

V-ROI를 이용한 고효율 실시간 차선 인식 알고리즘

Efficient Real-time Lane Detection Algorithm Using V-ROI

Ding Dajun* , 이찬호**

Ding Dajun* , Chanho Lee**

Abstract

Information technology improves convenience, safety, and performance of automobiles. Recently, a lot of algorithms are studied to provide safety and environment information for driving, and lane detection algorithm is one of them. In this paper, we propose a lane detection algorithm that reduces the amount of calculation by reducing region of interest (ROI) after preprocessing. The proposed algorithm reduces the area of ROI a lot by determining the candidate regions near lane boundaries as V-ROI so that the amount of calculation is reduced. In addition, the amount of calculation can be maintained almost the same regardless of the resolutions of the input images by compressing the images since the lane detection algorithm does not require high resolution. The proposed algorithm is implemented using C++ and OpenCV library and is verified to work at 30 fps for realtime operation.

요 약

자동차가 IT 기술과 융합되면서 편의성과 안전성 그리고 성능이 좋아지고 있다. 이와 관련하여 최근 자동차의 주행 안전 및 주변 환경과 관련된 정보를 제공하기 위한 많은 알고리즘이 연구되고 있으며 차선 인식 또한 그 중 하나이다. 본 논문에서는 입력된 영상에서 차선 경계선을 인식한 뒤 ROI를 경계선 주변으로 제한하여 연산량을 줄이는 알고리즘을 제안한다. 제안된 알고리즘에서는 선처리 과정을 통해 차선 경계선으로 추정되는 영역의 주변만을 ROI로 지정하는 V-ROI를 이용하여 연산 영역을 줄이고 이를 통해 연산량과 연산 시간을 줄인다. 또한 차선 인식의 경우 고해상도의 영상이 필요하지 않으므로 입력 영상을 축소하여 차선 인식 알고리즘을 적용하는 방법을 통하여 영상의 해상도에 관계없이 연산량을 비슷하게 유지할 수 있다. 제안한 알고리즘을 C++와 OpenCV 라이브러리를 이용하여 구현하였으며 초당 30 프레임 이상을 처리하는 실시간 동작을 확인하였다.

Key words : pattern recognition, ROI, smart car, OpenCV, lane detection

* ding.dajun@gmail.com, 02-825-8108, Dept. of Electronic Engineering, Soongsil University

★ chlee@soongsil.ac.kr, 02-820-0710, School of Electronic Engineering, Soongsil University

※ This work was sponsored by Industrial Strategic Technology Development Program funded by the Ministry of Knowledge Economy (MKE, Korea) (10039188, Development of multimedia convergence programmable platform for smart vehicles). The EDA tools are supported by IDEC.

Manuscript received Oct. 3, 2012; revised Nov. 14, 2012 ; accepted Nov 19. 2012

1. 서론

자동차에 IT 기술이 접목되어 지능형 자동차 개발이 활성화되고 있고 이에 따라 자동차의 편의성과 안전성 그리고 성능이 크게 개선되고 있고 이를 위한 연구 또한 활발히 진행되고 있다. 이러한 연구는 자동차 주행과 관련하여 성능 및 안전성을 개선하는 연구와 차량의 상태 및 주변 환경 등의 정보를 제공하여 편의성과 안전성을 높이는 분야로 나누어지는데 후자의 경우 카메라를 통한 영상을 분석하여 주변 상

황을 인식하는 연구가 중요한 부분을 차지한다.

그중 하나인 차선 인식은 차선에서 차량의 위치와 방향을 추정하는데 이용 가능하다. 이를 이용하여 자동차 전용도로 등에서 차량의 차선 이탈을 감지하여 운전자에게 알려주고 모든 주행 상황에서 차선을 인식하여 자동 운전 시스템에서 차량이 차선을 이탈하지 않고 주행하기 위한 정보를 제공한다[1-9]. 또한 다른 차량의 위치와 장애물에 대한 정보를 제공하여 장애물 회피 시스템에서도 이용될 수 있다. 차선 인식을 위해 카메라, 선 인식 센서, 레이저 탐지, GPS 등의 다양한 센서를 이용한 방법이 연구되었는데 이중에서 카메라를 이용한 방식이 가장 광범위하게 특정 조건에 구애받지 않고 사용 가능하다[2]. 카메라를 이용한 영상 인식 방식은 선 인식이나 레이저 탐지 방식처럼 차량 외부 또는 도로에 별도의 장치를 필요로 하지 않고 GPS 신호를 이용하는 방식이 요구하는 축적된 지도 정보 등이 없어도 정확한 위치 정보를 제공할 수 있다.

영상 인식을 기반으로 한 차선 인식 알고리즘에서 입력된 영상으로부터 차선 경계선 추출은 가장 핵심적인 동작으로 차량이 차선 안에서 좌우로 조금씩 움직이고 차선도 항상 직선을 유지하지는 않아 하나의 특징 추출로는 인식이 어렵다. 따라서 적절한 임계치 설정, 가장자리 탐지와 선 탐지 등의 방법을 통해 실시간으로 이루어진다. 패턴의 가장자리를 인식하는 방법은 차선 경계선처럼 두꺼운 분할선 인식에 적합하고 야간 주행을 돕기 위한 반사판에 의한 영향을 보상하는 데도 이용 가능하다[2]. 차선 인식을 위해 Xu Zhe[3] 등은 Canny 방식과 Progressive Probabilistic Hough Transform(PPHT)을 사용했고 S. Zhou[4] 등은 기하학적 모델과 Gabor 필터를 이용한 방식을 사용하였다. C. Jung[5] 등은 가장자리 탐지와 최소 제곱 각 예측 (least squares angular estimation), 그리고 Hough 변환(Hough Transform)을 사용하여 차선을 인식하였다. 실험 결과에 따르면 대체로 차선을 잘 인식하나 도로의 그늘진 곳이나 간섭이 있는 경우에는 문제가 있다.

고속도로의 중간 차선은 매우 환경이 좋은 편이지만 가장자리 차선이나 도심의 시가지 등 도로 환경이 매우 복잡한 경우에는 관심 영역(Region of Interest, ROI)의 선택이 차선 인식에 중요한 요소로 작용한다. ROI의 선택은 인식율과 인식 성능 모두에 중요한 영향을 미친다. Xe Zhu 등은 ROI를 설치된 카메라의 높이와 피치 각에 따라 설정하였는데 가로로 가운데의 1/3 또는 1/2의 영역을 선택하였다[3]. 선택된 영역은 직사각형 모양으로 프레임 간에 변경이 안 되어 고정된 영역을 가리키게 된다. V. Gaikwad 등은 카

메라 입력 영상의 아래 부분을 ROI로 선택하였는데 이를 다시 오른쪽과 왼쪽 영역으로 나누었다[6]. S. Zhou 등은 카메라와 도로 좌표의 관계에 따라 영상을 분할하였다[4]. 차량 앞의 도로는 10m를 기준으로 가까운 영역과 먼 영역으로 나누어 가까운 영역에 대해서만 차선 인식 알고리즘을 적용하였다. 이 경우도 ROI는 고정된 영역으로 선택된다.

기존의 차선 인식 알고리즘의 문제점 중 하나는 차선 인식률에만 집중하여 인식 시간에는 큰 관심을 보이지 않았다는 것이다. 따라서 인식 시간에 대한 언급이 없거나 실시간성을 보장하지 못하는 경우가 대부분이다.

본 논문에서는 ROI 영역을 줄여 연산량을 줄이는 차선 인식 알고리즘을 제안한다. 제안한 알고리즘에서는 입력된 영상에 대해 선처리(preprocessing) 과정을 통해 영상을 압축하여 이미지 크기를 줄이고 차선 경계선 후보를 선택하여 경계선의 기울기에서 일정한 값만큼 크고 작은 기울기를 갖는 두 개의 직선을 이용하여 그 내부만을 적용적으로 V-ROI로 지정하는 방법을 통해 기존에 비해 연산량을 줄이면서 입력 영상의 해상도에 관계없이 거의 일정한 연산량을 유지하여 고해상도 영상에서도 성능이 저하되지 않는다. 제안한 알고리즘을 이용하면 하드웨어 가속기의 도움이 없이도 실시간 차선 인식이 가능하다. 제안한 알고리즘을 적용하고 OpenCV 라이브러리를 이용하여 윈도우즈 환경에서 차선 인식 소프트웨어를 구현하였으며 노트북에서 실시간 동작함을 확인하였다.

II. 직사각형 ROI 방식의 차선인식

제안하는 V-ROI 영역을 설정하기 위해서는 차선 경계선의 기울기를 알아야 하는데 첫 번째 프레임을 처리할 때는 이 기울기를 알 수 없으므로 직사각형(Rectangular) ROI 방식을 통해 기울기를 구하는 V-ROI 초기화 작업이 필요하다. II 절에서는 직사각형 ROI를 이용하여 차선 경계선을 인식하는 방법을 설명한다. ROI 영역을 제외하고는 V-ROI를 이용한 방식에서도 동일한 알고리즘이 이용된다. 직사각형 ROI 방식은 V-ROI 영역에서 차선이 사라져 초기화할 때도 필요하다.

차선 인식 알고리즘에는 다음과 같은 가정이 포함되어 있다.

- 1) 도로와 차선 텍스처는 일정하다 (consistent).
- 2) 도로와 차선의 폭은 국지적으로 일정하다.
- 3) 도로는 편평하거나 높낮이 차이가 있을 경우에는 정해진 모델을 따른다.

직사각형 ROI를 적용한 방식은 기존에 발표된 방식을 포함하였다. 즉, 카메라의 위치에 따라 전체 영상의 1/2에 해당하는 고정적인 부분을 ROI로 설정하고 선처리 과정을 거친 뒤, ROI를 좌우로 나누어 좌우의 차선 경계선에 대해 차선인식을 실행한다. 그림 1(a)에 나타난 영상에서는 앞차와 내차 사이의 도로상의 차선을 인식하기 위해 그림 1(b)와 같이 세로축에서 위에서 1/3 지점부터 아래 1/2의 영역을 ROI로 설정하였다.

ROI가 설정되면 해당 영역에 대해 그림 1(c)와 같이 무채색(grayscale)으로 변환한다. 이는 일반적으로 차선 경계선을 찾을 경우 배경보다 밝은 색을 인식하여 처리하므로 색채는 필요 없기 때문이다. 무채색 영상은 다시 이진화 영상으로 변환하는데 이 때 임계값에 따라 결과가 큰 차이를 보인다. 임계값은 날씨와 도로상태, 시간대 등에 따라 달라지는데 상수값을 쓰게 되면 좋은 결과를 얻기 어렵다. 본 알고리즘에서는 픽셀값의 평균과 분산을 이용하는 방식과 Otsu 임계값[10] 결정 방식을 개선한 방식을 상황에 따라 적용하도록 하였다. Otsu 임계값 결정 방식과 이를 개선한 방식이 평균과 분산을 이용하는 방식에 비해 상대적으로 연산량이 많기는 하지만 임계값 결정 단계가 다른 연산 단계에 비해 단순하므로 전체 연산 시간에는 큰 영향을 미치지 않는다. 제한된 방식을 이용하면 임계값이 영상에 따라 적절한 값으로 변하므로 영상의 상태에 따른 영향을 최소화할 수 있다. 또한 이 단계에서 그늘이나 가드레일 등에 의한 영향도 최소화할 수 있다.

그러나 영상의 상태와 환경에 따라 이러한 영향을 완벽히 제거하지 못하는 경우가 발생한다. 그림 1(d)에서는 점 잡음과 가드레일에 의한 대시선이 나타나는 것을 볼 수 있다. 양쪽으로 흰 차선 경계선이 있는 경우는 임계값 조절로 대시선 제거가 가능하지만 노란선이 있는 경우에는 무채색 변환 후 노란색의 밝기가 흰색보다 약하므로 이를 보존하기 위해서는 임계값을 낮추어야 하고 이에 따라 비슷한 밝기를 갖는 가드레일의 대시선이 사라지지 않는다. 이외에도 밝은 작은 물체들이 있을 수 있으므로 이를 제거하는 단계가 필요하다. 일반적으로 차선 경계선의 수직성분은 이러한 잡음보다 크므로 적절한 크기의 오픈링(opening) 처리를 통해 제거하였다. 그림 1(e)의 예에서는 2x5 크기의 직사각형 구성요소를 적용한 오픈링 처리를 통해 잡음을 제거하였다.

마지막으로 남은 차선 경계선에 대해 Canny 가장자리 탐지 방법을 이용하여 가장자리 선을 탐지한다. 그림 1(f)에 나타난 바와 같이 매우 우수한 가장자리 탐지 방법 중 하나로 알려진 Canny 가장자리 탐지방

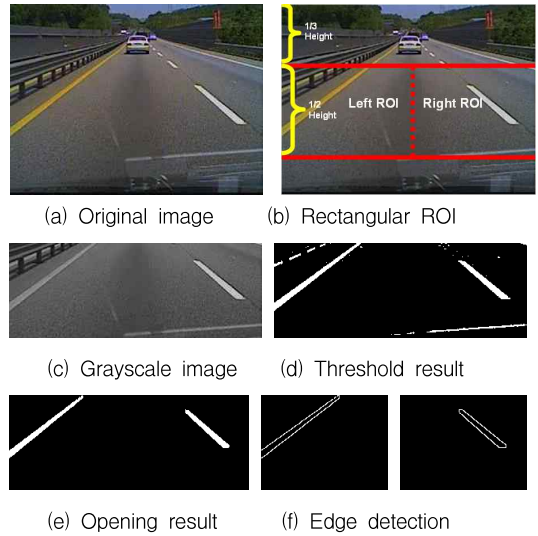


Fig. 1. Lane detection procedure in rectangular ROI
그림 1. 직사각형 ROI 방식의 차선인식 과정

법을 이용하면 한 픽셀 두께를 갖는 가장자리 선을 생성할 수 있다[7]. 최종 영상은 좌우 두 경계선에 대해 새로운 두 개의 ROI를 설정한다. 왼쪽의 ROI와 오른쪽의 ROI에 대해 적절한 각도를 갖는 선을 찾아 차선 경계선으로 설정한다. 그리고 그 외의 선들은 모두 제거한다.

차선의 위치는 Hough 변환을 이용하여 차선 경계선을 직선에 맞추어 결정한다[1-6]. 차선 경계선의 직선의 식은 다음과 같이 결정되고 ρ 는 직선 모델에 맞도록 결정된다[1].

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

본 알고리즘에서는 평균값을 구해 차선의 안쪽과 바깥쪽 가장자리에 있는 가장 강한 선을 선택하였다.

III. V-ROI 방식의 차선 인식

직사각형 ROI 방식을 이용하여 차선을 인식할 경우 잡음이나 잘못된 선들이 탐지되어 추가적인 처리 연산이 필요하고 차선을 잘못 인식하는 경우도 발생한다. 이러한 문제를 해결하기 위해 ROI 결정 방식을 개선한 V-ROI(V-shaped ROI)를 제안한다. 이를 위해 차선 경계선에 대한 방정식 (1)을 다음과 같이 수정한다.

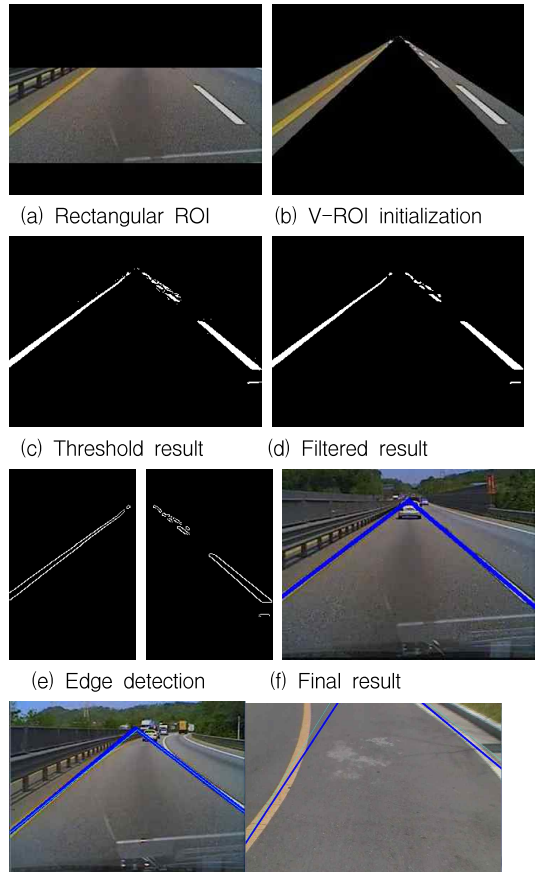
$$-y = k(x - x_0) \quad (2)$$

여기서 (x_0, y_0) 는 오른쪽과 왼쪽의 차선 경계선의 직선식이 만나는 점의 좌표이고 k 는 식 (1)의 θ 로부터 얻어지는 기울기이다. k 값을 ± 0.2 만큼 변화시켜 그림 2(b)와 같이 역방향 V자 모양의 ROI를 얻을 수 있다. ± 0.2 라는 값은 정상적인 주행 상태에서 차선이 움직이는 최대 범위를 경험적으로 얻은 값이다. 따라서 도로 상황에 따라서 변경하는 것도 가능하다. V-ROI 방식을 적용하기 위해서는 먼저 직사각형 ROI를 통해 차선 경계선을 탐지하고 그 기울기 k 를 구하는 초기화 과정이 필요하다. 그 이후에는 V-ROI 영역에서 k 값을 갱신한다. V-ROI는 기존의 방식에 비해 처리 영역을 크게 줄일 수 있어 연산 효율과 성능 향상에 효과가 좋다[1,3].

일반적으로 고속도로에서 정상 주행을 하는 경우 차량은 대체로 차선 가운데를 주행하므로 ROI의 변화가 크지 않다. 이러한 사실을 이용하여 이전 프레임의 차선 경계선 위치로부터 위에서 언급한 방법에 따라 V-ROI 영역을 구한다. V-ROI 내의 이미지는 직사각형 ROI 차선인식에서와 같은 방식으로 이진화하여 그림 2(c)와 같은 결과를 얻는다. V-ROI 영역은 차선 경계선 부분의 일부 영역만 포함되므로 잡음이나 가로 성분의 선, 또는 잘못된 선이 포함될 가능성이 낮다. 따라서 차선 경계선의 형태를 왜곡할 수 있는 오프닝 처리 대신에 메디안 필터를 이용하여 그림 2(d)와 같은 스무딩(smoothing) 결과를 얻는다. 이후에는 직사각형 ROI 차선인식과 마찬가지로 Canny 가장자리 탐지와 Hough 변환을 이용하여 그림 2(e)의 윤곽선을 구한다. 그림 2(f)에 최종 차선 인식 결과가 나타나 있다.

현재 구현된 소프트웨어에서는 인식된 차선 경계선에 직선을 그리는 방식이므로 곡선 차로의 경우에는 인식된 차선에 두 개의 직선을 그리면 서로 만나는 점이 그림 2(g)에 나타난 바와 같이 도로의 차선의 중앙에 위치하지 않는다. 그러나 실제 도로가 곡선이어도 완만하게 변하므로 차량 가까운 곳에서는 차선 경계선이 인식되고 계속 차량이 진행함에 따라 이를 따라감을 확인할 수 있었다. 따라서 곡선 도로에서도 차선 이탈 경고시스템은 작동한다.

V-ROI 방식에서는 이전 프레임에서 차선의 기울기에 대한 정보를 얻어오므로 차선 내에서 차량이 좌우로 조금 움직이는 경우에는 V-ROI 영역 내에 차선 경계선이 남아 있어 차선 인식에 영향을 미치지 않고 매 프레임마다 차선 경계선의 기울기가 갱신되어 적응적으로 V-ROI가 설정된다.



(g) Lane detection results in curved lane

Fig. 2. Lane detection procedure in V-ROI
그림 2. V-ROI 방식의 차선인식 과정

V-ROI 방식의 차선인식을 이용하여 차선이탈 탐지 및 경고 신호를 발생시키기 위해 제안한 알고리즘에서는 오류 프레임 카운터가 설정되어 다음 상황에서는 V-ROI 영역의 초기화가 이루어진다.

1. 주행속도에 따라 정해진 프레임 이상 V-ROI에서 하나의 직선이 발견되지 않는 경우
2. 탐지된 직선의 기울기가 허용된 범위를 벗어난 경우

1의 경우에는 차량이 급격히 움직여 차선 이탈이 예상되는 경우이며 2번의 경우는 서서히 차선을 이탈하고 있는 경우이다. 본 논문의 실험에서는 1의 경우 3 프레임을 기준으로 잡았고 이는 0.1초 동안 차선이 V-ROI 영역에 나타나지 않는 경우이다. 이 프레임 수는 차량의 이동 속도에 따라 적응적으로 설정한다.

둘 중 하나의 상황이 발생하면 V-ROI 영역의 초기화를 통해 차선 경계선을 재인식하고 차선 경계선의 기울기를 측정하여 차선이탈 여부를 판단하고 경고 신호를 발생시킨다. 그림 3(b)는 위에서 언급한 상황이 발생하여 직사각형 ROI 방식으로 전환한 뒤, 차선 경계선을 인식하고 경고 신호를 보내는 결과를 보여주고 있다.

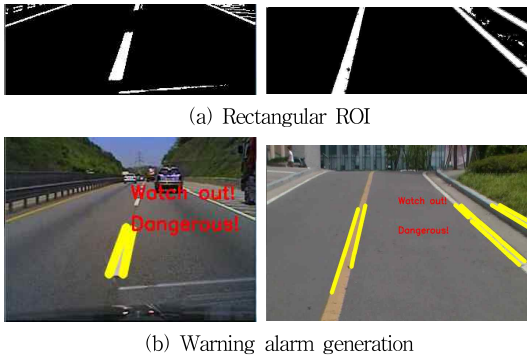


Fig. 3. ROI initialization and warning alarm generation when lane departure
그림 3. 차선이탈시 ROI의 초기화와 경고신호 발생

제안한 알고리즘을 정리한 순서도가 그림 4에 나타나 있다. 입력 영상이 들어오면 V-ROI를 적용할 수 있는 상황인지 아닌지를 판단하여 적용이 안 되는 상황이면 즉 V-ROI 초기화가 필요하면 직사각형 ROI를 먼저 설정한 후 차선을 인식하고 차선의 기울기를 구한 뒤 다음 영상에 대한 처리를 진행한다. V-ROI 처리가 가능한 상황에서는 V-ROI를 설정하고 선처리를 진행한 뒤 좌우 두 개의 분할 영상에 대해 각각 차선 경계선 인식 과정을 진행한다. 여기에서 차선 경계선을 탐지하면 기울기를 구하고 그렇지 않으면 이전 기울기 값을 이용하여 다음 프레임에 대한 인식 작업을 수행한다. 그 후 차선 이탈 여부를 확인하여 결과를 보내고 다음 영상에 대한 처리를 진행한다. 좌우 영상에 대한 차선 인식 과정은 가장 연산량이 많고 독립적으로 진행이 가능하므로 다중 프로세서가 있는 경우 병렬 처리를 통해 연산 시간을 줄일 수 있다. 따라서 듀얼 코어 이상의 스마트폰용 AP에서도 실시간 처리가 가능할 것으로 예상된다.

차선 인식 알고리즘은 일반적으로 전방 카메라로부터 입력된 영상에서 일정한 두께를 갖는 직선 또는 완만한 곡선을 인식하므로 그 대상의 형태가 복잡하지 않다. 따라서 고해상도의 영상을 필요로 하지 않는다. 이러한 사실을 이용하여 제안한 알고리즘에

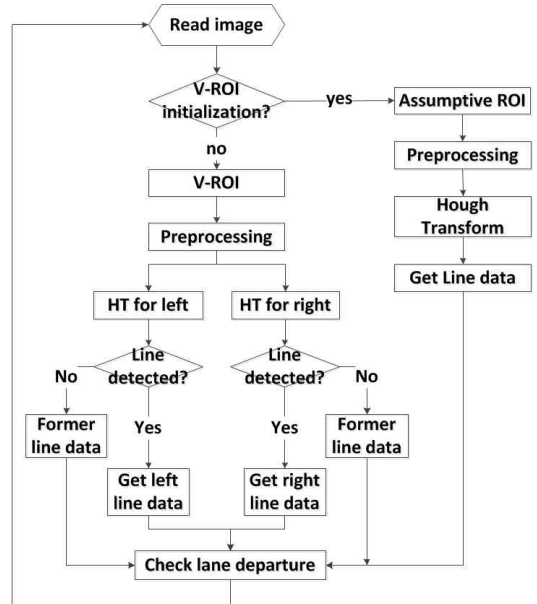


Fig. 4. Flow chart of the proposed algorithm
그림 4. 제안한 알고리즘의 순서도

서는 입력 영상의 해상도가 높을 경우에는 영상의 크기를 축소하여 차선을 인식하도록 하였다. 영상 크기 축소는 최종 영상의 크기가 QVGA(320x240)와 비슷해지는 범위에서 4개 또는 9개 픽셀의 평균값을 구하여 대체하였다. 따라서 영상 크기에 연산 시간이 거의 영향을 받지 않는다.

IV. 구현 결과

제안한 알고리즘에 따라 OpenCV 라이브러리를 이용하여 차선인식 및 차선이탈 탐지 기능을 구현하였다. 실행 환경은 2.83GHz의 CPU를 갖는 데스크탑 PC와 2.4GHz의 CPU를 갖는 노트북과 윈도우즈 7이고 단일 코어만을 이용하여 실행하였다. 실험에 사용한 영상은 QVGA와 SD(720x480) 해상도를 가지고 있다. PC와 노트북은 CPU 동작주파수는 큰 차이가 없으나 전체적인 성능은 메모리 등의 주변장치 성능 차이로 응용 소프트웨어를 실행시켜 보면 두 배 정도 차이가 있다. 표 1의 계산 시간 비교 결과를 보면 2.5 배 정도의 차이가 있음을 알 수 있다. 전체적인 계산 시간을 보면 Canny 가장자리 탐지와 Hough 변환이 대부분의 차지한다. Video2에서 괄호 안의 값은 영상 크기를 360x240으로 축소하기 전의 계산 시간이고 괄호 밖의 값은 축소 후의 계산이다. 입력 영상을 압축할 경우 영상 크기에 따른 계산 시간의 차이는 거의

없음을 볼 수 있다. 따라서 모두 프레임당 30ms 이내에서 차선 인식이 가능하므로 실시간 동작이 가능하다. 단일 코어만을 사용한 결과이므로 임베디드 시스템에서 듀얼 코어 이상을 사용하고 코드를 최적화할 경우 역시 실시간 동작이 가능할 것으로 예상된다. 또한 다양한 동영상을 입력받아 실행한 결과 모두 정상적으로 차선을 인식하였다.

Table 1. Calculation times according to the proposed algorithm

표 1. 제안한 알고리즘에 따른 계산 시간 [ms/frame]

Platform	Desktop		Notebook	
	video1*	video2**	video1*	video2**
① Pre-processing	1.1	1.3 (3.8)	2.7	2.8 (6.5)
② Canny edge	1.2	1.3 (3.6)	4.1	4.4 (14.1)
③ Hough Transform	1.3	1.7 (3.9)	5.2	4.9 (15.5)
④Calculation (①+②+③)	3.6	4.3 (11.3)	12	12.1 (36.1)
⑤Display	2.8	2.9 (6.2)	4.6	5.0 (6.7)
Total (④+⑤)	6.4	7.2 (17.5)	16.6	17.1 (42.8)

* QVGA ** SD

표 2는 ROI 방식에 따른 연산 시간을 비교한 결과이다. 표 1의 영상 1에 대해 데스크탑 컴퓨터에서의 연산 시간을 비교하였다. 첫 번째 열은 ROI를 사용하지 않은 경우이고 R-ROI는 전체 면적의 1/2을 ROI로 지정한 경우 세 번째 열은 V-ROI를 사용한 경우이다. ROI 영역 전체에 대해 연산을 수행하는 선처리 과정에서 예상대로 V-ROI가 가장 큰 효과를 보이고 있다. Canny 가장자리 탐지와 Hough 변환은 주변에

Table 2. Comparison of calculation times for various ROI methods [ms/frame]

표 2. ROI 방식에 따른 연산시간 비교 [ms/frame]

ROI	No ROI	R-ROI	V-ROI	V-ROI/R-ROI
① Pre-processing	2.0	1.5	1.1	0.73
② Canny edge	2.3	1.4	1.2	0.86
③ Hough Transform	2.9	1.9	1.3	0.68
④Calculation (①+②+③)	7.2	4.8	3.6	0.75

차선 경계선 이외의 선이 얼마나 탐지되었는지에 따라 결과가 달라지는데 임계값 결정과정에서 많은 불필요한 선들이 제거되므로 V-ROI의 사용으로 얻는 이득은 40% 정도이다.

표 3은 본 논문의 실험 결과와 기존 결과를 비교한 것이다. 비교 결과를 보면 플랫폼에 따른 성능 차이를 감안하여도 제안한 알고리즘에 따른 결과가 가장 우수함을 알 수 있다.

Table 3. Comparison results of calculation time with other works

표 3. 기존의 결과와의 계산 시간 비교

	Platform	Image size	Detection time [ms]
Qing Lin[1]	Intel Core2 1.86 GHZ	720x480	30 - 50
Gaikwad[6]	Intel i3 2.53 GHZ	N/A	54
Proposed	Intel Core2 2.83 GHZ	720x480	7 - 12

V. 결론

본 논문에서는 ROI 영역을 줄이고 입력 영상을 압축하여 연산량을 줄이는 차선 인식 알고리즘을 제안하였다. 제안한 알고리즘에서는 입력된 영상에 대해 선처리(preprocessing) 과정을 통해 영상을 축소하여 이미지 크기를 줄이고 차선 경계선 후보를 선택하여 그 주변만을 ROI로 지정하는 V-ROI 방법을 통해 기존에 비해 연산량을 줄이면서 입력 영상의 해상도에 관계없이 거의 일정한 연산량을 유지하여 고해상도 영상에서도 성능이 저하되지 않는다. 제안한 알고리즘을 적용하고 OpenCV 라이브러리를 이용하여 윈도 우즈 환경에서 차선 인식 소프트웨어를 구현하였으며 SD 해상도 영상에 대해 프레임당 PC에서 7ms, 노트북에서 17ms로 실시간 동작함을 확인하였다. 또한 추가적인 압축과 코드 최적화를 통해 계산 시간을 더욱 줄일 수 있어 임베디드 시스템에서도 실시간 동작이 가능할 것으로 예상된다.

References

[1] Qing Lin, Youngjoon Han and Hernsoo Hahn, "Real-time Lane Detection Based on Extended Edge-linking Algorithm", Second International Conference on Computer Research and Development, pp. 725 - 730, May 7-10, 2010, Kuala Lumpur, Malaysia
 [2] Joel C. McCall and Mohan M. Trivedi,

"Video-based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation", IEEE Transactions on Intelligent Transportation Systems, Vol. 7, pp. 20-37, 2006

[3] Xu Zhe, Li Zhifeng, "A robust lane detection method in the different scenarios", Proceedings of 2012 IEEE International Conference on Mechatronics and Automation, pp. 1358-1363, Aug. 5-8, 2012, Chengdu, China

[4] Shengyan Zhou, Yanhua Jiang, Junqiang Xi, "A Novel Lane Detection based on Geometrical Model and Gabor Filter", 2010 IEEE Intelligent Vehicles Symposium, pp. 59-64, June 21-24, 2010, San Diego, USA

[5] Claudio Rosito Jung, Christian Roberto Kelber, "Lane following and lane departure using a linear-parabolic model", Image and Vision Computing, Vol. 5, pp. 1192-1202, 2005

[6] Vijay Gaikwad, Shashikant Lokhande, "An improved lane departure method for Advanced Driver Assistance System", International Conference on Computing, Communication and Applications (ICCCA), pp. 1-5, Feb. 22-24, 2012, Tamilnadu, India

[7] P.M. Daigavan and P. R.Bajaj, "Road Lane Detection with Improved Canny Edges Using Ant Colony Optimization", Third International Conference on Emerging Trends in Engineering and Technology, pp. 76-80, Nov. 19-21, 2010, Goa, India

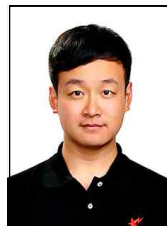
[8] Lee Kim Kuan, Ismail N.H, "Lane Guidance Warning System", International Conference on Computer and Communication Engineering (ICCC 2012), pp. 864 - 868, July 3-5, 2012, Kuala Lumpur, Malaysia

[9] Jung Gap Kuk, Jae Hyun An, "Fast lane detection & tracking based on Hough transform with reduced memory requirement", 13th International IEEE Annual Conference on Intelligent Transportation Systems, pp. 1344-1349, Sep. 19-22, 2010, Madeira Island, Portugal

[10] Nobuyuki Otsu, "A threshold selection method from gray-level histograms". IEEE Trans. Sys., Man., Cyber. Vol. 9(1), pp. 62-66. 1979

BIOGRAPHY

Ding Dajun (Student member)



2010 : BS degree in Electronic and Information Engineering, Shandong University of Science and Technology.

2011~present : Graduate student in MS degree in Dept. of Electronic Engineering,

Soongsil University

<Research Area> Image Pattern Recognition

Lee Chanho (Member)



1987 : BS degree in Electronic Engineering, Seoul National University.

1989 : MS degree in Electronic Engineering, Seoul National University.

1994 : Ph.D degree in Electrical Engineering, University of California Los Angeles

1994~1995 : Senior Research Engineer, Samsung Electronics.

1995~present : Professor, School of Electronic Engineering Soongsil University

<Research Area> SoC on-chip-network, Memory controller, Image Pattern recognition, SoC design methodology, Implementation of video codec