우 ROI에서 다수의 라인들이 인식이 되어 정확한 차선이 인식되는데 어려움이 있었는데, 본 논문에서 제안한 알고리즘의 개선된 accumulator cells을 통해 필요한 차선만을 검출하여 출력할 수 있다.



(a) 왼쪽 차선 경계



(b) 오른쪽 차선 경계

그림 11. 각각 다른 차선경계에서 발견된 다이어그램
Fig. 11. Diagrams found in each lane boundary.

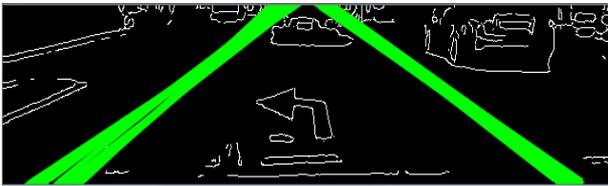


그림 12. 제안한 알고리즘을 사용한 영상.
Fig. 12. Example an image used to proposed algorithm.

그림 12는 제안한 알고리즘을 가우시안 필터를 통한 차선 이미지에 적용한 그림이다.

그림 12를 수식적으로 설명하게 되면, 제안하는 알고리즘의 차선경계는 다수의 차선이 필요하지 않고, 양쪽 차선 하나씩만 필요하기 때문에 식(2), (3)와 같이 θ 의 평균값을 계산한다.

$$\theta_1 = \frac{1}{n} \sum \Delta\theta_1; \quad LLI \leq \theta_1 \leq HLI \quad (2)$$

$$\theta_2 = \frac{1}{m} \sum \Delta\theta_2; \quad LRI \leq \theta_2 \leq HRI \quad (3)$$

계산의 결과를 식(2), (3)의 θ 값 대신에 계산된 θ 값으로 다시 바꾸면 다음 식(6), (7)과 같다. 여기서 ρ_1 , ρ_2 들은 축의 범위는 $\pm\sqrt{2} D$ 이다.

$$\rho_1 = x \cos(\theta_1) + y \sin(\theta_1) \quad (4)$$

$$\rho_2 = x \cos(\theta_2) + y \sin(\theta_2) \quad (5)$$

θ_1, ρ_1 값으로 왼쪽 차선경계를 인식하고 θ_2, ρ_2 값으로 오른쪽 차선경계를 인식하며 다음 장에서 결과를 보여 준다.

III. 차선검출 시스템 개발

본 논문에서 제안한 차선검출 시스템을 소프트웨어 검증 및 안드로이드 기반의 하드웨어 검증 했으며, 차선 검출 시스템을 실시간으로 개발하였다.

1. OpenCV를 이용한 Software 검증

첫 번째로 차선검출 알고리즘은 OpenCV library를 이용하여 구현했다. C/C++ 언어로 작성한 알고리즘은 그림 13과 같이 나타 낼 수 있다.

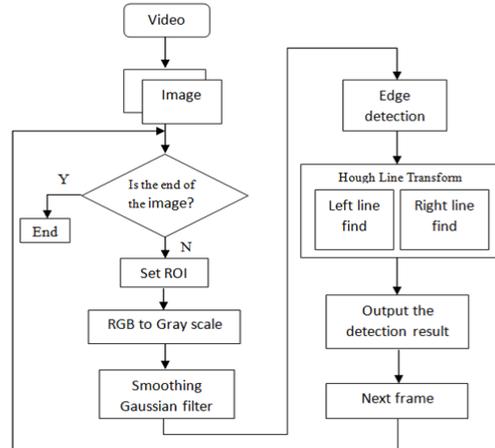


그림 13. 제안하는 차선검출 시스템의 OpenCV 순서도
Fig. 13. OpenCV flow chart of the proposed lane detection system.

본 알고리즘은 위에서 설명한 것과 같은 플로우를 통해 알고리즘이 진행 되었으며, OpenCV를 통한 소프트웨어 검증을 하였다.

2. 차선검출 시스템 개발

제안하는 차선검출 알고리즘 검증의 두 번째 단계는 안드로이드 SDK로 개발해서 안드로이드용 기기로 실

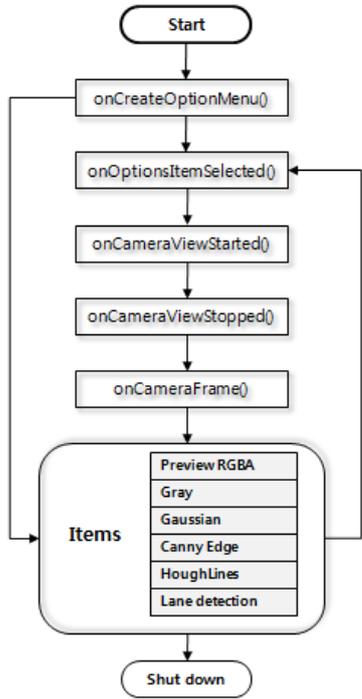


그림 14. 제안하는 차선검출 시스템의 JavaCV 순서도
Fig. 14. JavaCV flow chart of the proposed lane detection system.

제 도로면에서 차선검출 하는 것이다.

안드로이드 SDK는 일반적으로 Java 언어로 구현되어 있지만 소프트웨어 검증에서의 알고리즘은 OpenCV C/C++ 언어로 구현되어 있기 때문에 C언어로 구현된 OpenCV 함수를 안드로이드 상에서 개발하기 위해서 JavaCV로 변환하였다.

차선검출 시스템을 Eclipse 개발환경에서 JavaCV로 만들기 위한 순서도는 그림 14와 같이 나타 낼 수 있으며, 이와 같은 순서를 통해서 .apk 파일 생성을 수행했다.

그림 14에서 onCreateOptionsMenu 의 경우 이 메소드를 호출하게 된 프로그램은 시스템의 Menu 객체를 넘겨받게 되고, 여기에 사용자에게 보여 준 item 추가한다. 다음 onOptionsItemSelected에서 이 메소드는 사용자가 선택한 메뉴 항목을 처리한다. OnCameraViewStarted는 카메라 미리보기가 시작되었을 때 메소드가 호출된다. 이 메소드가 호출된 후 Camera Frame()가 callback 메소드를 통해 사용자에게 전달되기 시작된다. 그 뒤 onCameraViewStopped 메소드의 경우 카메라 미리보기가 어떤 이유로 중지 된 경

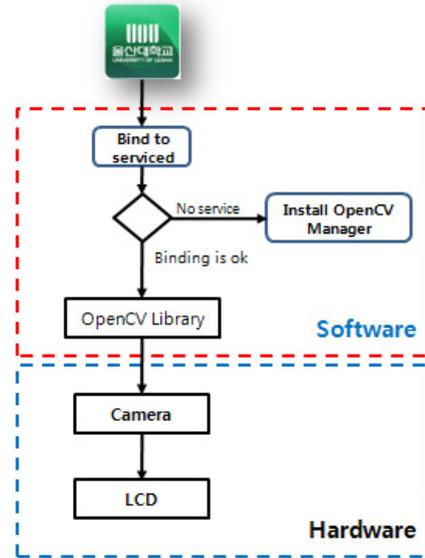


그림 15. 차선 경계 검출 시스템의 개발 구조
Fig. 15. The system architecture of lane detection.

우 이 메소드가 호출된 후 어떤 Camera Frame() callback를 통해 전달되지 않는다. OnCameraFrame 에서는 프레임의 전달을 수행해야하는 경우가 메소드가 호출된다. 반환 값은 화면에 표시 할 필요가 수정 된 frame이다. 여기 입력 프레임의 사이즈를 준다. 마지막으로 Items는 제안한 알고리즘을 JavaCV로 처리한다.

안드로이드 개발 환경을 갖추고 .apk 파일을 만들기 위한 구성 요소로는 기본적으로 JDK, Eclipse, 안드로이드 SDK의 3가지를 이용하며 Android SDK는 안드로이드 어플리케이션의 개발하기 위해서 필요한 모든 도구와 API를 갖고 있다. Eclipse는 Java 개발 분야에서 널리 사용되는 OpenCV IDE (Integrated Development Enviroment)로서 안드로이드의 개발환경으로 사용한다.

안드로이드 플랫폼은 크게 4가지 부분으로 나뉘며, 그 구조는 그림 15처럼 나타낼 수 있다. 첫 번째로 차선 검출 어플리케이션을 학교의 로고로 생성하는 단계이며, 두 번째로는 이를 OpenCV Manager를 설치해서 Library를 호출 하는 단계이다. 세 번째는 카메라를 통해 Library의 header파일을 인식하는 단계이고, 마지막으로 이를 LCD 화면으로 보여주는 단계이다.

IV. 실험 결과

1. OpenCV를 이용한 소프트웨어 검증 결과

제안한 알고리즘을 소프트웨어로 검증하기 위해 도로상에서 촬영한 초당 27 프레임 가량의 640 x 352 영상을 가지고 실험하였다. Intel(R) Core(TM)2 Quad Q8400 CPU @ 2.66GHz, 2GB RAM, Windows 7 Ultimate PC를 사용하였으며 Visual Studio 2010과 영상처리 공개 라이브러리인 OpenCV를 사용하여 구현하였다.

결과는 도로의 상황을 3가지 조건으로 나누어 진행

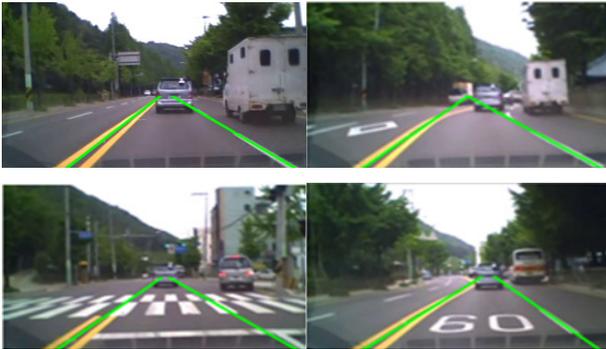


그림 16. 비디오 클립1의 결과
Fig. 16. Result of video Clip1.

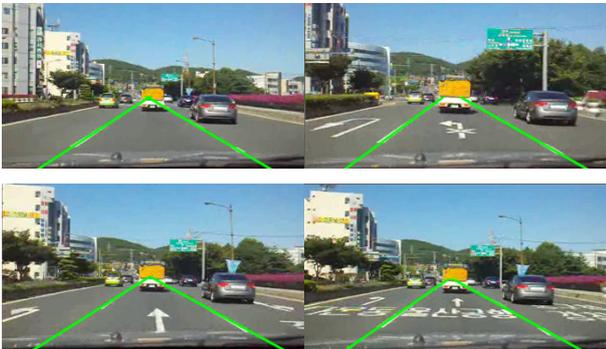


그림 17. 비디오 클립2 결과
Fig. 17. Result of video Clip2.

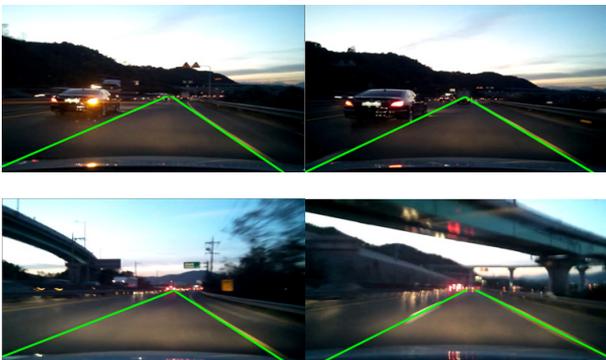


그림 18. 비디오 클립3 결과
Fig. 18. Result of Video Clip3.

했다. 그림 16, 17, 18에 그 결과를 보여주며, 표 1에서 성능을 나타내었다.

- 클립 1은 일반적인 도로의 황색 및 흰색 차선, 그리고 건물목이 있는 경우의 성공적인 차선인식을 보여준다.
- 클립 2는 차선이 많은 도로, 표지판, 도로위의 글자 및 화살표시 등의 복잡한 도로의 상황에서 성공적인 차선인식을 보여준다.
- 클립 3은 해질녘에 어두운 도로의 상황 및 고속 주행 상태에서 성공적인 차선인식을 보여준다.

고속 주행 및 야간에는 블랙박스 영상의 흔들림 현상과 미동 그리고 인공광 등에 영향을 받기 때문에 인식률이 낮게 나올 수 있지만 제안한 알고리즘을 적용한 클립 3영상의 경우에는 97.3% 을 높은 효율을 가지는 것을 확인 할 수가 있다.

결과적으로 평균적으로 인식률이 93.8%의 높은 효율을 가질 뿐만 아니라 저녁 및 고속주행(클립 3)에는 높은 효율을 가질 수 있는 것을 아래 표 1에서 확인 할 수 있다.

표 1. 제안하는 알고리즘의 s/w 검증결과
Table 1. Results of the proposed algorithm.

| Clip | Total frame | Detected frame | Correct rate | Frame/sec |
|-------|-------------|----------------|--------------|-----------|
| 1 | 1280 | 1170 | 91.4% | 27.0 |
| 2 | 1000 | 928 | 92.8% | 27.0 |
| 3 | 738 | 718 | 97.3% | 28.0 |
| Total | 3018 | 2816 | 93.8% | 27.3 |

2. 차선검출 시스템 개발 결과

차선검출 시스템을 개발하기 위해서 도로에서 실제로 실험할 필요가 있기 때문에 안드로이드용 기기로 개발하여 검증 하였다. 실험에 사용한 스마트폰의 하드웨어 사양의 세부 사항은 다음과 같다.

Android platform : Samsung galaxy S3

Version : Android 4.0.4 Ice Cream Sandwich (API level 15)

Camera : 13 Mega pixel, autofocus, LED flash

CPU : Quad-Core 1.6 GHz
Internal Storage : ROM 16 Gb, RAM 2 Gb

검증 과정에서의 신뢰도를 위해 두 대의 스마트폰을 동시에 사용하여 한 스마트 폰에서는 어플리케이션을



그림 19. 주간 환경에서 검증 과정을 위한 테스트 환경.
Fig. 19. Test environment for qualification process by day.



그림 20. 야간 환경에서 검증 과정을 위한 테스트 환경.
Fig. 20. Test environment for qualification process by night.



그림 21. 주간 도시 도로 상황에서의 (a) S/W 테스트 결과 (b) H/W 테스트 결과 (클립5)
Fig. 21. Result of on the city road by day.

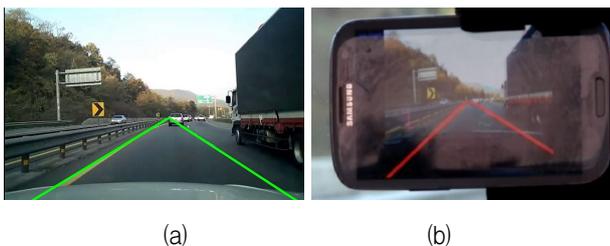


그림 22. 주간 고속 도로 상황에서의 (a) S/W 테스트 결과 (b) H/W 테스트 결과 (클립6)
Fig. 22. Result of on the highway road by day.

통한 H/W 검증을 할 수 있도록 하였으며 다른 스마트폰을 통해 동일한 영상을 촬영하여 저장된 데이터를 통해 S/W 검증을 하여 동일한 조건에서 두 가지의 검증을 동시에 할 수 있도록 환경을 구성하였다.

그림 19와 20은 각각 주간과 야간에서 테스트를 가질 때 구성하였던 실험 환경이다.

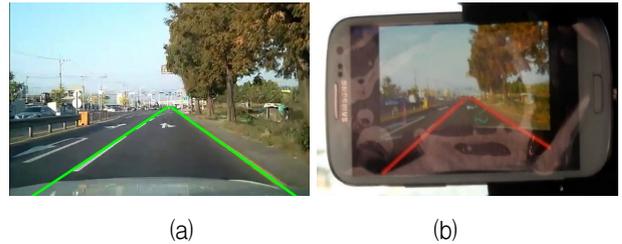


그림 23. 주간 일반 도로 상황에서의 (a) S/W 테스트 결과 (b) H/W 테스트 결과 (클립7)
Fig. 23. Result of on the simple road by day.

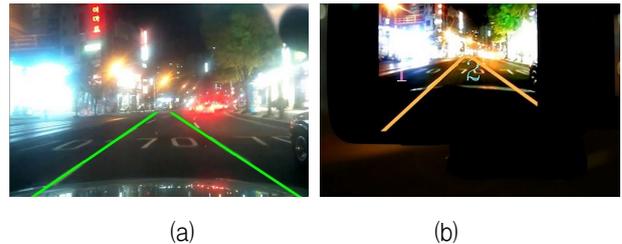


그림 24. 야간 도시 도로 상황에서의 (a) S/W 테스트 결과 (b) H/W 테스트 결과 (클립8)
Fig. 24. Result of on the city road by night.

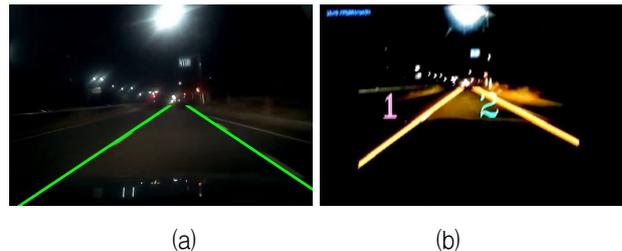


그림 25. 야간 고속 도로 상황에서의 (a) S/W 테스트 결과 (b) H/W 테스트 결과 (클립9)
Fig. 25. Result of on the highway road by night.



그림 26. 야간 일반 도로 상황에서의 (a) S/W 테스트 결과 (b) H/W 테스트 결과 (클립10)
Fig. 26. Result of on the simple road by night.

표 2. 제안하는 알고리즘의 S/W 및 H/W 검증결과
Table 2. Results of the proposed algorithm

| clip | | Total frame | Detected frame | Correct rate | Frame/sec |
|-------|-----|-------------|----------------|---------------|-----------|
| 5 | S/W | 1200 | 1103 | 91.9% | 27 |
| | H/W | 1200 | 705 | 58.75% | 27 |
| 6 | S/W | 1525 | 1423 | 93.3% | 28 |
| | H/W | 1525 | 1015 | 66.5% | 28 |
| 7 | S/W | 925 | 865 | 93.5% | 27 |
| | H/W | 925 | 633 | 68.4% | 27 |
| 8 | S/W | 1403 | 1295 | 92.3% | 27 |
| | H/W | 1403 | 905 | 64.5% | 27 |
| 9 | S/W | 1300 | 1203 | 93.8% | 28 |
| | H/W | 1300 | 1025 | 78.8% | 28 |
| 10 | S/W | 1705 | 1601 | 93.9% | 27 |
| | H/W | 1705 | 1304 | 76.4% | 27 |
| Total | S/W | 8058 | 7501 | 93.1% | 27.3 |
| | H/W | 8058 | 5551.1 | 68.89% | 27.3 |

- 클립 5는 주간 시내 도로 상황에서 성공적인 실시간 차선검출을 보여준다.
- 클립 6은 주간 고속 도로 상황에서 성공적인 실시간 차선검출을 보여준다.
- 클립 7은 주간 일반 도로 상황에서 성공적인 실시간 차선검출을 보여준다.
- 클립 8는 야간 시내 도로 상황에서 성공적인 실시간 차선검출을 보여준다.
- 클립 9은 야간 고속 도로 상황에서 성공적인 실시간 차선검출을 보여준다.
- 클립 10은 야간 일반 도로 상황에서 성공적인 실시간 차선검출을 보여준다.

V. 결 론

본 논문에서는 Accumulator cells를 최적화함으로써 고효율의 차선인식 및 연산시간을 줄이는 차선 검출 알고리즘을 제안하였다.

제안한 알고리즘은 OpenCV 기반 소프트웨어 검증시 약 93.1%의 성공적인 차선인식률을 보였다. 안드로이드 기반의 실시간 하드웨어 검증 결과 비슷한 환경에서 68.89% 가량의 차선인식률을 보였다. 현재 안드로이드 기반의 차선검출 시스템이 존재 하지 않은 상황이기 때문에 성공적인 차선인식률이다.

난반사 및 저조도의 우천 시 상황에서 차선인식률을 높이기 위해 알고리즘을 보완할 것이며, 정상적인 차선

변경 및 차선 이탈을 구분하는 차선검출경고시스템을 개발할 계획이다.

REFERENCES

- [1] Ji-Hue Bae and Jae-Bok Song, "Monocular Vision Based Lane Detection Using Segmented Regions from Edge Information," 2011 IEEE, 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), pp. 499-502, Incheon, Korea, Nov 2011.
- [2] Y.Wang, E. K. Teoh and D. Shen, "Lane detection and tracking using B-snake," Image and Vision Computing, Vol. 22, pp. 269-280. April 2004.
- [3] G. Keyou, Li Na and Z. Mo, "Lane detection Based on the Random Sample Consensus," Information Technology, Computer Engineering and Management Sciences (ICM), 2011 International Con. on Vol 3, pp. 38-41, Nanjing, Jiangsu, Sept. 2011.
- [4] T. Trung-Tien, Hyo-Moon Cho, Jong-Hwa Lee and Sang-Bock Cho, "A Novel Method for Lane Marking Detection in Various Conditions", 26th International Technical Conference on Circuits/Systems, Computers and Communications, (ITC-CSCC), pp. 347-349. Gyeongju, Korea, June 2011.
- [5] X. Wang, W. Yongzhong and W. Chenglin, "Robust Lane Detection Based on Gradient-Paris Constraint," Control Confrance (CCC), 2011 30th Chinese, pp. 3181-31850. Yantai, China, July 2011.
- [6] W. Phueakjeen, N. Jindapetch, L. Kuburat and N.Suvanvorn, "A Study of the Edge Detection for Road Lane," Electrical Engineering/ Electronics, Computer, Telecommunication and Information Technology (ECTI-CON), 2011 8th IEEE International Conf. pp. 995-998, Khon Kaen, May 2011.
- [7] G. Christos and V. Nikolaos "Real-Time Canny Edge Detection Parallel Implementation for FPGAs," Electronics, Circuits, and Systems (ICECS), 2010 17th IEEE International Conf. on. pp. 499-502. Athens, Dec. 2010.
- [8] Q.Lin, Y.Han and H. Hahn, "Real-time Lane Detection Based on Extended Edge-linking Algorithm," Computer Research and Development, Second International Conference on.

pp. 725-730. Kuala Lumpur, May 2010.

[9] 배찬수, 이종화, 조상복, “형태학과 색상 정보를 이용한 차선 인식 알고리즘,” 전자공학회, 전자공학 회논문지-SD, 제48권 SD편, 제6호, 15-24쪽, 2011년 6월.

[10] Fatemeh Mazrouei Sebdani, Hossein Pourghassem. “A Robust and Real-time Road Line Extraction Algorithm using Hough Transform in Intelligent Transportation System Application,” Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on. Vol. 3, pp. 256-260, Zhangjiajie, May 2012.

— 저 자 소 개 —



척트바타르 엘덴토야(학생회원)
 2010년 Mongolian university of science and technology 학사 졸업.
 2013년 울산대학교 전기전자 공학부 석사 졸업.

<주관심분야 : 영상 압축 및 처리, 지능형 자동차, VLSI 및 SoC설계>



조 재 현(학생회원)
 2013년 울산대학교 전기전자 공학부 학사 졸업.
 2014년 현재 울산대학교 전기전자 공학부 석사과정.

<주관심분야 : 영상 압축 및 처리, 지능형 자동차, CMOS VLSI 및 SoC설계, 아날로그 설계>



장 영 민(학생회원)
 2012년 울산대학교 전기전자 공학부 학사 졸업.
 2014년 울산대학교 전기전자 공학부 석사 졸업예정.

<주관심분야 : 영상 압축 및 처리, 지능형 자동차, CMOS VLSI 및 SoC설계, 아날로그 설계>



조 상 복(평생회원)
 1979년 한양대학교 전자공학과 학사 졸업.
 1981년 한양대학교 전자공학과 석사 졸업.
 1985년 한양대학교 전자공학과 박사 졸업.

1994년~1995년 Univ. of Texas, Austin 교환교수
 2003년~2004년 Univ. of California, San Diego 교환교수
 2011년~현재 울산대학교 e-vehicle 연구센터장
 2013년~현재 울산대학교 자동차·조선 전자융합 기술사업단장(BK21-PLUS)
 <주관심분야 : SoC/VLSI 설계 및 테스트, 자동차 전장시스템 설계, 영상처리 회로 설계 및 제작, 머신비전 시스템 개발, 초고집적 메모리 설계>