

그림자 정보를 이용한 전방 차량 검출 시스템 구현

한 상훈, 조형제
동국대학교 컴퓨터공학과

Implementation of a Front Vehicle Extraction System with Shadow Information

Sanghoon Han, Hyungje Cho
Dept. of Computer Engineering Dongguk University
E-mail : hansh@dgu.edu, chohj@dgu.edu

요약

차량이 증가함에 따라서 첨단 교통 시스템(intelligent Transportation System: ITS)은 교통의 효율성, 신뢰성, 안정성 향상에 중점을 두게 되었다. 첨단 교통 시스템의 일부분인 운전자 도움 시스템(Advanced Drivers Assistance System)은 운전을 하고 있는 상황에서 도움을 주기 위한 체계이고, 전방의 장애물을 찾는 운전자 도움 시스템에서 전방의 상황을 운전자에게 알려주기 위한 중요한 요소이다. 본 논문에서는 HSV 컬러모델을 이용하여 연속된 컬러 영상으로부터 도로상의 차선과 방향 표시자에 구애받지 않고 전방의 차량을 검출하는 방법을 제안한다.

HSV 컬러 모델에서 차량을 검출하기 위해서는 채도(Saturation)와 명도(Value)성분의 관계를 이용하여 차량 영역을 구하고, HSV성분과 위치 특징을 이용하여 이전 프레임의 차량인지 검증한다. 도로 영상에서 차량이 있는 경우 차량의 아래 부분에 그림자 영역이 존재한다는 점을 이용한다. 제안된 방식의 효과를 검증하기 위해 노트북 PC와 PC용 CCD 카메라로 도로에서의 영상을 촬영하고 차량검출알고리즘을 적용한 처리 시간, 정확도, 차량검지 등의 결과를 보인다.

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[1].

운전자 도움 시스템(Advanced Driver Assistance System : ADAS)은 안전하고, 효율적인 운전을 하는데 도움을 줄 수 있는 보조 장치를 제공한다.

전방의 차량의 검출 및 추적은 능동형 센서인 레이저레이더(Laser Radar)나 밀리미터 웨이브 레이더(Millimeter Wave Radar)센서를 이용하는 연구들과 여러 가지 장치를 혼합한 방법들이 있다[2]. 수동 센서인 카메라 센서를 이용한 연구들도 다양하게 이루어지고 있다. 스테레오 비전을 이용하는 방법과 그레이 영상을 이용하는 방법, 컬러 영상을 이용하는 방법들이 있다[3, 4, 5, 6].

컬러 영상을 이용하는 방법들에는 컬러 정보를 몇 개의 컬러로 양자화 한 후에 레이블링을 통해 영역을 분할한 후에 프레임간에 같은 영역을 찾아 차량을 탐지하는 방법[7]이 있다. 컬러, 에지, 모션 정보를 이용하여 실시간 추적을 통하여 차량을 인식하는 방법[8], 도로의 영역을 미리 지정해 두어 도로 영역인지 그 외의 영

역인지 판별하는 방법[9]이 있다. 이러한 방법들은 도로에 차량이 많거나 도로 영상이 복잡한 경우에 문제가 있는 것은 사실이다. 도로에 차량이 많은 경우에 차량의 영역을 구분하기 어려워지며, 컬러로 양자화 한 경우에는 차량에 후미등이나 번호판과 같은 다른 색깔 정보를 가지고 있기 때문에 주위의 다른 차량과 구분하기가 어려워진다. 또한 도로에 차량에 의한 그림자가 생기는 경우에 차량 검출 시에 여러의 소지가 높다.

또 다른 방법으로 차량의 후미등을 이용하여 전방의 차량을 탐지하는 방법[10]이 있다. 이 방법은 차량의 거리가 가까운 경우에는 좋은 성능을 발휘하지만 차량의 거리가 멀어지는 경우에 후미등의 색깔 정보를 찾기가 어려워진다.

본 논문에서는 교통흐름이 복잡하고, 도로 상에 도로 표시들이 다양하게 있는 상황에서도 CCD 카메라로 연속으로 촬영한 일련의 컬러 영상으로부터 전방의 차량을 검출하는 방법을 제안한다. 컬러모델은 HSV 컬러 모델을 기반으로 하고, 차량에는 차량의 밑 부분에 그림자 영역을 가지고 있고, 교통흐름이 복잡하더라도 도로의 영역이 주를 이룬다는 점을 이용하였다. 교통량이 많은 경우에 전방에 보이는 차량이 많아 배경부분이 수시로 변화하고, 도로상에 도로 표시들이 있는 경우에는 차량을 탐지하기가 어렵다. 하지만 본 연구에서는 교통량이 많고, 도로상에 도로 표시가 있는 경우에도 전방의 차량 검출이 가능하다.

본 논문의 구성은 2장에서 시스템 구성 및 컬러 모델을 소개하고, 3장에서는 전처리와 후보 점 검출 방법, 4장에서는 전방의 차량 검출 방법, 5장에서는 실험 결과를 제시하고, 6장에서 결론을 맺는다.

2. 시스템 구성 및 컬러 모델

컬러 CCD 카메라를 이용하여 전방 차량을 탐지하는 방법은 그림 1과 같다. 먼저 카메라로부터 RGB 영상이 입력으로 들어오면 HSV 컬러 공간으로 변환된다. 그리고 변환된 컬러 공간에서 색조(Saturation)와 명도(Value)성분 차를 적용한 전처리 과정을 거치고, 초기 설정과정에서 지정한 도로 영역의 평균값과의 차이가 많이 발생하는 화소를 차량의 후보 점으로 간주한다. 그리고 5*5 블록으로 나눈 뒤에 차량의 후보 영역을 구하여 예지 정보를 이용하여 차량의 영역을 결정하고, 전방 차량의 인식을 통해서 전방 차량이

제대로 찾아지는지 확인한다.

컬러 모델은 일반적으로 RGB, CMYK 등이 널리 알려져 있으나 본 연구에서는 HSV 컬러 모델을 이용한다. HSV 컬러 모델은 Hue(색조), Saturation(채도), Value(명도)로 이루어진다. 그림 2에 HSV의 육각뿔의 컬러 모델이 잘 나타나 있다.

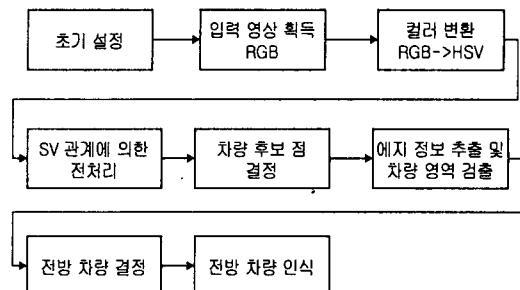


그림 1 시스템 구성 및 처리과정

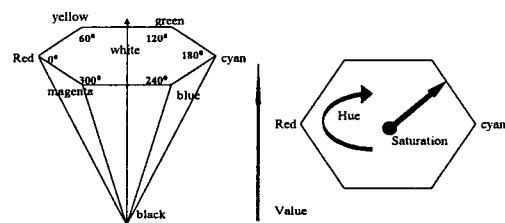


그림 2 HSV 컬러 모델

도로 영상에 대해서 일련의 연속된 영상을 촬영하여 보면 카메라로 들어오는 빛의 양에 따라서 영상의 전체에 영향을 준다. 카메라가 동적으로 움직이기 때문에 카메라에서 바라보는 시각이 다르고 전방의 물체 또한 계속 이동하기 때문에 도로의 색상모델을 정하기 어렵다. 도로 영상에 대한 모델을 정하더라도 도로의 상황이 다양하게 변하기 때문에 같은 도로라도 다른 특징을 가지는 경우가 있다. 그리고, 교통량이 많은 경우와 도로 위에 표시가 있는 경우, 차선이 있는 경우, 주위의 배경에 의해서 그림자가 드리워져 있는 경우에 차량을 검출하는데 문제점들이 발생한다.

본 논문에서는 도로상에 차량이 있는 경우에 차량의 아래 부분에 그림자(Cast Shadow)를 갖고 있다는 점을 이용한다. 차량이 도로 바닥에 붙어서 다니는 것이 아니라 어느 정도 떨어져서 다니기 때문에 그림 3과 같이 물체에 의해 빛이 가려지는 부분이 생기게 된다. 그림 3에서 보면 암흑부(Umbra)는 빛이 차단된 상태이고, 낮은 밝기 값을 갖는다. 반암부(penumbra)는 암흑부와 같이 빛이 차단된 상태는 아니고 점차적으로 밝기

값이 변하는 부분이다[11]. 차량의 높이가 너무 높아, 빛의 방향이 물체의 위치보다 낮은 경우에는 이런 암흑부가 생기지 않는 경우가 발생하지만 대부분의 경우에는 암흑부가 발생한다. 기본적으로 도로 상에 차량이 있는 경우에는 도로 상에 암흑부가 존재하며, 도로 상에서 이 부분을 포착한다면 차량의 유무를 탐지할 수 있다.

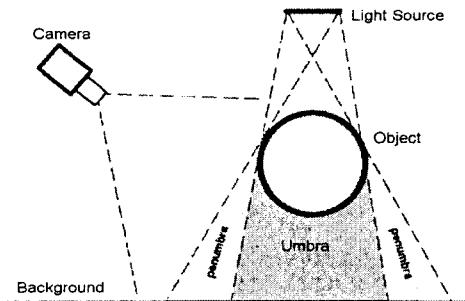


그림 3 그림자 모델

그림 4에서는 컬러 영상을 HSV 컬러 공간으로 변환하여 (a)는 S 성분, (b)는 V성분을 나타낸다. (a)에서 S성분을 보면 그림자가 있는 부분 즉 밝기가 낮은 부분에서 큰 값을 가지고 있음을 볼 수 있다. 차량이나 주위 배경에 의해서 생기는 그림자는 반암부에 해당하고 암흑부에 비해 채도와 명도의 차이가 많이 나지 않기 때문에 대부분은 제거가 가능하다.



(a) S 성분

(b) V 성분

그림 4 HSV 모델에서의 S와 V 성분

3. 전처리 및 차량 후보 점 검출

영상 입력부에서 RGB 값으로 입력된 영상에 대해서 먼저 HSV 컬러 공간으로 변환한다. 그리고 HSV 성분에서 S와 V성분을 이용하여 S 성분과 V성분의 차이를 구한다. 이때 S성분이 [0:1]사이의 값을 갖기 때문에 값을 [0:255]사이의 값으로 변환한 뒤 차이 값을 구한다. 여기서 도로에 해당되는 영역을 초기화 과정에서 지정하기 때문에 해당영역의 평균을 구하여 평균과의 차이

가 문턱치 이상이면 차량의 후보 점으로 지정한다. S성분과 V성분의 차이 값을 갖는 영상을 $F(x,y)$ 로 정하여식 1과 같이 구한다.

$$F(x,y) = S(x,y) - V(x,y) \quad (1)$$

여기서 $S(x, y)$ 는 S성분, $V(x, y)$ 는 V성분을 가리킨다.

이 $F(x, y)$ 값을 이용하여 초기 설정단계에서 지정한 영역 평균과의 차이값과 $F(x, y)$ 에 따라서 차량의 후보 점을 구한다.

$$R(x,y) = \begin{cases} 1, & |mean - F(x,y)| > T_1 \text{ and } F(x,y) \geq T_2 \\ 2, & |mean - F(x,y)| > T_1 \text{ and } F(x,y) < T_2 \\ 0, & \text{else} \end{cases} \quad (2)$$

여기서 mean은 영역 평균을 의미하고, T_1 과 T_2 는 임계값이다. $R(x,y)$ 가 1이면 차량의 후보 점에 해당하며, 2나 0이면 도로 및 배경 영역으로 간주한다. 이때 T_1 과 T_2 는 경험적 수치로 약 200여 개의 프레임 정보를 이용하여 가장 균일한 수치로 결정하였다. 차량의 후보 점을 표시한 결과는 그림 5와 같다.

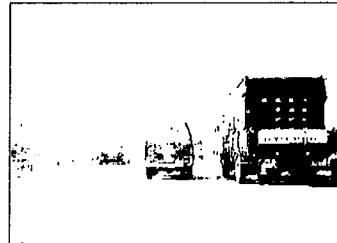


그림 5 차량의 후보 점

4. 전방의 차량 검출

전방에 있는 차량을 검출하기 위해서는 S와 V성분을 이용하여 에지 맵을 구하고, 앞에서 구해진 차량의 후보 점을 이용한다. 에지 맵은 Prewitt 에지 연산자[12]를 이용하여 구하였으며, 차량의 영역은 ROI 내에서만 검색한다. 그 이유는 ROI 범위 외에 있는 부분은 대부분이 차량의 후면이 보이지 않거나 차량의 일부분만 보이는 부분이기 때문이다.

4.1 차량 영역 검출

차량에 대한 후보 점들이 구해지면 차량의 영역을 구하기 위해 전체 영역을 5*5 블록으로 나눈다. 블록으로 나누는 이유는 화소단위의 처리 시에 작은 영역들이 많이 발생하기 때문에 블록단위의 처리를 통해서 작은 영

역을 제거하고 근접 영역에 대해서는 병합 효과가 있으며, 처리 시간을 줄일수 있다.

차량 영역은 ROI내에 있는 영역만을 대상으로 하며, 한 블록이 차량의 후보 블록인지 결정하는 것은 식 3과 같이 5*5 블록 내에 일정이상의 화소가 있으면 1, 없으면 0으로 표시한다. 이렇게 정의된 블록을 이용하여 차량 영역이 시작하는 위치를 찾는다.

$$B(x,y) = \sum_{i,j}^{i=5, j=5} R(x,y) \quad (3)$$

$$B'(x,y) = \begin{cases} 1 & \text{if } B(x,y) > T_3 \\ 0 & \text{else} \end{cases}$$

그림 6에서는 $B'(x, y)$ 가 1인 블록을 흰색으로 표시하였으며, ROI 기준선과 차량의 후보 블록을 표시하였다.

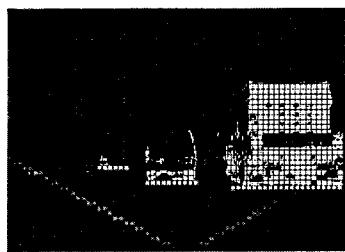


그림 6 블록 처리의 결과

차량의 후보 블록이 차량 밀 부분에 형성되어 있기 때문에 거리에 따라서 후보 블록의 위치가 다르다. 여기서는 Y축으로 가장 가까운 후보 블록들이 가장 가까운 차량의 시작 위치가 된다. 그래서 후보 블록들 중에서 수평으로 연결되어 있고, 가장 가까운 블록을 찾으면 차량의 후보 영역이 된다.

먼저 영상의 아래에서 시작하여 차량 영역에 해당하는 블록을 먼저 찾고, 그 블록에서 좌우로 연결된 블록을 추적한다. 이때 상하로는 1블록이상 넘어가지 않도록 한다. 그림 7에서 보면 이웃 블록을 추적하는 과정을 보여준다. 이때 블록이 4개 이상 연결되어 있으면 차량의 후보 영역으로 간주하고, 그 후보 블록의 수평 길이의 비를 고려하여 높이를 결정한다. 이렇게 구해진 영역 안에 수평, 수직 예지 성분이 존재하는지 보고, 존재한다면 차량 영역으로 결정하고, 그 위에 있는 후보 블록들은 검색 대상에서 제외한다. 이유는 차량의 위에 있는 것들은 대부분이 배경이거나 그 차량보다 멀리 떨어져 있는 경우이기 때문이다.

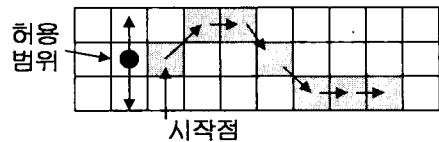


그림 7 이웃 블록 추적

차량 영역이 구하고 블록 단위의 차량 영역을 화소 단위의 차량 영역으로 바꾼다. 이때도 수평, 수직 예지 성분을 이용하여 세밀하게 차량의 영역을 결정한다.

4.2 전방 차량의 결정 및 인식

차량의 영역이 구해지면, 차량 영역 중에 전방 차량에 해당하는 영역을 결정하는데 차선 정보를 구한다면 왼쪽 차선과 오른쪽 차선 사이에 있는 차량을 전방 차량이라고 결정을 하지만, 본 연구에서는 차량의 중심이 카메라의 중심에 가장 가까운 영역을 전방 차량으로 지정하였다.

전방 차량의 인식은 차량이 차선을 바꾼 다거나 자기 차량이 차선을 바꾸는 경우에 전방 차량의 위치가 달라지게 되는데 이때 상황이 정상적임을 판단해야 한다. 이전 프레임에서의 전방 차량과 현재 프레임에서의 전방 차량이 같은 차량인지 확인하는 방법은 컬러 성분의 차이와 영역 중심점의 차이를 이용한다.

$$Dr_{ab} = \sqrt{(Mh_a - Mh_b)^2 + (Ms_a - Ms_b)^2 + (Mv_a - Mv_b)^2}$$

$$Dc_{ab} = \sqrt{(Dc_a - Dc_b)^2}$$

$$DT_{ab} = w_1 * Dr_{ab} + w_2 * Dc_{ab}$$

Dr_{ab} 는 HSV에서 각 성분의 평균값의 차이이고, Dc_{ab} 는 중심점의 위치간의 거리이다. DT_{ab} 가 최소가 되는 영역을 이전 프레임에서의 전방 차량 영역과 동일한 차량 영역으로 정한다.

5. 실험 결과 및 분석

5.1 실험 환경

실험은 올림픽도로상에서 USB 방식의 PC 카메라를 이용하여 초당 10프레임으로 영상을 캡쳐 하였으며, 영상의 크기는 320*240을 사용하였다. 실험 시에 카메라

의 위치는 차량의 운전석 앞부분에 설치하였으며, 카메라의 위치 보정을 하거나 수평을 유지도록 하는 보조 장치는 사용하지 않았다. 영상은 약 5분씩 5번 캡쳐 하였으며, 프레임의 수는 14000여 프레임을 이용하였다.

5.2 실험 결과

전방차량 검출에 대한 결과로는 표1과 같다. 전방 차량에 대한 검출의 정확도를 구해보면 약 91.24%의 검출율을 보이고 있다. 실험에서 정확도를 다른 시스템과 환경이 달라 비교가 불가능하고, 수치상으로 표현할 수 없기 때문에, 전방 차량을 검출한 외곽 사각형이 제대로 차량의 위치에 위치하는지 확인하는 방법을 이용하였다.

프레임 수	에러(frame)	에러율(%)	정확도(%)
14006	1227	8.76	91.24

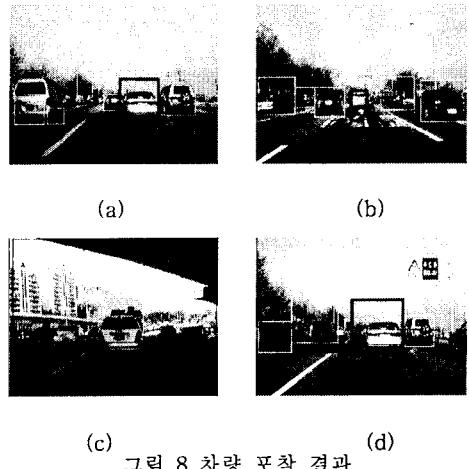
표 1 전방 차량 검출 결과

처리시간은 Pentium III 700MHz CPU를 갖는 노트북에서 약 220ms 정도의 시간이 소요되어 초당 약 4 프레임 정도를 처리할 수 있었으며, 대부분의 영상에서 위치를 전방 차량의 위치를 잘 포착하고 있으나, 그림자가 심하게 드리워져 있는 경우와 전방 차량의 영역이 앞 차량의 영역과 같이 병합되어 크게 추출되는 경우가 있었다.

그림 8에서 결과 영상을 보여주고 있는데 (a)는 그림자가 있는 경우에 전방 차량의 하단 부분을 정확히 포착하고 있으며, (b)는 도로 위에 차선 외의 표시가 있는 경우이다. 그리고 (c)는 고가 도로 아래를 지나는 경우에 차량을 포착하지 못하는 결과이고, (d)는 원래 차량의 영역보다 크게 포착되는 경우이다. 차량이 고가도로 아래를 통과하는 과정에서 보면, 고가도로에 의해서 빛을 받지 못하는 부분이 발생하는데 전방의 모든 영역이 차량의 영역으로 인식되는 경우이다.

그림 9는 전방 차량이라고 결정된 영역의 Y좌표 값과 이전 프레임의 전방 차량의 Y좌표값의 차이를 전방 차량이 차선 변경을 하는 과정 중 약 150frame에 대한 그래프를 그려보았다. 그림 10은 같은 차량을 추적하여 Y좌표값의 차이를 보았다. 이때 차량의 인식에 의해서 같은 차량을 찾는 경우의 Y좌표값의 변화는 거의 없는데 비해서 전방 차량만을 이용한 경우에는 차선을 변경하는 과정에서 많은 오차를 보이고 있다. 이것은 전방 차량을 결정하는 방법에서 차선 정보를 활용하

지 못하고 있으며, 차량의 영역의 위치가 조금씩 바뀌어감에 따라서 전방 차량의 영역이 변경될 수 있기 때문이다.



(a) (b)
(c) (d)
그림 8 차량 포착 결과

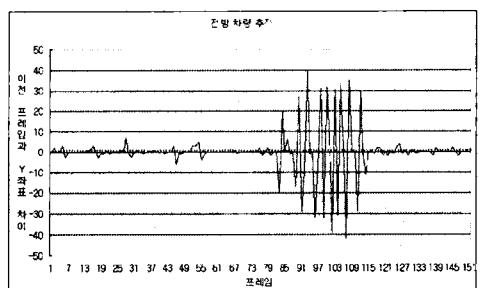


그림 9 전방 차량의 Y좌표

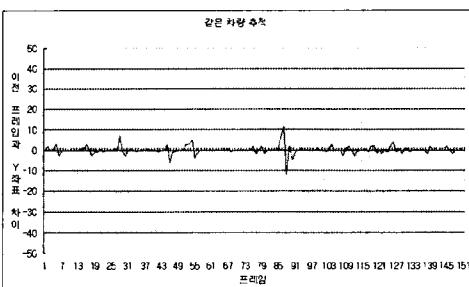


그림 10 같은 차량의 Y좌표

6. 결 론

컬러 영상에서 채도와 명도 성분을 이용하여 도로 영상에서 차량을 검출하고 차선의 정보를 추출하는 방법

을 알아보았다. 채도 성분과 명도 성분의 차이를 이용하여 배경과 차량 후보 점을 구분하였고, 차량의 후보 점에서 차량의 후보 영역의 시작 위치를 구하고 에지 맵을 이용하여 차량의 영역을 구하였다.

카메라의 수평을 유지하도록 하는 장치와 같이 별도의 장치를 사용하지 않고, USB 포트를 이용한 PC 카메라를 통해 영상을 입력받아 처리하기 때문에 차량의 거리를 계산해내기는 어렵지만 전방 차량이 멀어지고 있는지, 가까워지고 있는지는 판단할 수 있었다. 운전자는 전방의 차량을 따라가면서 거리를 유지하고자 하는 속성이 있어서 연속된 영상을 이용하는 방법이 좋은 효과를 보기 위해 단일 영상을 이용하여 전방 차량을 검출하였다.

도로 바닥에 방향 표시나 낮은 요철이 있는 경우에도 좋은 결과를 보이고 있지만, 주위의 환경에 따라 결과가 달라질 수 있는 것은 사실이다. 특히 차량의 속도에 관계없이 처리할 수 있다는 장점이 있다. 그리고 별도의 DSP 장비가 없어도 초당 약 4프레임을 처리 할 수 있는 능력을 보임으로써 저가의 하드웨어를 이용하여 처리할 수 있는 기반이 될 수 있으리라 본다.

향후 연구과제로는 차선의 검출방법과 악천후에서의 차량 검출 방법과 야간에서의 처리 방법 등이 연구되어야 할 것이고, 운전자 도움 시스템에서 도로 표지판과 같은 물체의 인식을 통하여 도로의 정보를 영상 처리를 통하여 수집할 수 있는 방법의 연구가 되어야 할 것이다.

[참고문헌]

- [1] 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.
- [2] C. Stiller, J. Hipp, C. Rossig, A. Ewald, "Multisensor obstacle detection and tracking", Image and Vision Computing 18, pp 389-396, 2000.
- [3] Dong Guo, Thierry Fraichard, Ming Xie, Christian Laugier, "Color Modeling by Spherical Influence Field in Sensing Driving Environment", Proceedings of the IEEE Intelligent Vehicles Symposium 2000, pp 249-254, 2000
- [4] Nobuhiro TSUNASHIMA, Masato NAKAJIMA, "Extraction of the Front Vehicle using Projected Disparity Map", Conference Visual Communications and Image Processing '99, California, January, pp 1297-1304, 1999.
- [5] 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, 1995.
- [6] Ernst Lissel, Peter Andreas, Ralf Bergholz, Hubert Weisser, "From Automatic Distance Regulation to Collision Avoidance", AVEC' 96, International Symposium on Avoidance Vehicle Control, pp 1367-1378, 1996.
- [7] B. Heisele, W. Ritter, "Obstacle Detection Based On Color Blob Flow", Proceedings of the Intelligent Vehicles '95 Symposium., pp 282 -286 ,1995
- [8] Betke, M., Haritaoglu, E., Davis, L.S., "Highway Scene Analysis in Hard Real-Time", Intelligent Transportation System, 1997. ITSC '97., IEEE Conference on , pp 812 -817, 1997
- [9] De Micheli, E., Prevete, R.; Piccioli, G., Campani, M., "color cues for traffic scene analysis", Intelligent Vehicles '95 Symposium., Proceedings of the , pp 466 -471, 1995
- [10] 노광현, 한민홍, "저속 주행 환경에서 월라비 전기반의 근거리 전방차량추적", 한국 정보처리학회 논문지, 제7권 제9호, pp 3037-3047, 2000
- [11] Jurgen Stauder, Roland Mech, Jorn Ostermann, "Detection of Moving Cast Shadows for Object Segmentation", IEEE Transaction on Multimedia, Vol. 1, No. 1, pp 65-76, 1999
- [12] J. Scharcanski, A.N. Venetsanopoulos, "Edge Detection of Color Images Using Directional Operators", IEEE Trans. Circuits And System For Video Technology, Vol. 7, No. 2, pp 397-401, 1997