전방의 차량포착을 위한 연속영상의 대상영역을 제한한 효율적인 차선 검출

한 상 훈† · 조 형 제††

요 약

이 논문에서는 카메라로 연속으로 촬영한 일련의 그레이레벨 영상으로부터 전방의 차량을 포착하기 위한 빠른 차선검출방법을 제안한다. 개별 영상에서 가려지지 않는 제한된 영역을 대상으로 차선의 위치를 검출하고, 에지 영상을 이용하여 차선의 기울기를 구한다. 이를 근거로 차량이 존재할 가능성이 있는 관심영역을 구하고 그 영역 내에서 에지 성분을 이용하여 구조적 방법으로 전방 차량의 위치를 포착한다. 제안된 방식의 효과를 검증하기 위해 노트북 PC와 PC용 CCD 카메라로 도로에서의 영상을 촬영하고 차선검출알고리즘을 적용한 처리 시간, 정확도, 차량검지 등의 결과를 보인다.

Efficient Lane Detection for Preceding Vehicle Extraction by Limiting Search Area of Sequential Images

Sang-Hoon Han† · Hyung-Je Cho††

ABSTRACT

In this paper, we propose a rapid lane detection method to extract a preceding vehicle from sequential images captured by a single monocular CCD camera. We detect positions of lanes for an individual image within the limited area that would not be hidden and thereby compute the slopes of the detected lanes. Then we find a search area where vehicles would exist and extract the position of the preceding vehicle within the area with edge component by applying a structured method. To verify the effects of the proposed method, we capture the road images with a notebook PC and a CCD camera for PC and present the results such as processing time for lane detection, accuracy and vehicles detection against the images.

키워드 : 차선 검출(Lane Detection), 차량 검출(Vehicle Extraction), 장애물 검출(Obstacle Extraction), 연속영상(Sequence Image), 지역 밝기 차(Local Difference), 카메라(CCD Camera)

1. 서 론

현재 차량이 급격히 증가하고, 운송 수단의 기능만을 가지고 있던 차량이 이제는 생활과 밀접한 관계를 가지고 있다. 이에 차량과 도로 이용자들의 요구가 날로 증가하고 있고, 이 요구에 부응하는 정보를 제공하고자 하는 노력도 또한 증가하고 있다. 특히 주행중인 자동차의 전방에 존재하는 장애물 및 차량의 유무를 판단하고, 거리를 계산하는 것은 무인 자율 자동차 관련의 핵심 기술이다. 진행 방향에 있는 장애물이나 차량은 안전 운행에 직접적인 영향을 미치고, 이를 미리 찾아내어 운전자에게 경고를 하는 것은 안전 운전에 큰 도움이 될 것이다. 이러한 전방 차량에 대한 검출은 차량 자동화(vehicle automation) 연구 외에도 운전자가 안전운전을 하는데 전방의 장애물을 인식하여 경고를 해 주는 운전 도움(driving assistance)시스템의 목적으로도 연구들이 진행되고 있다[14].

차선을 인식하는 연구의 대부분이 차선 추출, 잡음의 제거와 곡선 도로의 추출에 관하여 연구되고 있다. 기존의 차선 인식 연구들은 영상의 모든 영역을 이용하여 차선을 인식하고 있으며, 기하학적인 변환과 모폴로지(morphology)를 이용하는 방법[7], 허프 변환(hough transform)을 이용하는 방법[20], 히스토그램(histogram)을 이용하는 방법[9], 스네이크(snake)를 이용하는 방법[15], 에지(edge) 연결정보를 이용한 방법[1]들이 있다. 이러한 방법들은 기상 변화나 빛의 명암, 도로 환경의 변화에 잘 적응되나 복잡하며, 처리 시간이 길어 실시간으로 구현하기에는 고가의 하드웨어들이 필요하다. 일부 관심 영역을 지정하여 관심 영역 내에서 차선을 검출하는 방법[2,

† 준 회 원 : 동국대학교 대학원 컴퓨터공학과
†† 정 회 원 : 동국대학교 영상정보통신대학원 멀티미디어학과 교수
논문접수 : 2001년 3월 15일, 심사완료 : 2001년 10월 17일

18]들이 있지만 이 경우에도 도로 상에 차량들이 있고, 차량에 의해서 차선이 가려지는 경우에 문제점들을 가지고 있다.

차량이나 장애물 인식에 관해서는 주행중 장애물 검출, 거리 및 속도, 상대 위치, 차량의 종류 인식에 관한 연구들이 진행되고 있다. 장애물을 검출하는 방법은 능동형 장치인 레이저(laser), 초음파(ultrasonic wave) 등을 이용하는 방법[4]들도 있고, 수동형 장치인 CCD 카메라 두 대를 이용하는 스테레오 비젼을 이용하는 방법, CCD 카메라와 다른 센서를 조합하여 이용하는 방법[4], 하나의 CCD 카메라를 이용하는 방법[14]들이 있다. 스테레오 비전을 이용하는 경우에는 대부분 disparity map을 구하였다.[3, 6, 10, 11] 그 외 하나의 카메라를 이용한 경우에는 구석점(corner point)과 광류(optical flow)를 이용하는 방법[17], 차량과 비 차량을 인식하는 방법[5], 엔트로피(entropy)를 구하여 salient map을 구하여 차량을 포착하는 방법[14]들이 있다. 이 방법들은 고속의 처리 시간이 요구되며, 차량의 그림자에 의한 문제들이 발생한다. 그리고 일정 영역의 부 윈도우(sub-window)를 이용하여 노면과의 밝기 차이를 이용하는 통계적 방법[18], 에지를 이용하여 차량의 모델[8]을 이용하는 방법 등이 있는데 이런 방법들은 노면의 상태나 주위 배경에서 발생하는 그림자에 의한 영향을 많이 받는다. 대부분의 연구들이 모두 고가의 영상처리용 프로세서(digital signal processor)를 이용하고, 차량에 장착하기 위해서는 고가의 비용을 지불해야 한다.

본 연구에서는 차량의 중앙에 설치된 하나의 CCD 카메라를 이용하여 전방의 차량을 포착하기 위해 빠른 시간 내에 차선을 검출하는 방법을 소개한다. PC 카메라와 노트북을 이용하여 영상을 얻기 때문에 모든 처리를 CPU가 담당해야 하고, 낮은 해상도와 별도의 영상 처리용 프로세서를 사용하지 않기 때문에 빠른 처리 시간이 요구된다.

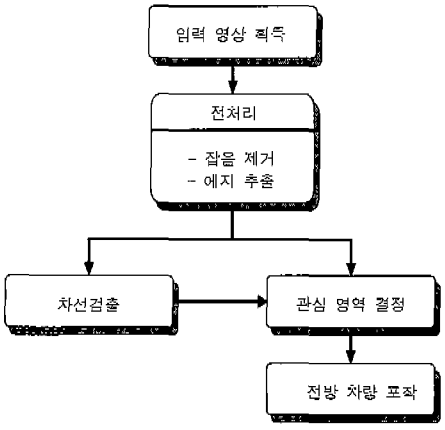
차선 검출은 일반적으로 전역정보를 이용하여 차선에 대한 기울기, 곡선 정보 등을 찾는데, 본 연구에서는 전역 정보를 이용하지 않고 다른 차량에 의해 방해받지 않는 제한된 영역의 정보만을 이용하여 빠른 시간 내에 차선을 검출한다. 차선의 검출은 좌, 우측 차선만을 검출대상으로 한다. 차량 포착은 차선 검출 단계에서 검출된 좌, 우측 차선 안에 있는 영역만을 대상으로 차량의 수평, 수직 에지 성분을 이용하여 포착한다.

본 논문의 구성은 2장에서 전체 시스템의 구성과 전처리 부분을 소개하고, 3장에서는 차선 검출 방법, 4장에서는 전방 차량을 포착하는 방법을 소개한다. 5장에서는 연구 결과를 제시하고, 6장에서 결론을 맺는다.

2. 시스템 구성 및 전처리

2.1 시스템 구성

전방의 차량을 포착하기 위한 차선 검출 시스템의 전체 구성도는 (그림 1)과 같이 5단계로 나뉘어 진다. 입력 영상 획득 단계에서는 도로의 영상을 그레이레벨로 입력을 받는다. 두 번째는 전처리 단계로써 잡음 제거와 에지 정보를 구한다. 잡음 제거 방법으로는 CCD 카메라로부터 얻어진 영상에서 잡음 제거와 차량 그림자에 의한 에러를 최소화하기 위해 지역적 밝기 차 방법을 이용한다. 그리고, 에지 정보는 Robinson 연산자를 이용하여 구하고, 차선의 기울기와 전방 차량을 포착하기 위한 정보로 활용된다. 차선을 찾는 단계에서는 전체 영상을 이용하지 않고, 제한된 영역의 정보만을 이용하여 빠른 시간 내에 차선을 검출한다. 차선을 찾는 알고리즘은 차량의 전방에 해당하는 제한된 영역에서 3개의 부 영역으로 나누어 각 영역에서 차선의 특징에 해당하는 지점이 서로 일치하는가를 보고, 일치하는 영역에서 차선의 기울기를 계산한다. 앞에서 찾은 차선 정보와 에지 정보를 이용하여 관심 영역을 결정하고, 관심 영역 내에서 차량의 위치를 찾는다. 전방 차량 포착 단계에서는 차선 안에 있는 영역만을 이용하여 전방의 차량을 포착한다.



(그림 1) 전체 구성도

2.2 전처리(Preprocessing)

달리는 차량의 전면에 부착된 카메라를 이용하여 얻은 영상에서 잡음을 제거한다. 잡음제거 방법으로는 지역 밝기 차 방법(local difference)을 이용한다. 영상의 특징을 보존하고, 그림자 및 조명에 의한 특이점을 제거하는데 적절한 방법이다. 지역 차 밝기 방법은 현재 화소에서 n×n 윈도우를 설정하여 윈도우의 평균값을 구하고, 현재 화소의 값에서 빼는 방법이다.

$$M = \frac{\sum_{i}^{n^2} F(i)}{n^2} \tag{1}$$

$$F'(i) = (F(i) - M) + bias \tag{2}$$

여기서 F(i)는 i번째 이웃 화소의 값을 나타내고, M은 영역 평균을 의미한다. n^2은 윈도우의 크기로써 여기서는 3

×3 윈도우를 이용한다. bias는 작은 값들로 이루어지고, 음수의 부분도 있기 때문에 더해주는 값이다. 여기서는 128을 사용했다.

에지 추출 알고리즘은 대표적인 방법으로 Sobel 연산자나 Canny 연산자 등이 있는데 여기서는 Robinson 연산자를 이용한다. Robinson 연산자는 Sobel 연산자와 비교하여 결과에 큰 차이가 없고, 정수 연산만을 이용하고, 각 에지 화소에 대한 대표 방향 값을 구할 수 있기 때문에 에지 추출방법으로 Robinson 연산자를 이용하였다.

Robinson 연산자는 (그림 2)와 같이 4개의 방향 마스크를 이용한다. 기본 방향을 0, 45, 90, 135도로 정한 4개 마스크를 이용하여 기울기의 크기가 가장 큰 값을 대표 방향으로 하고, 그 중 문턱치 이상인 점을 에지 화소로 결정한다. M1이 기울기가 0도 인 수평 에지 성분에 해당하고, M2는 45도, M3는 90도 그리고 M4는 135도를 나타낸다.

1	2	1
0	0	0
-1	-2	-1

M1

2	1	0
1	0	-1
0	-1	-2

M2

1	0	-1
2	0	-2
1	0	-1

M3

0	-1	-2
1	0	-1
2	1	0

M4

(그림 2) Robinson 연산자 4방향 마스크

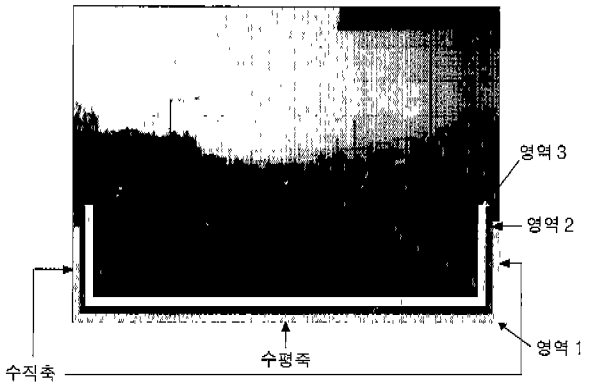
3. 차선 검출

본 알고리즘은 차선을 검출하는 데 전역 정보를 이용하는 것이 아니라 다른 차량에 의해서 방해받지 않는 제한된 영역만을 이용한다. 차선은 기본적으로 도로보다 밝은 색으로 칠해져 있으며, 점선이나 직선으로 이루어져 있다. 대부분의 차선 검출 알고리즘은 전방의 차량이 많은 경우에 차선을 효과적으로 추출하기가 어렵다.

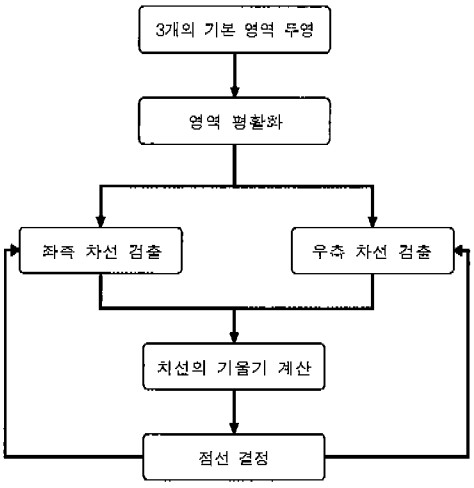
차선 검출을 위해 영역을 제한하는 방법으로 폭과 높이가 20×50인 크기를 갖는 부 윈도우를 좌, 우로 각각 4개씩 설정하여 이전 프레임에서 검출된 차선 방정식을 이용하여 차선을 찾는 방법이 있다[18]. 이 방법은 설정된 영역에서 차선이 차량에 의해 가려진 경우에 에러가 발생할 가능성이 높다.

먼저 입력 영상에서 차선을 추출하기 위해 다른 차량에 의해서 차선을 추출하는 데 방해받지 않을 뿐만 아니라 차선이 가장 뚜렷하게 나타나는 부분을 설정한다. 그리고 이 영역은 차량이 달리고 있는 상황에서는 가려지지 않는 부분이다. 이 제한적인 영역은 (그림 3)과 같이 영상의 하단에 위치하며 이 영역을 3개의 기본 영역으로 구분한다. 3개의 기본 영역으로 구분하는 이유는 차량이나 차선이 아닌 부분을 결정하기 위함이다. 차선이 보이는 부분은 차선이 수평축에만 존재하지 않고 양 옆 부분인 수직축에도 나타나기 때문에 수평축과 수직축을 이용한다. 입력 영상의 크기가 320×240의 크기이기 때문에 수평축으로 320에 수직축으로 좌, 우 80을 사용하여 이 기본 영역을 펼쳐 놓으면 하나의 영역이 5×480에 해당하는 크기가 된다.

이렇게 구해진 기본 3개의 기본 영역을 이용하여 차선을 검출하는데 차선을 검출하는 흐름도는 (그림 4)와 같다. 먼저 3개의 기본 영역에 대해서 투영(projection)을 통하여 화소의 밝기 값의 합을 구하고 영역 평활화 과정을 거친다. 영역 평활화 과정을 거치고 나면 3개의 기본 영역에 대해서 1차원 정보로 변환되는데 기본 영역의 값에서 미분을 통하여 차선의 후보 위치를 구하고 3개의 기본 영역에서 차선의 특징에 해당하는 지점이 일치하는 곳을 찾아 왼쪽과 오른쪽 차선으로 간주한다. 그리고 차선이 점선인 경우에는 검출 영역에 차선이 들어오지 않기 때문에 차선을 구할 수가 없기 때문에 이 경우에는 추정된 차선의 정보를 이용한다.



(그림 3) 차선 검출 영역



(그림 4) 차선 검출 흐름도

3.1 기본 영역 투영

세 개의 영역에 대해서 x, y 축에 대한 명암의 밝기 합을 구한다. 수평축에 해당하는 영역은 x축에 대해서 명암의 밝기 합을 구하고, 수직축에 해당하는 부분은 y축에 대해서 명암의 밝기 합을 구한다.

$$P_k(i)=\sum_{j=0}^{4} f(x_{c+j}, y_c) : \text{왼쪽 수직축}$$

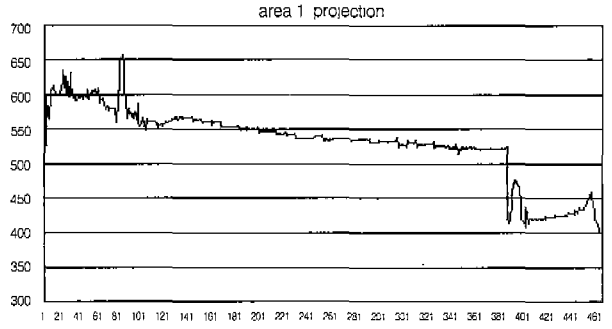
$$P_k(i)=\sum_{j=0}^{4} f(x_c, y_{c+j}) : \text{수평축}$$

$$P_k(i)=\sum_{j=0}^{4} f(x_{c+j}, y_c) : \text{오른쪽 수직축} \quad (3)$$

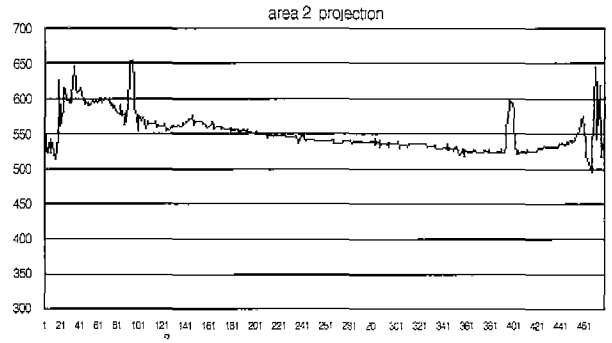
(x_c, y_c) : 기준점

$$\text{i의크기} : \begin{cases} \text{왼쪽수직축} : & 80+(k-1)*5 \\ \text{수 평 축} : & \text{width}-2(k-1)*5 \\ \text{오른쪽수직축} : & 80+(k-1)*5 \end{cases}$$

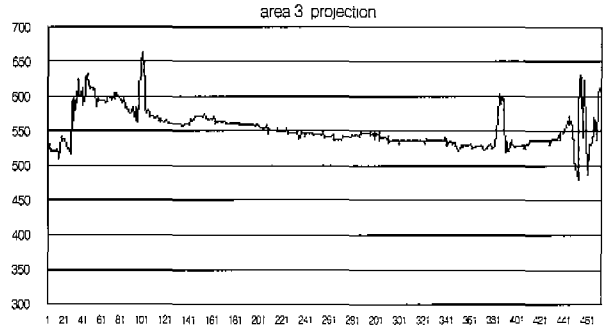
$P_k(i)$는 각 영역에 대한 밝기 합이다. $P_k(i)$에서 i는 0~479까지이며, 투영의 방향은 왼쪽과 오른쪽의 수직축에서는 x축 방향이고, 수평축에서는 y축 방향이다. $P_i(i)$에서 i의 범위는 0에서 79는 왼쪽 영역으로 y값은 고정하고, x축으로 투영하고, 80에서 399는 x축을 고정하고, y축으로 투영한다. k는 3개의 영역번호를 나타낸다. 영역 2에서는 수평축의 길이가 줄어들고, 수직축의 길이는 늘어난다. $P_k(i)$값을 그래프로 그려보면 (그림 5)와 같이 나타난다. (a)는 영역 1이고, (b)는 영역 2, (c)는 영역 3을 나타낸다. x축은 영역의 위치에 해당하고, y축은 $P_k(i)$를 나타낸다.



(a) 영역 1



(b) 영역 2



(c) 영역 3

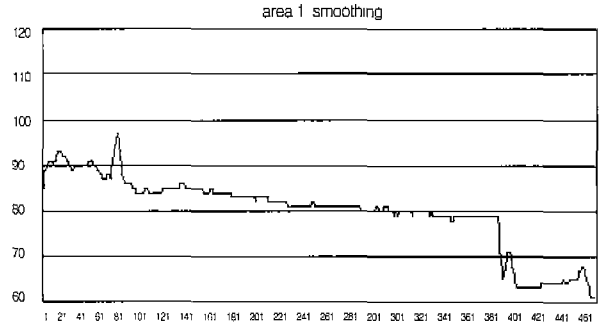
(그림 5) 3개 기본 영역의 투영 결과

3.2 영역 평활화

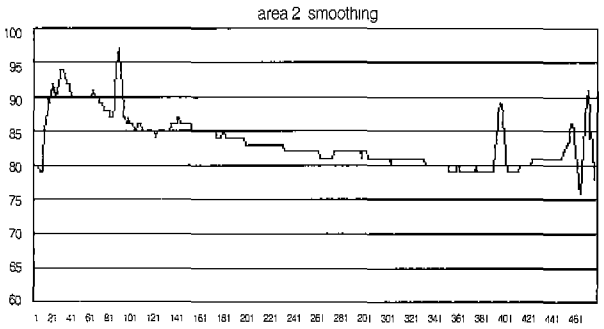
각 영역에서 명암 값의 합을 구한 결과를 보면 작은 지역적 최고점들이 있기 때문에 평균에 의한 평활화 과정을 거친다. (그림 6)에서 각 영역의 평활화 결과를 보인다.

$$P'(i)=\frac{\sum_{j=-n/2}^{n/2} P(i+j)}{n} \quad (4)$$

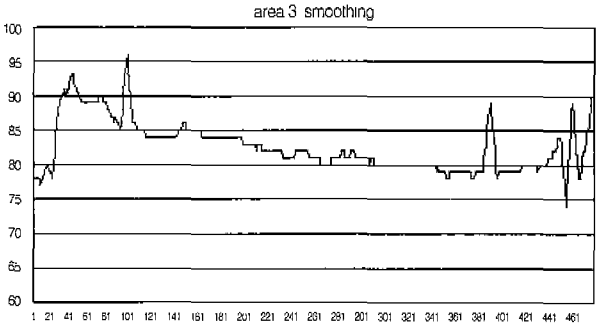
이 때 n의 값은 5를 이용했다.



(a) 영역 1



(b) 영역 2



(c) 영역 3

(그림 6) 3개 기본 영역의 평활화 결과

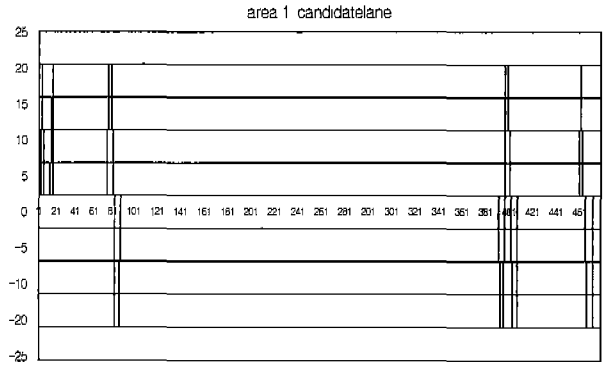
3.3 차선 결정

차선의 후보 영역은 1차 미분을 통하여 기울기를 구하고, 기울기를 이용하여 차선의 위치를 구한다. 기울기가 연속으로 (+)인 위치와 기울기가 연속으로 (-)인 위치를 찾아 차선의 후보로 지정하고, 기울기가 (+)방향을 시작 위치로 기울기가 (-)방향을 끝 위치로 결정한다. 이렇게 찾은 차선의 후보 영역 중에서 세 영역간 위치 정보의 차이가 비슷한 경우가 차선에 해당한다. 본 연구에서는 좌, 우측 차선만을 검출하기 때문에 영상의 중심을 기준으로 좌, 우측으로 처음 발견되는 차선의 후보가 실제 차선에 해당한다. 식 (5)에서 C(i)는 i번째 차선의 중심점을 나타내고, 각 영역간의 위치 정보가 유사한 점을 찾는다.

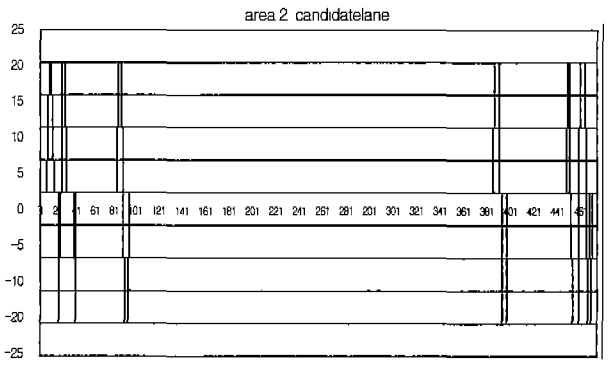
$C(i) = |\, i\text{번째 차선의 시작 위치} - i\text{번째 차선의 끝 위치} \,|$
$C_1(i)$: 1번 영역의 차선후보

$$\begin{aligned} &\text{if}(|C_1(i)-C_2(i)|-|C_2(i)-C_3(i)|)<Th \\ &\text{then} \quad \text{Lane} \\ &\text{else} \quad \text{Not Lane} \end{aligned} \tag{5}$$

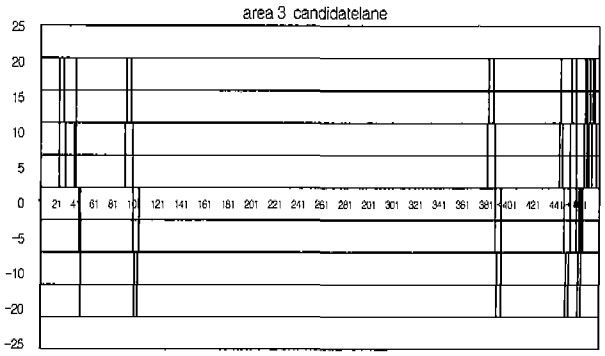
이렇게 구해진 위치를 다시 실제 영상의 위치로 변환한다. 영상 내에서 제한적인 지역적 영역 정보만을 이용하더라도 차선을 충분히 찾을 수 있다. (그림 7)은 차선의 후보 영역에 대한 위치들을 보여준다. (그림 7)에서 위치 61부분에 보면 세 개의 영역에서 모두 같은 결과를 보이고 있다.



(a) 영역 1



(b) 영역 2



(c) 영역 3

(그림 7) 3개 기본 영역에서의 차선 후보 위치

3.4 차선의 기울기 결정

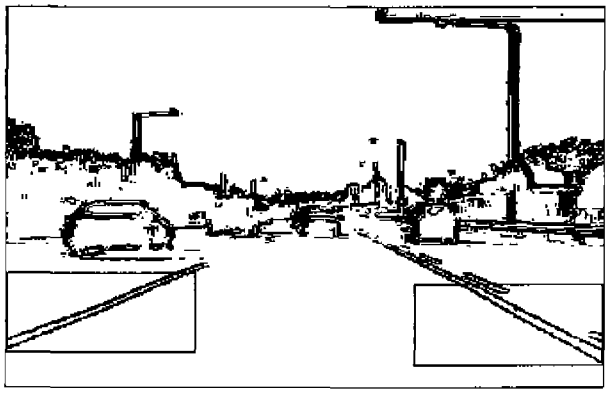
차선의 기울기는 에지(edge) 영상을 이용하여 구한다. 앞에서 구해진 차선의 위치를 기준으로 윈도우를 설정하여 기울기를 구한다. 좌측 차선인 경우에는 기준점에서 오른쪽 방향으로 윈도우를 지정하고, 우측 차선인 경우에는 기준점에서 왼쪽 방향으로 윈도우를 지정한다. 이것은 차선이 최대한 보이는 곳으로 윈도우를 설정하기 위함이다. 윈도우의 크기가 작으면 기울기의 오차가 크고, 처리시간이 빠르며, 윈도우의 크기가 크면 오차는 줄어들지만 처리 시간이 길어진다. 여기서는 윈도우의 크기를 100×50으로 이용한다. (그림 8)은 차선에서 윈도우를 지정한 결과이고, (그림 9)는 차선의 기울기를 구하는 방법을 도식화하였다.

$$\overline{PQ_i} = \sum_{(x,y)\ \text{on}\ PQ_i} (e(x,y)) \tag{6}$$

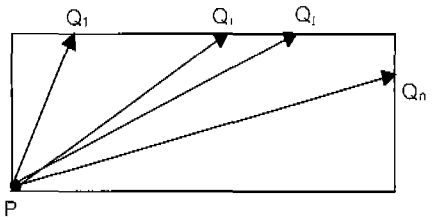
$$e(x,y) = \begin{cases} 1, & \text{if edge pixel} \\ 0, & \text{else} \end{cases}$$

$$\text{기울기} = \max(\overline{PQ_i}) \tag{7}$$

식 (7)에서 PQ_i가 최대가 되는 것이 차선의 기울기가 된다. (그림 9)에서 이때 Q_i는 P점의 반대 방향으로 에지 화소가 있는 점을 기준으로 하였으며, 에지 화소가 없는 경우에는 상단의 전 범위를 지정한다.



(그림 8) 기울기 추정을 위한 윈도우



(그림 9) 기울기 측정 방법

이 방법에서 차선이 완만한 곡선인 경우에는 거의 직선에 가깝기 때문에, 큰 문제없이 차선의 정보를 유지할 수 있다. 그리고 양옆의 차선의 기울기를 구하여 두 점이 만나는 소실점(vanish point)을 구하면 카메라 정면에 보이는 차량을 검출하는데 효과적이다.

3.5 점선의 결정과 추적

차선에는 점선과 실선이 있다. 도로공사의 시설물 관리기준에 보면 고속도로인 경우에 점선의 차선은 도색된 길이가 8m이며, 12m간격으로 그려진다[19]. 영상내의 지역 정보만을 이용하기 때문에 차선을 구하는 과정에서 차선이 점선인 부분은 설정된 영역에서 보이지 않을 수 있다. 차선으로 결정되고, 차선의 기울기가 구해지면 이 차선에 대한 정보를 이용하여 차선을 추적한다.

$$S'(X) = \alpha * C(X) + (1-\alpha) * S(X) \quad (8)$$

일단 차선이라고 판정되면 차선이 같은 영역에서 연속으로 검출되는 경우에 실제 차선으로 지정하고, 추적을 시작한다. 같은 차선임을 판별하는 방법은 차선의 시작점의 거리가 일정거리 내에 있으면 같은 차선으로 간주한다. 이전 프레임에서 검출된 차선의 위치와 현재 프레임에서 검출된 차선의 거리를 이용하여 같은 차선임을 판단하고, 차선의 정보를 보정한다. S'(X)는 추정된 차선의 다음 정보이며, S(X)는 추정된 차선의 현재 정보이다. 그리고 C(X)는 현재 구해진 차선의 정보이다. 여기서 α 값은 0.1을 이용하였다.

차선이 점선임을 판단하는 근거는 차선을 검출하는 과정에서 차선이 나타나는 프레임 수와 나타나지 않는 프레임 수를 비교하여 이 주기가 반복적이면 점선으로 결정하고, 차선이 나타나지 않는 경우에도 추정된 값을 이용하여 차선의 정보를 유지한다. 한쪽 차선이 점선이고, 다른 한쪽 차선이 실선인 경우에, 점선인 차선이 영역 내에서 보이지 않을 때는 실선인 차선의 기울기와 시작점의 정보를 이용하여 점선 차선의 정보를 보정한다.

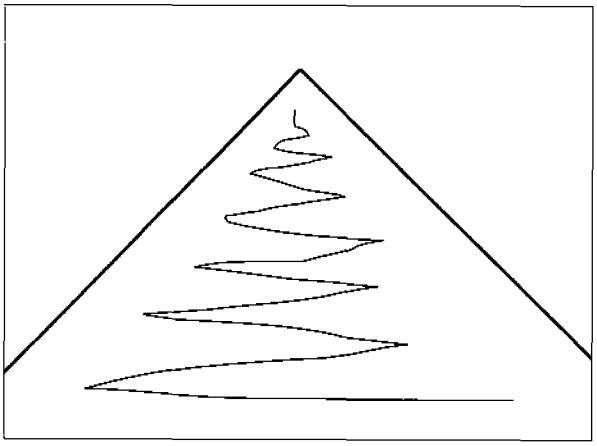
4. 전방의 차량 포착

도로상의 차량은 이미 검출된 차선 안에 있는 전방의 차량을 찾는 것으로 차량의 수직, 수평 에지 성분을 이용한다. 한 차선 내에서 차량, 그림자와 도로의 상태에 따라서 에지 성분들이 구해진다. 하지만 그림자와 도로의 상태에 의해서 발생하는 에지 성분들은 지역 밝기 차 필터를 적용하여 대부분 제거가 가능하나 밝기 차이가 많이 나는 경우는 제거하기가 어렵다. 그리고 차량에 대해서는 특히 수직과 수평 에지가 두드러지게 나타나기 때문에 전방의 차량을 검출하기 위해서는 수직과 수평 에지 성분을 이용한다. 차량을 검출하기 위해서 먼저 수직 에지 성분을 이용하여 관심 영역을 구하고, 그 영역 내에서 차량의 위치를 찾는다.

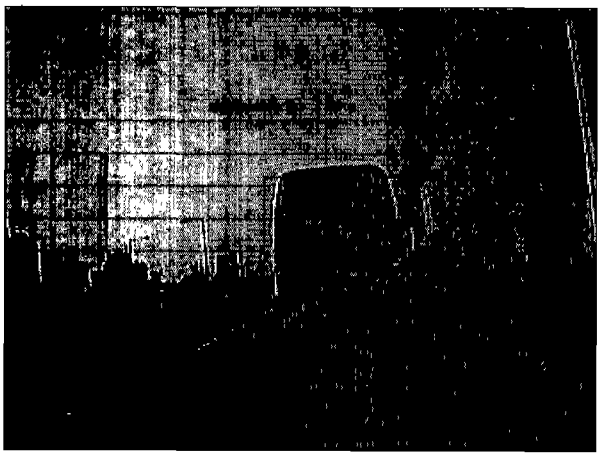
4.1 관심 영역 지정

관심 영역은 앞에서 구한 차선 정보를 이용하여 영상의 밑에서부터 수직 에지 성분을 찾는다. 이때 에지 정보를 찾는 과정에서 수평 성분은 차선이나 외부의 그림자 및 차량의 그림자에 의해서 발생하기 쉬운 성분이기 때문에 수직 성분만를 이용하여 관심 영역을 찾는다. (그림 12)는 Robinson 연산자를 이용하여 수직 에지 만을 찾은 결과이다.

(그림 10)과 같이 차선 내에서 지그재그(zigzag)로 수직 성분을 찾는다. 이렇게 찾은 영역이 (그림 11)과 같다. 차량

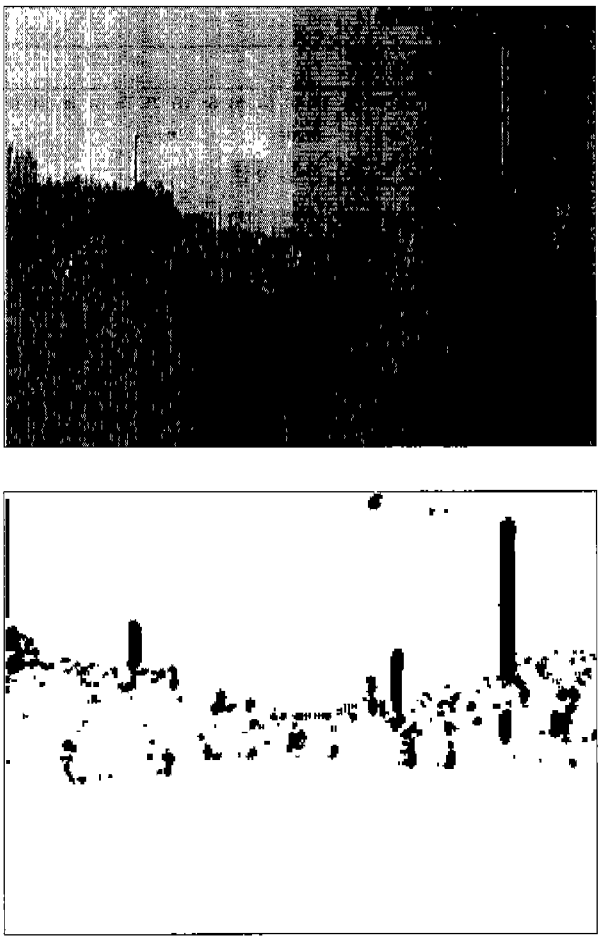


(그림 10) 관심영역결정



(그림 11) 관심영역추출

의 넓이와 높이에 따라서 관심 영역의 크기가 중요한데 여기서는 관심영역의 시작점과 소실점까지의 거리의 2배가되는 점을 이용하여 관심 영역의 높이를 구한다. 앞에서 구한 에지 중에서 수직방향에 해당하는 에지를 이용하면 쉽게 관심 영역을 구할 수 있다.

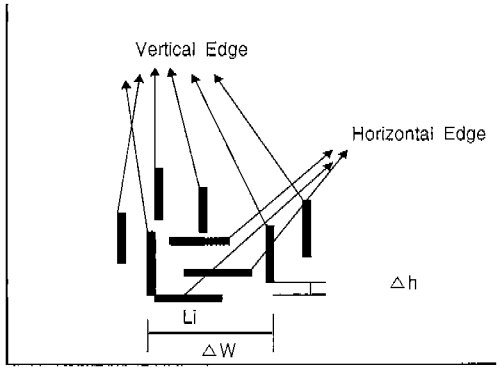


(그림 12) 수직 에지 추출 결과

4.2 차량 포착

차량 포착 방법은 에지 추출단계에서 구해진 에지 성분들을 이용하여 구조적 방법을 적용한다. 수평 성분과 수직 성분을 이용하여 구조적 방법을 통해 차량의 후미부분을 찾는 것으로 차량의 거리를 가늠할 수 있는 차량 후미의 시작 위치를 찾는다.

앞에서 구해진 관심 영역에는 차량의 정보뿐만 아니라 주위의 배경에 해당하는 부분도 포함되기 때문에 차선 정보를 이용하게 된다. 관심 영역 내에서의 에지는 (그림 13)과 같이 여러 개의 수직, 수평 성분들로 존재한다. 이때 수직, 수평 에지 성분을 8방향 연결도를 적용한 blob coloring을 통해 소 영역으로 분할하고, 차선 안에 존재하는 소 영역들만 남긴다. 남은 소 영역들 중에서 수직 에지 성분 사이의 넓이정보(△W), 수직 에지 성분 사이의 높이 정보(△h), 수평 에지 성분의 길이(L_i)를 이용하여 차량의 후미 위치를 구한다.



(그림 13) 에지들의 관계

차량 포착 방법을 단계별로 살펴보면 다음과 같다.

단계 1. 수직 에지 성분의 쌍과 수평 에지 성분의 집합을 구한다.
수직 성분 V = V_1, V_2, ….
수직 성분의 쌍 VP = (VP_12, VP_13, … VP_23, …, VP_n)
수평 성분 H = H_1, H_2

단계 2. 수평 에지 성분이 차선 안에 존재하는지 검사하여 차선밖에 있는 것은 제거

단계 3. 수직 에지 성분의 쌍이 모두 차선 안에 있는지 검사하여 차선밖에 있는 것은 제거.

단계 4. 수직 에지 성분의 쌍의 넓이가 차선의 넓이보다는 작고, 차선의 넓이의 1/3보다는 큰지 검사하여 범위를 벗어나면 제거.

단계 5. 수평 에지 성분을 포함하고, 수직 에지 성분의 쌍의 높이가 △H보다 작은 수직 에지 성분의 쌍을 구한다.

단계 6. 단계 1~5까지 통과한 나머지 수직, 수평 에지를 이용하여 차량의 위치를 포착한다. 여기서 에지 성분이 하나도 없는 경우에는 차량이 없는 것으로 간주하고, VP와 H의 후보가 여러 개인 경우에는 다음 단계로 간다.

단계 7. 단계 6에서 구해진 수직 에지 성분 쌍과 수평 에지 성분을 이용하여 수평 성분을 이용하여 차량 후미 부분의 시작(Y) 위치를 찾고, 수직 성분을 이용하여 차량의 폭을 지정한다. 후보가 여러 개인 경우에는 시작(Y) 위치가 최대인 후보를 차량의 후미 시작 위치로 결정한다.

단계 8. 이때 위에서 구해진 수평 에지 성분의 위치를 이용하여 판별하지 못하면 관심 영역의 시작 위치로 정한다.

5. 실험 결과

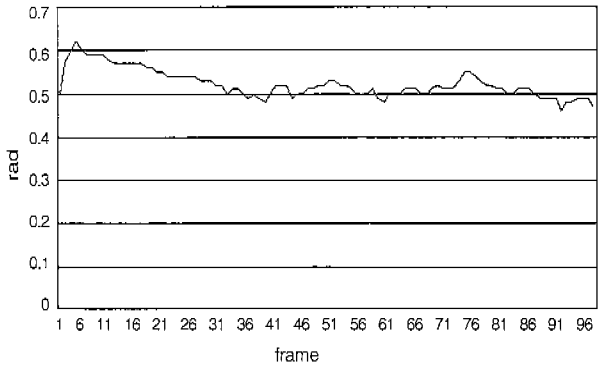
5.1 실험 환경

차량의 중앙에 USB 방식의 PC용 CCD 카메라를 설치하고, 올림픽도로 상에서 실험 영상을 얻었으며, USB용 PC 카메라를 이용하기 때문에 원거리 영상인 경우에는 영상이 일반 비디오 카메라나 CCD 카메라를 이용하는 것보다 화질이 선명하지 않은 점이 있다. 실험 영상은 15fps의 속도로 촬영하였으며, 영상을 AVI 형태의 파일로 저장한 뒤에 Adobe사의 프리미어 5.5를 이용하여 2000개의 연속적인 정지 영상을 만들어 실험하였다.

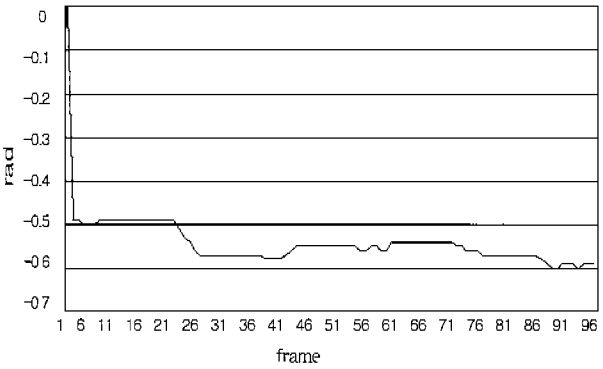
5.2 차선 검출 결과

차선을 추출한 결과로 먼저 차선의 기울기 변화를 보았다. (그림 14)에서는 좌측차선과 우측 차선에 대한 기울기로 라디안값을 표시하였다. (그림 14)의 (a)는 좌측 차선으로 차선의 기울기가 양의 방향이고, (b)는 우측 차선으로 기울기가 좌측 차선과 반대인 음의 방향으로 나타난다. 차선의 기울기는 도로의 기울어짐과 차량의 위치에 따라서 조금씩 변하고 있다. (그림 15)에서 나타난 바와 같이 소실점은 거의 변하지 않고 있다. 이것은 하나의 소실점에 거의 일치하고 있으며, 영상에서 차선의 위치가 조금씩 변해감에 따라서 기울기가 변하여 일정한 소실점를 구하고 있다. 처음에는 차선 정보를 구할 수 없기 때문에 소실점을 구할 수 없고, 두 개의 차선을 검출한 뒤에 소실점을 구하고 있다. x, y축에 대한 변화를 별도로 나타내었다. (그림 16)은 검출된 차선의 위치를 보여주고 있으며, 차선의 위치를 찾은 영역에 대한 결과를 작은 사각형으로 표시하였고, 기울기를 직선으로 표시하였다. (a)와 (b)에서는 양쪽 차선을 정상적으로 찾은 결과를 보여주고 있으며, (c)에서 오른쪽 차선은 추정된 차선을 표시하였다. 검출된 차선의 검출결과는 98%의 정확도를 보였으며, 2%의 오류는 차선이 너무 희미하게 나타나서 차선의 위치는 찾았으나 기울기를 구하는 과정에서 오차가 발생하는 경우와 차선을 추적하는 과정에서 차선의 범위를 벗어나는 경우가 있었다. 차선 내에 있는 글자가 있는 경우에는 차선의 추적과정에서 좌, 우측 차선의 정보와 다르기 때문에 차선의 후보에서 제거하기 때문에 가능하다. 하지만 차선의 위치와 유사한 글씨 부분에서는 차선으로 잘못 인식되는 경우가 있었다.

(그림 17)의 (a)는 차선의 끝 부분에서의 기울기를 구하는 과정에서 에러가 발생한 것이다. 기울기의 각도가 차이가 얼마 나지 않지만 소실점 부근에서는 많은 차이를 보인다. (b)는 오른쪽 차선에서 차선의 위치는 제대로 찾았으나 기울기를 구하는 과정에서 차선의 정보를 잘못 구한 경우이다. 이것은 시작점에서 에지 정보를 이용하여 차선의 기울기를 구하는 과정에서 에지 정보가 소실되어 나타나는 결과이다.

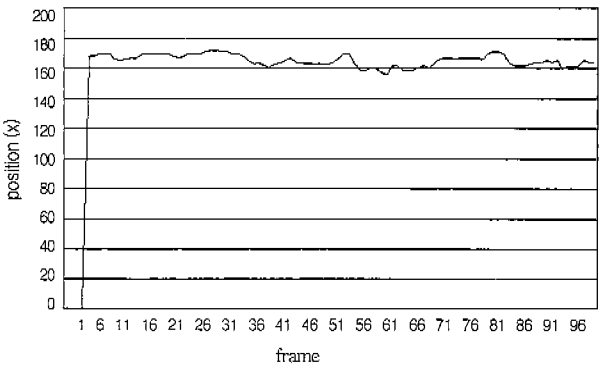


(a) 좌측 차선

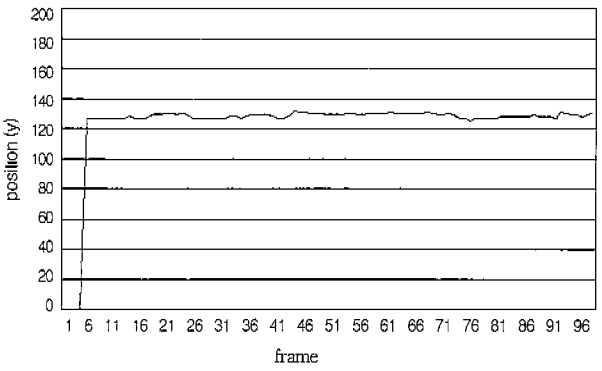


(b) 우측 차선

(그림 14) 차선별 기울기 변화



(a) x값 변화

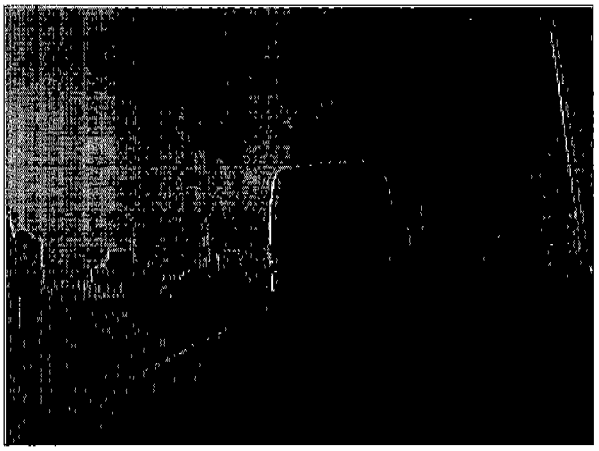


(b) y값 변화

(그림 15) 차선의 소실점 변화

(a)

(b)

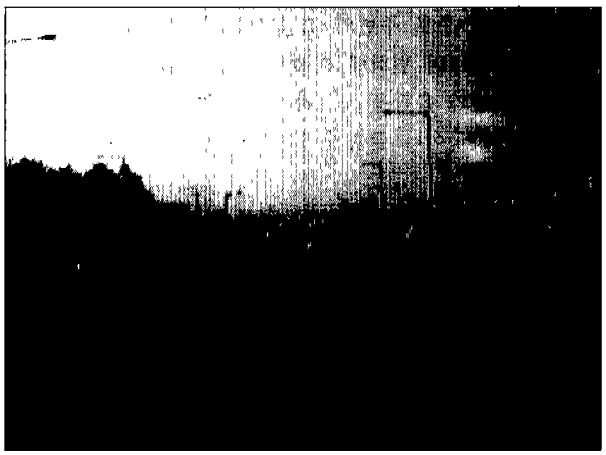


(c)

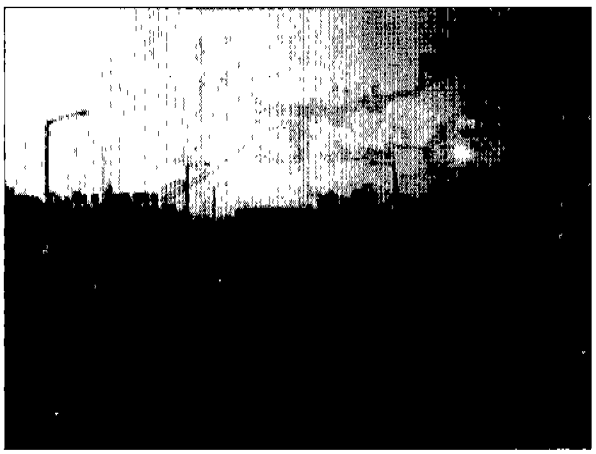
(그림 16) 차선 검출 결과

차선을 검출하는데 있어서 도로 영상의 전 부분을 사용하더라도 차량이나 카메라에서 멀리 떨어진 부분에서는 차선의 정보를 정확히 파악 할 수 없다. 도로에 차량이 있는 상황에서 차선을 검출하는데는 영상의 전 영역을 이용하지 않고 다른 차량에 의해서 방해받지 않는 제한된 영역만을 이용하더라도 좋은 효과를 보이고 있으며, 처리시간 면에서도 좋은 성능을 보이고 있다.

처리 시간은 약 10.7ms가 소요되고, <표 1>에서 보인바와 같이 전처리 과정에서 5.4ms, 차선을 검출하는데 5.3ms의 처리 시간이 소요된다. 다른 방법들에 비해서 처리 시간이 전체 영상의 10%에 해당하는 영역만을 이용하기 때문에 처리 속도면에서 보면 아주 좋은 성능을 보이고 있다.



(a) 차선의 끝 부분에서의 에러



(b) 기울기 측정단계에서의 에러

(그림 17) 기울기 측정 에러

〈표 1〉 차선 검출율 및 수행 시간

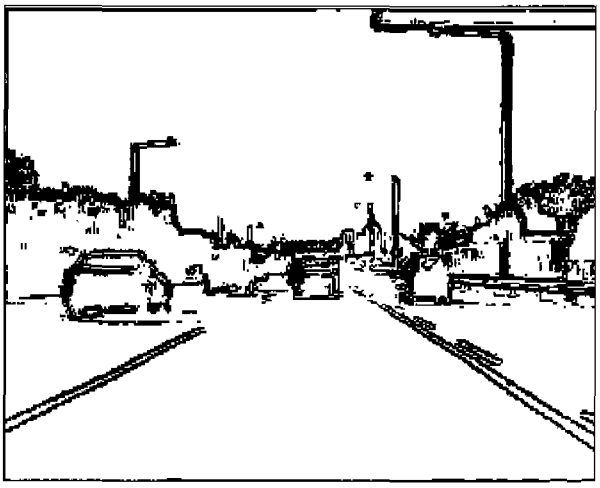
차선 검출	검출율			처리시간(ms)		
	프레임수	정확도	에러율	전처리	차선검출	전체
	2000	98 %	2 %	5.4	5.3	10.7

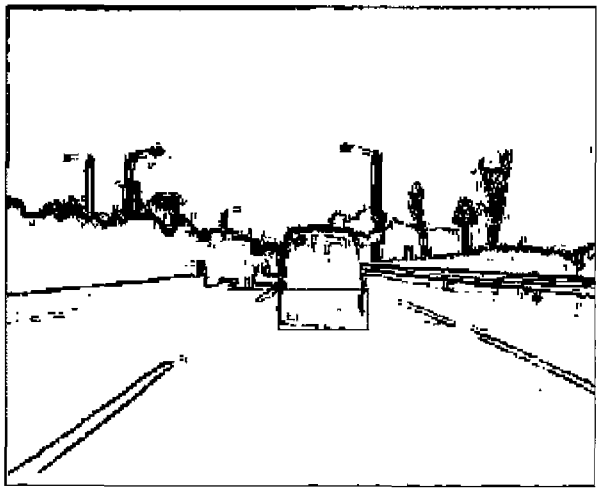
5.3 차량 포착 결과

차량의 포착 결과는 약 92%의 포착율을 보이고 있으며, 결과 영상은 (그림 18)과 같다. 차선 내에 있는 차량만을

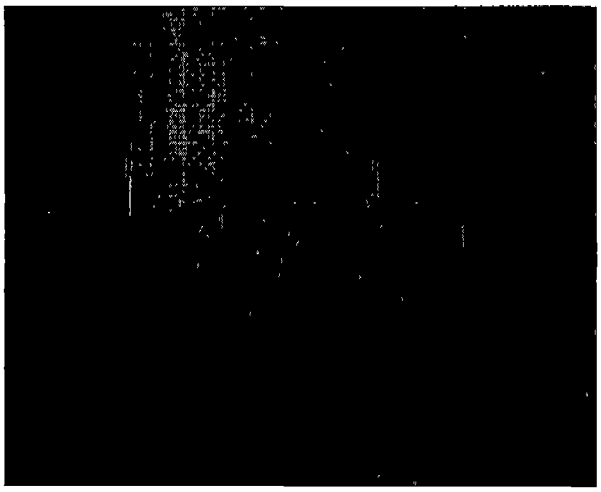
검출한 결과로써 차량 후미의 시작 위치만을 구했다. 그리고 다양한 환경에 대해서 모든 영상을 취할 수 없기 때문에 올림픽도로에서 임의로 위치에서 입력된 영상을 이용하였다. 차량의 위치는 사각형을 이용하여 위치를 표시하였다. 차량의 검출결과 또한 정량적 분석이 어려워 시각적으로 판별하였으며, 차량 후미 부분의 시작 위치를 기준으로 하여 오류를 판별하였다. (그림 19)는 차선, 관심 영역과 장애물을 검출한 결과를 보이고 있다. (a)는 차선과 관심 영역을 표시하였고, (b)는 차량의 검출 결과를 표시하였다. 차량 검출 시에 도로 주변의 그림자나 차량에 의한 그림자와 같은 주변 환경의 변화에 의해서 에지 점들이 서로 연결되어 있어 차량의 윤곽을 구분하지 못하는 경우가 있고, 도로상에 바퀴 자국이나 글씨가 있는 경우에도 오 추출되는 경우가 있다.

처리 시간을 구해보면 약 94.5ms 정도이고 단계별 소요 시간은 <표 2>에 나타내었다. 차량을 포착하기 위해서 차선을 검출하였으며, 검출된 차선에서의 전방 차량을 포착하는데 소요되는 시간으로써는 좋은 결과라 할 수 있다.

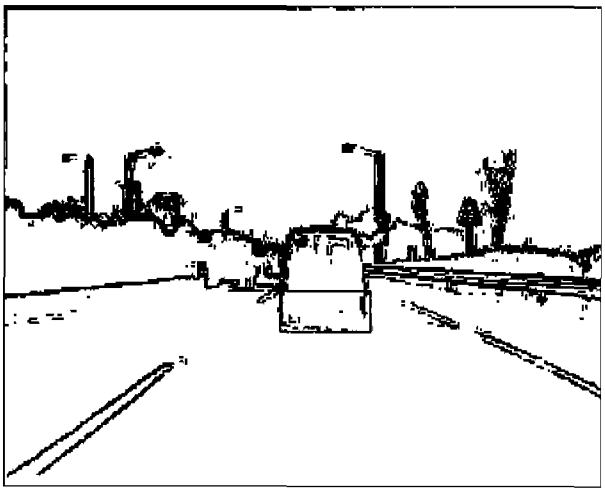




(그림 18) 차량 검출 결과



(a) 차선, 관심 영역



(b) 차량

(그림 19) 차선, 관심 영역, 차량 검출 결과

〈표 2〉 차량 포착율 및 수행 시간

차량 포착	포착율			처리시간(ms)				
	프레임수	정확도	에러율	전처리	차선검출	관심영역	차량포착	전체
	2000	92 %	8 %	43.2	10.7	0.3	40.3	94.5

6. 결 론

안전 운전에 도움을 줄 수 있는 시스템의 일부분으로 빠른 시간에 도로 영상에서 차선을 검출할 수 있으면, 전방의 차량을 보다 효율적으로 검지 할 수 있다. 요즘 내장형 시스템을 통하여 많은 장비들이 나오는데 별도의 DSP를 사용하지 않고서도 전방 차량을 검출하여 운전자들이 운전하는데 있어서 도움을 줄 수 있을 것으로 기대된다. 기존의 다른 여러 방법들이 있으나 이들 방법들은 별도의 DSP를 이용하여 고가의 장비가 요구된다. 하지만 USB 포트를 이용하여 영상을 캡춰(capture)하는 PC 카메라를 이용하여, 해

상도가 떨어지더라도 빠른 시간 내에 차선을 검출하고, 전방 차량을 검지할 수 있어 저가의 비용으로 차량에 장착하기 쉬울 것이다. 대부분의 차선을 검출하는 방법들은 전체 영역을 이용하는데 비해 다른 차량에 의해 방해받지 않는 제한된 영역의 정보만을 이용하여 빠른 시간에 차선을 검출함으로써 실시간 처리에 적합하다. 차선 정보를 이용하여 전방의 차량을 정확히 잦을 수 있다면 차후 차량과의 거리와 상대 속도를 이용하여 운전자에게 안전거리 미 확보에 대한 경고를 줄 수 있어 차량간의 추돌 사고를 미연에 방지할 수 있는 시스템이 될 것이다. 차후 연구사항으로는 하나의 CCD 카메라로는 전방의 차량을 검지하기가 어려운데 컬러 영상을 이용하여 전방 차선이 아닌 모든 영역에 있는 차량을 검출하고, 도로의 환경에 강인한 방법을 적용하여 스마트 카를 구성하는 것이다.

참 고 문 헌

[1] Swee Meng Wong, Ming Xie, "Lane Geometry Detection for the Guidance of Smart Vehicle," Proceedings of the IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems, Tokyo, Japan, pp.925-928, 1995.

[2] Youngguk Yim, Se-Young Oh, "Three-Feature Based Automatic Lane Detection Algorithm(TFALDA) for Autonomous Driving," Proceedings of the IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems, pp.929-932, 1999.

[3] Stefan Ernst, Christop Stiller, Jens Goldbeck, Christoph Roessig, "Camera Calibration for Lane and Obstacle Detection," Proceedings of the IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems, pp.356- 361, 1999.

[4] C. Stiller, J. Hipp, C. Rossig, A. Ewald, "Multisensor obstacle detection and tracking," Image and Vision Computing 18, pp.389-396, 2000.

[5] Takeo Kato, Yoshiki Ninomiya, "Preceding Vehicle Recognition based on learning from sample images," The fifth ICARCV '98, singapore, 9-11, pp.1632-1636, December, 1998.

[6] Nobuhiro TSUNASHIMA, Masato NAKAJIMA, "Extraction of the Front Vehicle using Projected Disparity Map," Conference Visual Communications and Image Processing '99, califonia, pp.1297-1304, January, 1999.

[7] Xuan Yu, Serge Beucher and Michel Bilodeau, "Road Tracking Lane Segmentation and Obstacle Recognition by Mathmetical Morphology," Intelligent Vehicles '92 Symposium., Proceedings of the, pp.166-172, 1990.

[8] Sholin Kyo, Takuya Koga, Kazuyuki Sakurai, Shin'ichiro Okazaki, "A Robust Vehicle Detecting and Tracking System for Wet Weather Conditions using the IMAP-VISION Image Processing Board," Proceedings of the IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems, pp.423-428, 1991.

[9] Juan Pablo Gonzalez, Umit Ozguner, "Lane Detection Using Histogram-Based Segmentation and Detetion Trees," IEEE Intelligent Transportation Systems Conference Proceedings Dearborn, USA, pp.346-351, October, 2000.

[10] Ernst Lissel, Peter Andreas, Ralf Bergholz, Hubert Weisser, "From Automatic Distance Regulation to Collision Avoidance," AVEC '96, International Symposium on Avoidanced Vehicle Control, pp.1367-1378, 1996.

[11] Gian Luca Foresti, Vittorio Murino, Carlo Regazzoni, "Vehicle Recognition and Tracking from Road Image Sequence," IEEE Transaction on vehiclular technology, Vol. 48, No.1, January, 1999.

[12] Massimo Bertozzi, Alessandra Fascioli, Alberto Broggi, "Performance Analysis of a Low-Cost Solution to Vision-Based Obstacle Detection," Proceedings of the IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems, pp.350-355, 1999.

[13] 노광현, 한민홍, "저속주행 환경에서 컬러비전 기반의 근거리 전방차량추적", 정보처리논문지, 제7권 제9호, pp.3037-3047, 2000.

[14] M. Werner, W. V. Seelen, "An Image processing system for assistance, U. Handmann, T. Kalinke, C. Tzomakas," Image and Vision Computing 18, pp.367-376, 2000.

[15] 이응주, "그룹화 블록 스네이크 알고리즘을 이용한 차선추출", 멀티미디어학회논문지, 제3권 제5호, pp.445-453, 2000.

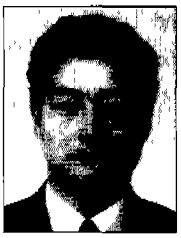
[16] Rong YANG, Francois CABESTAING, Jack-Gerard POSTAIRE, "Obstacle Detection in a Sequence of Rearview Images," International Conference on Information, Communications and signal processing, ICICS '97, singaporc, 9-12, pp.200-204, September, 1997.

[17] S. M. Smith and J. M. Brady, "ASSET-2 : Real-Time Motion Segmentation and shape Tracking," IEEE Trans. Pattern Analysis And Machine Intelligence, Vol.17, No.8, pp. 814-820, August, 1995.

[18] 정전익, 최성구, 노도환, "무한원점을 이용한 주행방향 추정과 장애물 검출", 전자공학회논문지, 제34권 제11호, pp.1302-1313, 1997.

[19] 한국도로공사 고속도로 시설물 관리기준.

[20] 권화중, 이준호, "전방 장애물 검출을 위한 효율적인 차선 인식 알고리즘", 한국정보과학회 봄 학술발표회논문집, Vol.26, No.1, pp.573-575, 1998.



한 상 훈

e-mail : hansh@dongguk.edu

1990년 동국대학교, 전자계산학과 졸업 (학사)

1990년~1993년 공군 학사 장교 근무

1995년 동국대학교 대학원 컴퓨터공학과 (공학석사)

1995년~현재 동국대학교 대학원 컴퓨터공학과(박사수료)

관심분야 : 멀티미디어 정보처리, 컴퓨터비젼, 형태인식, ITS, 게임

조 형 제

e-mail : chohj@dongguk.edu

1973년 부산대학교, 전자공학과(학사)

1975년 한국과학기술원, 전기 및 전자 공학과 대학원(공학석사)

1975년~1982년 금성통신(주)연구소 실장

1986년 한국과학기술원, 전기 및 전자 공학과 대학원(공학박사)

1986년~현재 동국대학교 영상정보통신대학원 멀티미디어학과 교수

관심분야 : 멀티미디어 정보처리, 컴퓨터비젼, 컴퓨터 그래픽스 형태인식, 게임