

그림자 정보를 이용한 HSV 컬러 모델 기반의 전방 차량 검출 및 차선 정보 검출

한상훈[†] · 조형제^{**}

요 약

차량이 증가함에 따라 전방의 상황을 운전자에게 알려주기 위한 운전자 도움 시스템(Advanced Drivers Assistance System)과 같은 체계가 요구된다. 본 논문에서는 전방의 상황을 운전자에게 알려주기 위한 기본 과정으로 연속된 컬러 영상으로부터 영상 처리만을 이용하여 전방의 차량과 차선을 검출하는 방법을 제안한다. 도로 전방의 상황은 차량이 많다고 하더라도 도로의 영역이 많은 부분을 차지하고 있으며, 차량이 있는 경우에 차량의 하단에 그림자와 같이 어두운 영역이 존재하는 점을 이용하여 전방의 차량을 검출한다. 그리고 차선은 그림자 영역의 반대 특징으로 흰색 계열이라는 점을 이용하여 차선 정보를 추출한다. 이 방법은 도로가 혼잡하거나 도로상에 방향 표시가 있는 경우에도 좋은 결과를 보인다.

차량과 차선을 검출하는데는 HSV 컬러 모델에서 채도 성분과 명도 성분을 이용하여 후보 점을 검출하고, 차량과 차선의 영역을 검출하며 에지 정보를 이용하여 차량의 영역을 결정한다. 그리고 검출된 차량 영역이 이전 프레임의 차량 영역과 같은 차량인지 알기 위해서는 HSV 성분과 위치 정보의 통계적 특징을 이용한다. 제안된 방식의 효과를 검증하기 위해 노트북 PC와 PC용 CCD 카메라로 도로에서의 영상을 촬영하고 차량 및 차선 검출 알고리즘을 적용한 처리 시간, 정확도 및 차량검지 결과를 보인다.

HSV Color Model Based Front Vehicle Extraction and Lane Detection using Shadow Information

Han Sanghoon[†] and Cho Hyungje^{**}

ABSTRACT

According as vehicles increases, system such as Advanced Drivers Assistance System(ADAS) to inform forward situation to driver is required. In this paper, we proposes method to detect forward vehicles and lane from sequential color images by basis process to inform forward situation to driver. We detect a front vehicle using that shadow area exists on part under vehicles and that road area occupies many parts even if road traffic is confused. We detect lane information using that lane part is white order by reverse characteristic of shadow area. This method shows good result in case road is confused or there is direction indication to road.

HSV color space is selected for color modeling. This method uses saturation component and value component in HSV color model to detect vehicles and lane. It uses statistics features of HSV component and position to know whether detected vehicles area is vehicles such as vehicles previous frame. To verify the effects of the proposed method, we capture the road images with notebook and CCD camera for PC and present the results such as processing time, accuracy and vehicles detection against the images.

Key words: 영상처리(Image Processing), HSV 컬러(HSV Color), 그림자(Shadow), 차량 검출(Vehicle Detection), 차선 검출(Lane Detection), CCD 카메라(CCD Camera)

[†] 준회원, 동국대학교 시간강사

^{**} 정회원, 동국대학교 영상정보통신대학원 멀티미디어학과 교수

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 관련 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. 지능형 교통 시스템(Intelligent Transportation System)은 현재 교통 시스템의 안전성과 효율성, 신뢰성을 증가시키기 위해서 노력하고 있으며, 그 중의 한 부분이 바로 운전자 도움 시스템이다. ADAS는 복잡한 운전 환경속에서 운전자에게 운전 상황에 대한 제어, 실행 등을 처리하기 위해서 도움을 주는 시스템이다. ADAS의 기본 요소 중 하나인 전방 차량의 검출에 대한 연구는 하나의 센서를 이용한 접근 방법을 이용한 것이 상업화가 되어 있으나 현존하는 시스템들은 제한적인 것들이 많다. 게다가 하나의 센서를 이용하는 것은 복잡한 교통환경에서 높은 수준의 운행 환경을 관리하기에는 부족하다. 그래서 여러 가지 센서를 혼합한 방법들이 연구되고 있다.

전방 차량의 탐지 및 추적은 현재까지 능동형 센서인 레이저 레이더(Laser Radar)나 밀리미터 웨이브 레이더(Millimeter Wave Radar) 센서를 이용하는 연구들과 여러 가지 장치를 혼합한 방법들이 있다[2,3]. 이들은 고속 주행 상황에서도 100m 이상의 거리를 정확하게 탐지 할 수 있지만 고가의 가격 때문에 일반 차량에 장치를 장착하기 어렵다. 그리고 많은 수의 차량이 이런 장치를 장착한 경우에 신호 간

섭 및 여러 가지 부가적인 문제들이 발생할 소지가 있다.

수동 센서인 카메라 센서를 이용한 연구는 스테레오 비전과 모노 비전을 이용하는 방법, 컬러 영상을 이용하는 방법과 그레이 영상을 이용하는 방법으로 나눌 수 있다. 스테레오 비전을 이용하는 경우에는 대부분 disparity map을 구하여 장애물을 탐지한다[5,6]. 모노 비전을 이용하는 경우에는 대부분 그레이 영상을 이용하는데 광류(Optical Flow), 엔트로피(Entropy) 등 모션 분석(Motion Analysis)를 이용하고 있다[7,8]. 그레이 영상을 이용하는 방법들은 차량이나 주변환경의 그림자에 의한 영향을 많이 받는다는 것이 문제점이고, 이런 문제들 때문에 다수의 차량보다 한 두 대의 차량에 대해서 실험하는 경우가 많다. 차량의 하단부의 밝기가 낮다는 특징을 이용하여 다가오는 차량에 대한 검출방법[9]이 있고, 차량의 뒤 부분의 특징이 대칭이라는 점을 이용한 방법[10]도 있다. 차량의 하단부의 그림자 정보를 이용한 방법은 카메라가 이동하는 상황이 아닌 고정된 상황에서의 방법으로 차선을 통해 기준 영역을 결정하여 화소의 밝기가 확연히 낮은 부분을 찾는 방법이다. 그 외에 차선만을 별도로 탐지하는 연구들도 이루어지고 있다[4,11,12].

컬러 영상에서 모노 비전을 이용하는 방법도 또한 다양하게 연구되고 있는데 컬러 공간을 $L*a*b^*$ 를 이용하여 클러스터링을 통해 유사한 영역을 그룹화하고, 그 중 도로에 대한 클러스터 모델을 구하는 방법이 있다[3]. 이런 방법은 도로의 컬러 모델을 구하는데 도로의 컬러 모델이 주변 환경에 따라서 변화하는데 대응을 하기가 어려운 단점이 있다. 도로에서 테스트 영역을 지정하여 테스트 영역과의 차이를 이용하여 도로 영역을 재구성하고, 도로 위의 차선이나 장애물을 검출하는 방법이 있다[13]. 이 방법은 테스트 영역을 수작업으로 지정하고, 테스트 영역에 그림자가 있는 경우에 에러율이 높아지는 단점이 있다. 컬러, 에지, 모션 정보를 이용하여 실시간 추적을 통하여 차량을 인식하는 방법이 있다. 이것은 차량의 흐름에 따라서 추적하기 위한 파라미터를 조절해야 하는 단점이 있다.[14] 그리고 컬러 정보를 몇 개의 컬러로 양자화 한 후에 레이블링을 통해 분할한 후 프레임간의 같은 영역을 찾아 차량을 탐지하는 방법[15]이 있다. 이 방법은 도로 영상을 16컬러로 양자화

하는데 도로 영상이 복잡한 경우에 문제점이 있다. 마지막으로 지속 환경 하에서 차량의 후미등을 이용하여 전방 차량을 탐지하여 거리를 측정하는 방법도 있다[16].

본 논문에서는 교통흐름이 복잡하고, 도로 상에 표시들이 다양하게 있는 상황에서도 PC용 CCD 카메라로 연속으로 촬영한 일련의 컬러 영상으로부터 전방의 차량과 차선을 검출하는 방법을 제안한다. 컬러 모델은 HSV 컬러 모델을 기반으로 하고, 차량에는 차량의 밑 부분에 그림자 영역을 가지고 있는 점과 S와 V성분의 상관관계를 이용하여 차량의 후보점을 찾아, 차량 후보 영역을 결정하고, 에지 맵(edge map)을 이용하여 차량의 영역을 결정한다. 또한 차선도 또한 S와 V성분의 관계 영상에서 차선의 후보점을 찾아 차선을 검출한다. 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 차량을 잘 검출함을 보인다.

본 논문의 구성은 2장에서 시스템 구성 및 전처리에 대한 사항을 소개하고, 3장에서는 전방의 차량 검출 방법, 4장에서는 차선 검출 방법을 소개한다. 5장에서는 전방 차량을 인지하는 방법에 대해서 기술하고, 6장에서 연구 결과 및 분석 내용을 제시하고, 7장에서 결론을 맺는다.

2. 시스템 구성 및 전처리

2.1 시스템 구성

컬러 CCD 카메라를 이용하여 전방 차량과 차선을 탐지하는 방법은 그림 1과 같다. 초기 설정부분에서는 입력영상에서 도로의 영역을 지정하는 단계로써 초기에 한번 설정한다. 차선의 정보가 구해지면 설정값이 양쪽 차선의 무한 원점으로 변경된다. 그리고, 카메라로부터 입력되는 RGB영상을 HSV 컬러 공간으로 변환하고 변환된 컬러 공간상에서 색조(Saturation)와 명도(Value)성분을 이용하여 전처리 과정을 거친 뒤에 차량의 후보 영역과 차선의 후보점들을 구한다. 차량과 차선 후보점을 이용하여 차량 영역과 차선을 검출하고, 차선 정보를 이용하여 전방 차량을 결정한다. 이때 전방 차량이 이전 프레임의

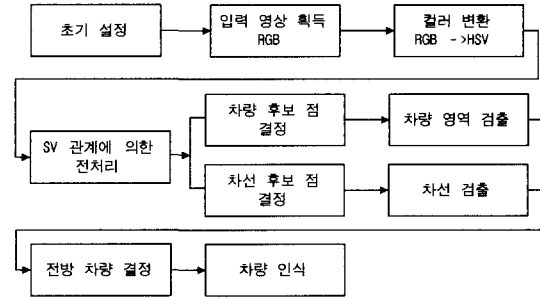


그림 1. 전체 시스템 구성

전방 차량과 같은 차량인지 인식하여 앞 차량이 차선을 변경하는 과정에서 발생하는 위치의 변화를 감지한다.

2.2 컬러 모델 및 그림자 정보

컬러 모델은 일반적으로 RGB, CMYK 등이 널리 알려져 있으나 이 컬러 모델들은 디스플레이 장치와 프린터와 같은 하드 카피 장치에 적합한 모델들이다. 하지만 이 모델들은 명암 및 컬러 요소가 분리되어 있지 않기 때문에 영상처리 알고리즘 구현에는 자주 사용되지 않는다. 그리고 CIE에서 제정한 컬러 모델들이 있다. CIEXYZ, CIEYUV, CIELab, HSI, HSV 등 여러 가지 컬러 모델들이 있다. 이런 컬러 모델중 RGB, L*a*b*, HSI 컬러 모델이 가장 많이 사용되고 있다. 그 중에서 사람의 시각 인식방법과 비슷한 HSI 나 HSV 컬러 모델을 이용하는데 프로그램 구현이 용이한 HSV를 이용한다. HSV는 Hue(색조), Saturation(채도), Value(명도)를 나타내고, HSV 컬러 모델은 그림 2와 같이 육각뿔의 모양을 갖는다.

햇빛이 비치는 아스팔트 도로 부분과 차량의 하단 부에서 빛의 밝기를 비교해보면 빛의 밝기의 차이가 많이 발생함을 알 수 있다. [9]에서 컬러와 휘도를

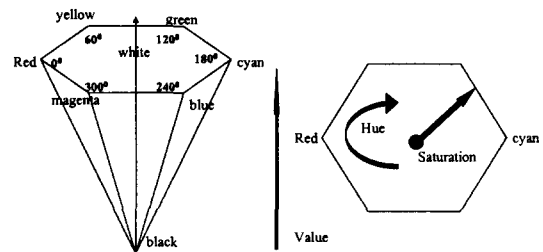


그림 2. HSV 컬러 모델

측정하는 장치로 측정된 결과 햇빛이 비치는 도로의 휘도(luminance)는 3500~4800(cd/m²)이고 차량의 하단부는 100~200(cd/m²)이다. 이렇게 기본적으로 빛의 밝기 차이는 도로 상에서 차량을 검출할 수 있는 특징이 된다.

Dichromatic Reflection Model[20]에 의해 식 (1)과 같이 정의하였을 때 Shading 부분은 $C_L \cong m_a C_a$ 로 근사화가 가능하고, 조명을 직접 받는 부분(Highlight)은 C_a 는 매우 작기 때문에 $C_L \cong m_d C_d + m_s C_s$ 로 근사화가 가능하다. 여기서 백색광에서는 C_s 가 상수로 가정할 수 있어

$$C_L = m_d C_d + m_s C_s + C_a$$

C_d : diffuse reflection, C_s : specular reflection
 C_a : ambient reflection
 m_a, m_b : scale factor

(식 1)

색조 성분은 변하지 않고, 채도 부분은 값이 감소하게 된다. 그림자 부분(Shadow)부분에서는

$C_L \cong m_a C_d + C_a$ 로 근사화가 가능하고, C_a 가 상수로 가정할 수 있어 색조(Hue)부분의 값은 이론적으로 변하지 않는다[20]. 그리고 여러 영상을 이용하여 실험하여 본 결과 도로 위에서의 채도 값은 차량의 하단 부에서 높아짐을 볼 수 있었다. 그림 5에서 S성분을 보면 차량의 하단부의 채도 값을 보면 주위 도로의 채도 값에 비해서 높은 값을 가지고 있음을 알았다. 이렇게 HSV 영상에서 채도 성분이 차량의 하단부에서 높게 나타난다는 점과 명도 성분은 반대로 아주 낮은 값을 갖는다는 점을 이용한다. 그리고 색조 성분은 거의 변하지 않는 성분이기 때문에 사용하기 어렵다.

그림 4와 같이 야간이 아니라면 차량의 하단에 그림자가 존재하게 되고, 그림 3에서와 같이 물체에 의해 빛이 가려지는 부분이 생기게 되는데 암흑부(umbra)는 빛이 차단된 상태로 차량의 하단부에 해당하고, 반암부(penumbra)는 암흑부와 같이 빛이 차단된 상태는 아니고 점차적으로 밝기 값이 변하는 부분이다[17].

본 논문에서는 차량의 하단부인 암흑부를 위주로 차량의 영역을 탐지하고자 한다. 차량의 하단 부가 채도(S) 성분은 주위보다 높아지고, 명도(V) 성분은 매우 낮은 값을 갖는 특징을 이용하는데 명도 성분만을 이용하는 경우에는 차량의 주변에 존재하는 반암

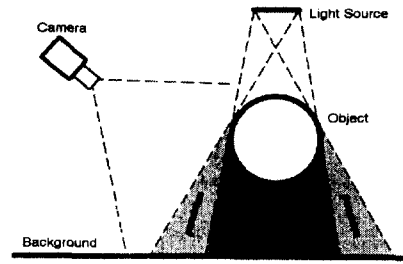


그림 3. 그림자 모델

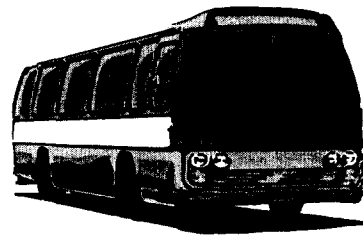


그림 4. 차량의 그림자 형태

부 부분을 제거하기가 어려워 채도 성분을 함께 이용한다. 차량의 하단 부에서는 채도성분의 값이 높아지기 때문에 채도 성분과 명도 성분의 차이를 이용하면 차량의 하단 부의 특징이 더욱 뚜렷해진다. 그리고 이 반대의 특징이 차선의 기본 특징이 된다. 두 성분을 이용한 부분은 전처리 부분에서 설명하기로 한다.

그림 5에서 (a)는 원 영상이며, (b)는 HSV 컬러 공간상에서 S의 성분, (c)는 V성분을 표시한 것이다. (b)에서 보면 차량의 하단에서 비교적 높은 값을 갖는 것을 볼 수 있을 것이다.

2.3 전처리

HSV 컬러 공간으로 변환된 영상에서 채도 성분과 명도 성분의 차이를 구한다. 이때 채도 성분은 0에서 1사이이고, 명도 성분은 0에서 255이기 때문에 채도의 값에 255를 곱하여 차이를 구한다. 그림 6의 (a)가 그 결과이다. 여기서 초기 설정단계에서 미리 지정한 도로 영역에 대해서 평균값을 구한다. 이것은 전방 도로 영상에서 주로 차지하는 부분이 도로이기 때문에 평균값을 이용한다. 그리고 채도와 명도의 차이값을 이용하기 때문에 차선과 그림자 부분의 차이가 명확해 지고, 도로 영역은 평균에 가까워진다. 채도 성분과 명도 성분의 차이 값을 갖는 영상을

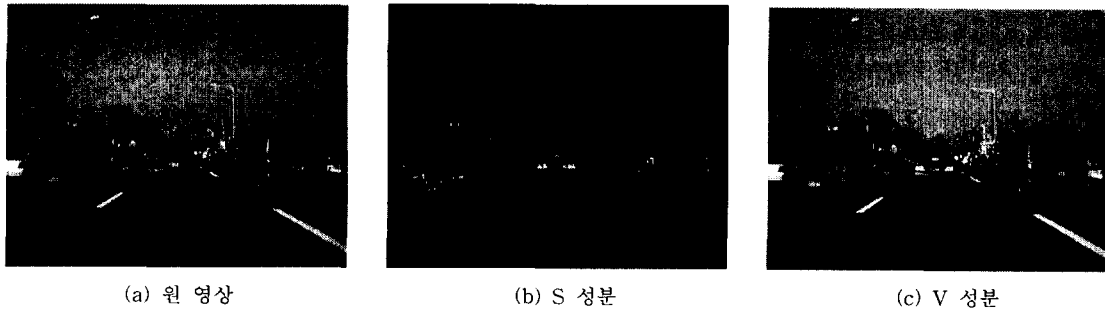


그림 5. S와 V성분의 관계

$F(x,y)$ 로 정하여 식 2와 같이 구한다.

$$F(x,y) = S(x,y) \times 255 - V(x,y) \quad (\text{식 2})$$

여기서 $S(x,y)$ 는 채도 성분, $V(x,y)$ 는 명도 성분을 가리킨다.

이 $F(x,y)$ 값을 이용하여 미리 지정한 영역 평균과 다른 정도를 나타내는 $W(x,y)$ 를 구하고, 이 $W(x,y)$ 가 0보다 작으면, 비 배경 영역이 되고, 0보다 크면 도로 영상과 유사한 배경 영역이 된다.

$$W(x,y) = 1 - K_a \times \frac{(F(x,y) - \mu_r)^2}{\sigma_r^2} \quad (\text{식 3})$$

$$R(x,y) = \begin{cases} 1, & W(x,y) < 0 \text{ and } F(x,y) > \mu_r \\ 2, & W(x,y) < 0 \text{ and } F(x,y) < \mu_r \\ 0, & \text{else} \end{cases}$$

여기서 μ_r 는 지정 영역의 평균을 의미하고, σ_r^2 는 지정 영역의 표준편차를 의미한다. 그리고 K_a 는 가중치로 상수값이다. $R(x,y)$ 가 1이면 차량의 후보 점에 해당하며, 2이면 차선의 후보 점에 해당하고, 0이면 도로 및 배경 영역으로 간주한다. 식 6에 의해서 구한 결과는 그림 6과 같다. 그림 6의 (a)는 $F(x,y)$ 를 표시한 것이고, (b)는 차량에 대한 후보점이고, (c)는

차선의 후보 점에 해당한다.

3. 전방의 차량 검출

전처리 과정에서 구해진 차량 후보 점들을 이용하여 전방의 차량을 검출하는 과정을 소개하면 그림 7과 같다. 5단계의 과정을 거쳐 차량의 영역을 추출한다. 먼저 영역의 주변을 매끄럽게 하고, 작은 홀을 제거하기 위하여 모폴로지 Closing 방법을 적용한다. Structure Element는 3×3 window를 이용하고, Dilation과 Erosion을 통하여 영역의 외곽부분을 부드럽게 함으로써 연결 성분 레이블링을 통해서 영역을 찾는데 여러 개의 작은 영역으로 분할되는 것을 방지한다. 그리고 8방향 연결 성분 레이블링을 통해 영역을 분리하였다. 연결 성분 레이블링 방법은 재귀 순환 알고리즘을 이용하였다[18].

$$Dilation : B \oplus S = \bigcup_{b \in B} S_b$$

$$Erosion : B \ominus S = \{ b \mid b + s \in B \ \forall s \in S \} \quad (\text{식 4})$$

$$Closing : B \cdot S = (B \oplus S) \ominus S$$



그림 6. 차량과 차선의 후보점

B는 이진화된 원 영상을 의미하고, S는 Structure Element이다.

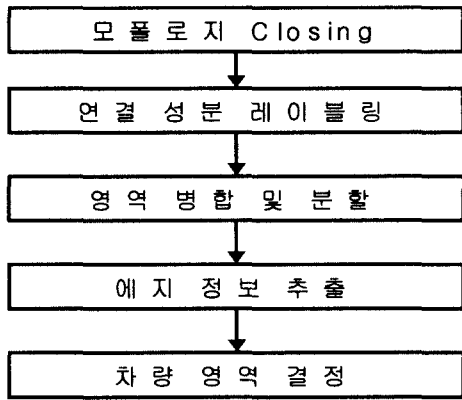


그림 7. 차량 영역 검출 흐름도

이렇게 분리된 영역은 영역 병합 및 분할 과정을 거쳐 차량의 후보 영역을 찾고, 차량의 후보 영역에서 에지 정보를 추출하여 차량의 영역을 결정하였다.

3.1 영역 병합 및 분할

연결 성분 레이블링을 통해서 구해진 차량의 후보 영역들은 작은 영역들로 나누어지거나 하나의 큰 영역으로 분리되는 경우들이 있다. 따라서 영역들에 대한 크기와 위치 정보를 이용하여 영역을 병합하고 분할한 뒤에 작은 영역을 제거한다. 영역 병합은 하나의 영역이 2개 이상의 영역으로 나뉘어진 경우로써 다음의 조건을 만족하면 영역을 병합한다. 조건 3까지 만족하면 영역 P_i 로 영역 P_j 를 병합한다.

- 조건 1. 모든 영역 P_i 에 대해서 수직 위치가 유사한 영역 P_j 탐색
 $|E_y(P_i) - E_y(P_j)| < T_1$
- 조건 2. 조건 1을 만족하는 영역 P_i 에 대한 P_j 영역 중 수평 위치가 유사한 P_j 영역 $E_x(P_i) - S_x(P_j) < T_1$
- 조건 3. 조건 2를 만족하는 영역 P_j 의 크기가 임계값 이하인 영역
 $Size(P_j) < T_2$

영역의 분할은 영역의 넓이가 임계치 이상이면 분할하는데 여기서는 영상 폭의 1/2 이상되는 영역에

대해서 분할한다. 영역분할 방법은 영역의 하단에서 위로 투영하여 영역에서 레이블링된 화소와 만나는 점까지의 거리를 구한다. 그리고 이웃 투영값과의 차이가 최대인 점을 분할점으로 정한다. H를 투영값이라고 하면 투영값이 차이가 가장 큰 위치를 분할점으로 잡는다. 이것은 두 차량이 하나의 영역으로 결정되는 과정에서 거리의 차이가 있기 때문에 투영값으로 분할이 가능하다. H_i 를 i 의 위치에서 투영값이라고 하면 분할점은 다음과 같이 구해진다.

$$\text{분할점} = \max_i (|H_i - H_{i+1}|) \quad (\text{식 5})$$

마지막으로 크기가 작은 영역들은 제거하도록 한다. 작은 영역의 크기는 폭이 20화소 이내인 영역을 제거한다.

3.2 에지 검출

에지를 추출하는 방식은 그레이 영상에서 많이 이루어지고 있으며, 다양한 방법들이 많이 소개되어 있다[18]. 컬러 영상에서의 에지를 추출하는 방법들도 여러 가지 방법들이 있으나 이런 방법들은 에지 맵을 이용한 영상 분할이나 영상 분석을 하기 위한 방법들이며, 본 연구에서는 Prewitt 연산자를 이용한다. Prewitt 연산자는 3*3 행과 열 방향의 방향성을 이용한 에지 필터를 이용한다. 식 6과 같이 수평과 수직 방향의 필터를 이용한다[19].

$$\Delta H = \frac{1}{3} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \Delta V = \frac{1}{3} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad (\text{식 6})$$

컬러 영상에서의 에지 추출은 그레이 영상에서 처리하는 방법과 유사하게 처리하는데 Prewitt 연산자를 이용하여 수평과 수직 방향에 대한 영상의 기울기를 구한다. 식 6과 유사하게 수평과 수직의 방향성 필터를 구성하고, 각각이 음의 성분과 양의 성분을 나타낸다. 이 배열의 크기는 가변적으로 다양하게 구현이 가능하다. 식 7에서 보면 파라미터 w 는 양의 정수를 나타내고, V, H 는 수직, 수평 방향의 방향을 나타낸다. 이들의 결과는 지역 컬러의 평균을 의미한다.

화소 (x_0, y_0) 에서 컬러 기울기를 구하기 위해서 식 7과 같이 계산한다. 여기서 $\vec{c}(x, y)$ 는 S와 V성분의 벡터이다.

$$\begin{aligned} \vec{H}_+(x_0, y_0) &= \frac{1}{w(2w+1)} \sum_{y=y_0-w}^{y_0+w} \sum_{x=x_0+1}^{x_0+w} c(x, y) \\ \vec{H}_-(x_0, y_0) &= \frac{1}{w(2w+1)} \sum_{y=y_0-w}^{y_0+w} \sum_{x=x_0-1}^{x_0-w} c(x, y) \\ \vec{V}_+(x_0, y_0) &= \frac{1}{w(2w+1)} \sum_{y=y_0+1}^{y_0+w} \sum_{x=x_0-w}^{x_0+w} c(x, y) \\ \vec{V}_-(x_0, y_0) &= \frac{1}{w(2w+1)} \sum_{y=y_0-1}^{y_0-w} \sum_{x=x_0-w}^{x_0+w} c(x, y) \end{aligned} \quad (\text{식 7})$$

지역적 통계정보는 (V+, V-, H+, H-) 성분에 영향을 미치고, 수직, 수평 방향의 지역적 변화량은 벡터의 차이를 이용하여 계산한다.

$$\begin{aligned} \Delta\vec{H}(x_0, y_0) &= \vec{H}_+(x_0, y_0) - \vec{H}_-(x_0, y_0) \\ \Delta\vec{V}(x_0, y_0) &= \vec{V}_+(x_0, y_0) - \vec{V}_-(x_0, y_0) \end{aligned} \quad (\text{식 8})$$

스칼라 값인 ||H||와 ||V||는 수직방향과 수평 방향으로 점(x0, y0)에서의 변화율을 나타내고, B(x0, y0)는 점 (x0, y0)에서 최대 변화량을 나타낸다.

$$B(x_0, y_0) = \sqrt{\|\Delta\vec{V}(x_0, y_0)\|^2 + \|\Delta\vec{H}(x_0, y_0)\|^2} \quad (\text{식 9})$$

그리고, 점 (x0, y0)에서의 최대 변화율을 가진 방향을 Θ 이다. 여기서 k는 정수로 정규화하기 위한 상수이다.

$$\Theta = \arctan\left(\frac{\Delta V'(x_0, y_0)}{\Delta H'(x_0, y_0)}\right) + k\pi \quad (\text{식 10})$$

그리고 스칼라 값인 ||H|| 값은 유클리디안 거리를 이용하며, 흑백 영상에서 에지를 추출한 것과 일관성을 유지하기 위해서 $\sqrt{s^2 + v^2}$ 와 같이 구한다.

$$\begin{aligned} \Delta V'(x_0, y_0) &= \|\vec{V}_+(x_0, y_0)\| \geq \|\vec{V}_-(x_0, y_0)\| ? \|\Delta\vec{V}(x_0, y_0)\| : -\|\Delta\vec{V}(x_0, y_0)\| \\ \Delta H'(x_0, y_0) &= \|\vec{H}_+(x_0, y_0)\| \geq \|\vec{H}_-(x_0, y_0)\| ? \|\Delta\vec{H}(x_0, y_0)\| : -\|\Delta\vec{H}(x_0, y_0)\| \end{aligned} \quad (\text{식 11})$$

에지를 찾은 결과가 그림 8에 나타나 있다. 그림 8에서 (a)는 B(x, y)가 임계치 이상 되는 점을 표현한 것이고, (b)는 수평과 수직 성분의 에지 정보만을 나타낸 것이다. Θ 가 -10~10도, 170~190도 사이인 점을 수직방향으로 지정하고, 80~100도, 260~280도 사이에 있는 점을 수평방향의 화소로 간주한다.



(a) 에지 추출 결과



(b) 수평 수직 성분의 에지

그림 8. 에지 추출 결과

3.3 차량 영역 결정

먼저, 에지 정보를 이용하여 에지 정보의 수직, 수평 성분이 영역 내에 존재하는지 비교하여 에지 정보가 없으면, 차량 영역에서 제외한다. 주변 풍경에 의해서 발생하는 그림자인 경우에는 에지가 거의 발생하지 않기 때문에 에지 정보를 이용하여 차량 영역에서 제외시킨다.

이렇게 구해진 영역에서도 현재 차량 외에 주변 영역이 포함되는 경우들이 있다. 가까운 차량과 멀리 있는 차량이 하나의 영역으로 검출되는 경우가 있다. 거리 상으로는 멀리 있지만 원근 투영에 의해서 하나의 영역으로 검출되는 경우들이 있기 때문에 영역 분할 시에 사용했던 방법과 마찬가지로 수직 방향으로 투영하여 투영값의 변화가 큰 지점을 좌/우 측에서 찾아 주변 영역을 제거하여 차량 영역을 결정한다. 마지막으로 영역의 하단부 폭을 이용하여 기본적인 차량의 높이를 결정하고, 에지 정보에서 수직/수평 성분이 존재하는 위치를 찾아서 차량의 높이로

결정한다. 여기서 종횡비는 1.2로 지정했다. 그림 9에서 왼쪽이 주변 영역을 제거하기 전 상태이고, 오른쪽이 주변 영역이 제거된 상태이다. 분할점은 투영값을 이용하여 결정하게 된다.

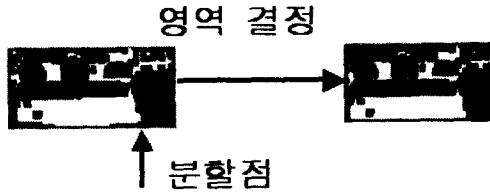


그림 9. 차량 영역의 결정

4. 차선 검출

차선을 검출하는 최종 목표는 전방의 차량을 결정하기 위함이다. 차선을 검출하기 위해서는 먼저 차선 후보점들을 모폴로지 Closing을 이용하여 내부에 생긴 홀들을 제거한다. 그리고 도로상에 도로 표시가 있는지 확인하여 도로 표시가 있으면 차선을 추출하지 않고, 예측된 차선으로 처리한다. 이 부분에서는 에러의 요소가 많기 때문에 회피하는 방법을 이용한다. 차선을 찾는 단계는 라인 패치점들을 구해서 후보 차선을 결정하고, 후보 차선에 대해서 차선 영역 탐색을 통해 기울기와 실제 차선임을 결정한다. 무한 원점을 구하고, 차선을 추적한다.

차선을 추출하기 위해서는 다음의 가정을 한다. 차선은 그림 11에서 보는 바와 같이 차선은 세 가지 형태로 나타난다. 그리고 차선이 가려져 있지 않다면

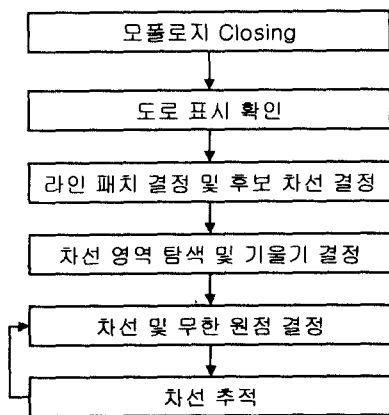


그림 10. 차선 검출 흐름도

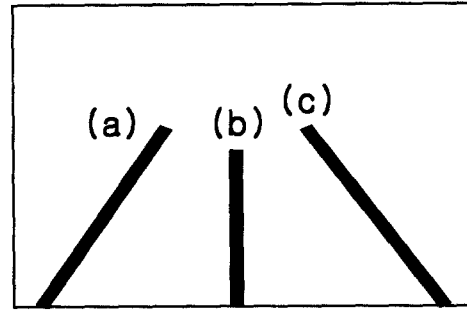


그림 11. 차선의 형태

도로 상에서 처음 발생하는 특징이다. 이런 가정을 통해서 차선의 패치 포인트를 찾는다. 여러 개의 차선의 보이더라도 제일 가까운 차선만이 관심 대상이다.

도로에 표시가 있는 부분인지 알기 위해서는 영상의 하단부분에서 차선 후보 점을 이용하여 투영값을 구한다. 이때 투영값의 히스토그램을 보면 도로 표시가 있는 경우에는 여러 위치에서 값이 나타나고, 차선만 있는 경우에는 0에서 2개의 위치에서만 나타난다. 차선이 이렇게 많이 보이는 경우가 없기 때문에 이런 경우에는 도로의 표시 부분으로 간주하여 차선을 추출하는 과정을 생략한다.



그림 12. 도로에 표시가 있는 경우

4.1 라인 패치 및 후보 차선 결정

차선이 도로 영상에서 가장 앞서 있는 정보이기 때문에 영상의 하단에서부터 시작점을 찾는다. 이때 시작점은 영상의 하단에서 위로 투영하여 처음 만나는 점을 이용한다. 시작점을 찾으면 이웃점과 비교하여 거리가 가까우면 계속 이웃점을 찾고, 이웃점의

거리가 일정거리 이상 되는 경우에 끝점으로 정한다. 그리고 다시 시작점과 끝점을 찾는다. 이때 시작점과 끝점을 하나의 패치로 결정하고, 다음의 조건에 만족하면 차선의 후보에 해당한다.

이렇게 구해진 패치는 그림 12와 같은 형태인지 체크하여 아니면 제거한다. 그리고 기울기를 구하여 (a), (b), (c)의 조건에 만족하는 후보만을 차선의 후보로 간주하고, 차선의 시작점이 도로 영역 밖에 있는 경우와 길이가 짧은 경우에는 제거한다.

조건 1. 차선의 시작점은 도로 영역 안에 있어야 한다.

조건 2. 패치의 길이는 일정길이 이상 되어야 한다.

조건 3. 영상을 3등분하여 왼쪽 부분의 차선은 기울기가 (+) 방향이고, 가운데 부분은 수평으로 0이나 180도와 유사하며, 오른쪽 부분은 기울기가 (-) 방향이어야 한다. 패치가 $P_i(i=1..n)$ 라고 하면, E_x 는 끝점의 X좌표, S_y 는 시작점의 Y좌표를 의미한다. 이때 θ 의 값이 기울기가 된다.

$$\theta = \tan^{-1} \left(\frac{|E_y(P_i) - S_y(P_i)|}{|E_x(P_i) - S_x(P_i)|} \right) \quad (\text{식 12})$$

위 조건을 만족하면 차선의 후보 영역으로 간주한다.

4.2 차선 영역 탐색 및 기울기 결정

앞에서 구한 차선의 후보 영역에서 40*3 블록을 이용하여 차선의 후보 영역을 추적한다. 차선의 후보 영역을 추적하는데 있어서 다음의 조건을 만족하지 않으면 추적을 중단한다.

- 다음 블록에서 차선의 중심점에 해당하는 부분이 있어야 한다.
- 차선의 넓이가 전 블록에서의 넓이와 유사해야 한다.

위의 조건이 맞아 블록의 개수가 4이상 되면 차선으로 간주한다. 여기서 블록의 개수가 3개 이하이면 차선의 기울기를 구하는데 제외시킨다. 이유는 실제 차선인 경우라도 차선의 기울기를 구하는데 어려움이 많으며, 너무 짧은 경우에는 차선이 아닌 확률이 높기 때문에 차선 영역에서 제거한다. 4개의 블록이상 연결되어 있으면 최소 자승법(Least Square Me-

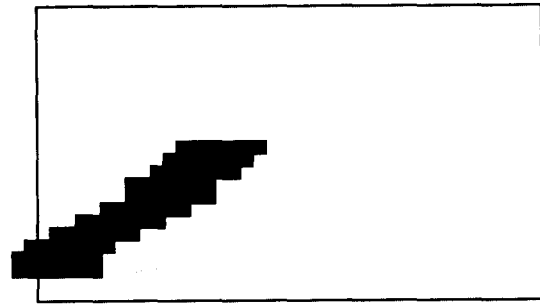


그림 13. 블록 윈도우를 이용한 차선 검출



그림 14. 차선의 후보와 차선 탐색 결과

thod)을 이용하여 차선의 기울기를 구한다.

그림 14에서 보면 차선이라고 판단되는 차선의 후보와 차선의 탐색결과를 표시하였다. 조그만 사각형이 탐색하는 과정에서 중심점을 표시한 것이다. 여러 곳에 차선의 후보 영역이 나타나지만 기울기를 구하는 과정에서 보다 확실한 차선의 정보를 얻을 수 있다.

4.3 차선의 예측

차선은 직선과 점선으로 이루어져 있다. 특히 점선인 경우에는 매 프레임마다 차선이 나타난다는 것을 보장할 수 없다. 그리고 점선에서는 양쪽 차선 중에서 한쪽 차선만 보이는 경우가 많다. 이런 경우에 보통 차선의 기울기를 추적하는데 카메라의 흔들림이나 차량의 이동에 따라서 차선의 위치와 기울기가 다양하고 빠르게 변하기 때문에 본 연구에서는 차선을 예측하는데 최근의 정보를 잘 활용하는 방식으로 최근 n개의 프레임에 대한 가중치 평균을 이용한다. 가중치 평균을 이용하는데 있어서도 최근의 프

레이에 대한 가중치를 높게 줌으로써 좀 더 최근에 구해진 차선의 정보를 따라가게 된다.

차선의 정보는 $y=ax + b$ 형태로 나오는데 이전 n 개의 프레임에서의 기울기 a 와 절편 b 를 가중치 평균한다. 여기서 A 와 B 는 예측된 차선의 정보를 나타내고, a 와 b 는 각 프레임에서의 차선 정보이다.

$$Y = Ax + B \tag{식 13}$$

$$A = \frac{\sum_{i=c, j=1}^{i=c+A, j=n} w_j a_i}{n}, \quad B = \frac{\sum_{i=c, j=1}^{i=c+B, j=n} w_j b_i}{n}$$

5. 전방 차량 결정

5.1 전방 차량의 결정

차량 영역 검출과정에서 하나의 영역이 검출되는 것이 아니라 여러 개의 영역이 검출된다. 여러 개의 영역 중에서 전방 차량을 결정함으로써 차량 전방의 상황을 알 수 있다. 전방 차량을 결정하는 방법은 현재 왼쪽 차선과 오른쪽 차선이 구해졌다면 양쪽 차선을 연장하여 무한 원점을 구해 그 영역에 들어오는 차량을 전방 차량으로 간주한다. 이때 1개의 차량이 있으면 그 차량이 전방 차량이 되고, 2개 이상이면 Y축 좌표가 최대인 차량이 전방 차량이 된다. 차선 안에 존재하는지 판별하는 방법은 차량 영역의 외곽 영역을 이용하지 않고, 레이블링된 화소들이 어느 정도 포함되는지 비교한다. 차선이 없는 경우에는 영상의 중심을 기준으로 중심에 가장 근접한 차량을 찾아 전방 차량으로 결정한다.

5.2 같은 차량 인식

전방 차량을 검출한 뒤에 이전 프레임의 전방 차량과 현재 프레임의 전방 차량이 같은 차인지 구분하는 것이 중요한데, 이것은 자기 차량이 차선을 변경하거나 다른 차량이 차선을 변경하는 경우에 차선이탈과 위치적인 특징에 의해서 전방 차량이 바뀌는 경우가 있다. 이때 전방 차량의 위치 정보가 급격하게 변하는데 이 경우에는 차량의 인식을 통하여 전방의 상황이 정상적임을 판단해야 한다. 이전 프레임에서의 전방 차량과 현재 프레임에서의 전방 차량이 같은 차량인지 확인하는 방법으로는 통계적 특징과 유클리디안 거리를 이용하여 최소 거리에 있는 영역

을 같은 차량으로 간주한다.

사용하는 특징으로는 컬러 공간의 HSV의 성분의 평균값과 차량 영역의 중심점을 이용한다.

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

$$Dc_{ab} = \sqrt{(Dc_a - Dc_b)^2} \tag{식 14}$$

$$DT_{ab} = w_1 * Dr + w_2 * Dc$$

DT_{ab} 가 최소가 되는 영역을 이전 프레임에서의 전방 차량 영역과 동일한 차량 영역으로 정한다. Dr_{ab} 는 두 영역의 각 성분별 평균의 거리를 의미하고, Dc_{ab} 는 영역의 중심점의 거리를 의미한다. 여기서 w_1 과 w_2 는 가중치이다.

6. 실험 결과 및 분석

6.1 실험 환경

실험은 올림피도로상에서 USB 방식의 PC 카메라를 이용하여 초당 10프레임으로 영상을 캡처 하였으며, 영상의 크기는 320*240을 사용하였다. 실험 시에 카메라의 위치는 차량의 운전석 앞부분에 설치하였으며, 카메라의 위치 보정을 하거나 수평을 유지토록 하는 보조 장치는 사용하지 않았다. 영상은 약 5분씩 5번 캡처 하였으며, 영상을 캡처한 당시의 환경은 주변 차량이 많은 경우, 주변 차량이 없는 경우도 있었으며, 전방에 차량이 없는 경우도 있었다. 실험에서 사용된 시스템은 인텔 펜티엄III 650MHz 에 256MB의 노트북이며, 윈도우 2000에서 실험하였다.

6.2 실험 결과 및 분석

전방차량 검출에 대한 결과로는 표1과 같다. 전방 차량에 대한 검출의 정확도를 구해보면 약 93%의 검출율을 보이고 있다. 실험에서 정확도라는 것은 다른 시스템과 환경이 달라 비교가 불가능하고, 수치상으로 표현할 수 없기 때문에, 전방 차량을 검출한 외곽 사각형이 제대로 차량의 위치에 위치하는지 확인하는 방법을 이용하였다. 이 실험 결과에서 에러가 발생하는 부분은 차량이 고가도로 아래를 통과하는 과정에서 보면, 고가도로에 의해서 빛을 받지 못하는 부분이 발생하는데 이 지점에서는 거의 터널을 통과하는 것과 비슷한 효과가 발생한다. 그리고 에러의 조건은 차량의 영역이 너무 크거나 너무 작은 경우에

표 1. 전방 차량 탐지 결과

영상 번호	프레임 수	에러 (frame)	에러율 (%)	정확도 (%)
영상 #1	2481	121	4.88	93.12
영상 #2	2868	115	4.00	94.32
영상 #3	2912	190	6.52	92.0
영상 #4	2969	172	5.79	93.67
영상 #5	2776	336	12.10	87.9
전체	14006	934	6.67	93.33

에러라고 판정하였으며, 전방 차량의 위치가 완전히 다른 경우와 전방 차량이 있는데 찾지 못하는 경우를 에러라고 판정하였다.

그림 15에서 (a)는 원 영상이고, (b)는 전처리 과정에서 생성된 $F(x,y)$ 영상이다. (c)는 차량의 후보점을 표시한 것이고, (d)는 차선의 후보점을 표시한 것이다. (e)는 라인 패치와 차선의 기울기를 구하는 과정을 보인 것이며, (f)는 차량과 차선을 검출한 최종 결과이다.

그림 16의 (a)는 전방 차량의 Y축 좌표값의 변화를 보여주고 있다. 500 프레임의 데이터로써 전방 차량의 위치를 정확히 따라가고 있음을 보인다. 그리고 (b)는 전방 차량 폭(width)의 변화로 거의 유사하게 차량의 폭을 검출하고 있다. 여기서 351 프레임에서

보면 전방 차량의 위치와 폭의 변화가 심한데 이것은 옆 차선에서 차량이 차선을 변경하는 과정에서 발생하는 데 차량의 폭이 넓어지는 것은 두 대가 하나의 차량으로 잘못 검출되는 경우이다. 일정 프레임이 지난 뒤에는 폭의 변화가 거의 없어진다.

에러를 분석해 보면, 차량의 존재 유무를 판별하지 못하는 경우가 300프레임 정도로 전체의 영상에서 2.1%이고 대부분의 에러는 차량의 영역을 결정하는데 넓은 영역이나 좁은 영역으로 검출되는 것이다. 그림 17의 (a)에서 보면 두 대의 차량이 하나의 차량 영역으로 지정된 것이고, (b)에서 보면 차량의 영역이 너무 작게 지정된 경우이다. (a)에서는 왼쪽 차량의 그림자가 오른쪽 차량과 같이 연결되어 있어서 영역 분리 시에 하나의 영역으로 분리되었으며, 분할 과정에서 분할되지 못한 경우이다. 그리고 (b)는 영역 분할과정에서 잘못 분할된 예이다. 차량의 크기가 다양하고 그림자의 위치가 다양하게 발생하기 때문에 이런 에러가 발생할 가능성이 있다.

차선은 흰색 차선이 있는 경우에는 거의 정확한 검출이 가능했으며, 노란색과 파란색 차선은 검출하지 못했다. 그리고 검출된 차선의 길이가 짧은 경우에 기울기를 구하는 과정에서 약간의 오류가 있었으며, 차선에 그림자가 있는 경우에도 찾지 못하는 경

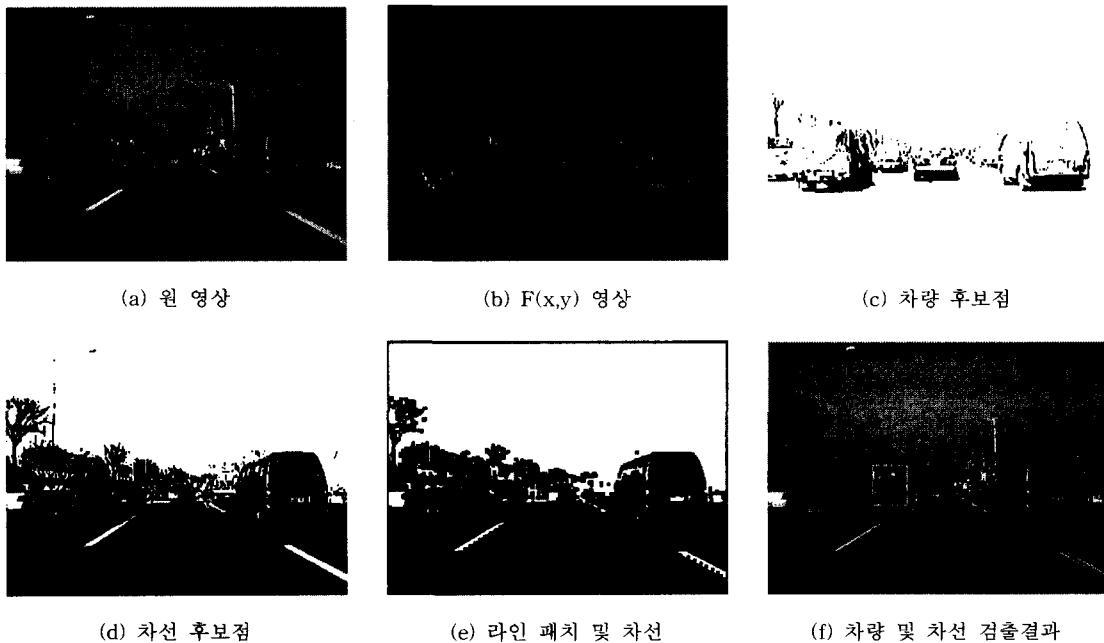


그림 15. 차량 검출과 차선의 기울기를 표시한 결과

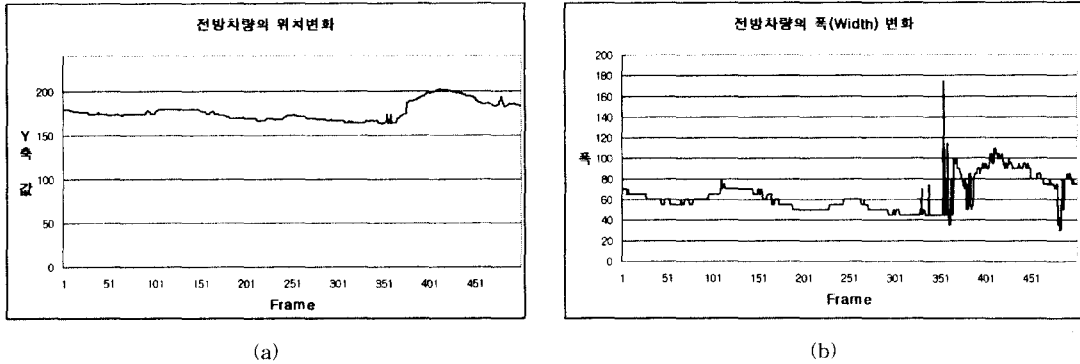


그림 16. 전방차량의 위치 변화 및 폭 변화

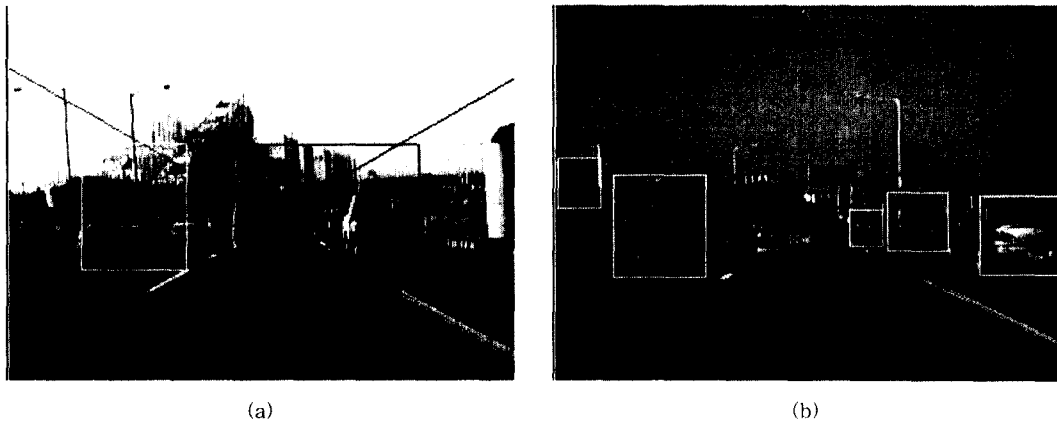


그림 17. 어러 영상

우가 있었다. 기울기 오류는 최소 자승법에 의해 기울기를 구하는 과정에서 오류가 발생했다. 차선을 예측하는데 있어서 차선 변경을 하는 동안에는 예측하기가 어려운데 차선을 변경하는 과정에서는 차선의 기울기가 급격하게 변하기 때문에 차선이 검출되지 않는 부분에서의 예측값에 에러가 많았다. 차선이 제대로 검출되고 있는지 알아보는 방법으로 직선도로인 경우 양쪽 차선을 연결한 무한 원점이 한곳에 모여 있다면 차선을 정확히 포착한 것이 될 것이다. 그림 18과 같이 약 260개의 프레임에서 무한 원점을 표시해보면 하나의 점에 모여 있는 것을 볼 수 있다. 그리고 그림 19는 양쪽 차선의 기울기 변화를 보여주고 있다. 양쪽 차선의 기울기 변화가 발생하는 것은 차량이 흔들리면서 카메라가 진동하기 때문이다.

처리 속도는 100프레임에 대해 평균해본 결과 약 320ms로써 실제 차량에 장착하여 수행시키는데는 디스플레이 속도까지 감안하면 초당 3프레임정도를

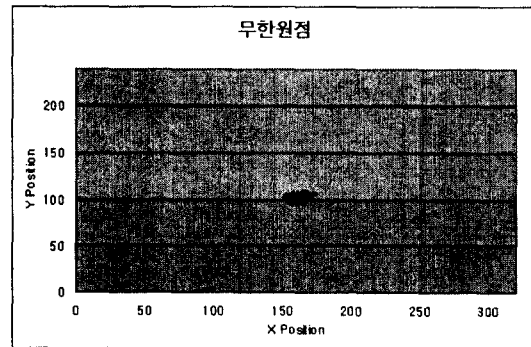


그림 18. 무한 원점의 분포

처리할 수 있다. 그림 20에서의 (a)는 도로 위에 표시가 있는 경우에 전방 차량을 검출한 결과로 도로에 차선 이외의 도로 표시가 있는 경우에도 차량 검출이 용이하다. 그리고 (b)는 요철이 있는 경우에 실험한 것이다. 하지만 모든 요철에 대해서 가능하지는 않

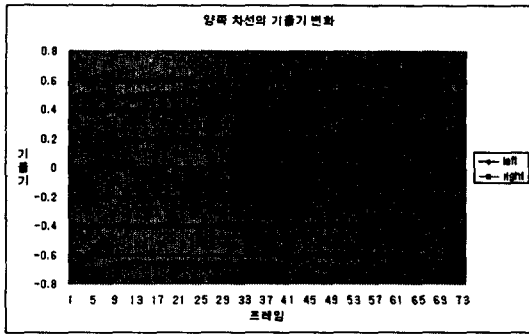


그림 19. 차선의 기울기 변화

다. 조금 높은 요철인 경우에는 에러의 소지가 있다. 마지막으로 그림 21은 프레임 단위로 연속된 프레임의 결과를 보인다.

7. 결 론

컬러 영상에서 영상 처리를 이용하여 전방의 차량을 검출하고, 차선의 정보를 추출하는 방법을 알아보았다. 카메라를 통해 보이는 영상에서 대부분의 영역이 도로 영역이며 도로상에서 차량이 존재하는 경우 나타나는 특징은 차량의 하단부에 어두운 영역이 있다는 것이다.

전방의 차량을 찾기 위해서 RGB 컬러모델을 HSV모델로 변환한 후에 채도 성분과 명도 성분을 이용하여 차량의 후보와 차선의 후보를 분리하였으며, 도로 영역에 해당하는 전체의 평균 정보를 이용하여 평균에서 일정 거리 이상 떨어진 점을 차량과 차선의 후보 영역으로 정하였다.

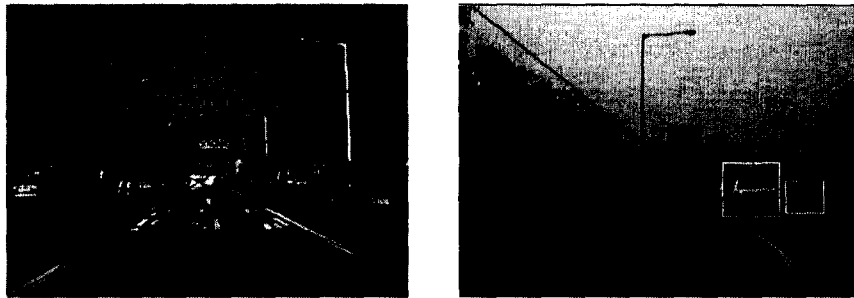


그림 20. 도로에 표시가 있는 경우의 결과

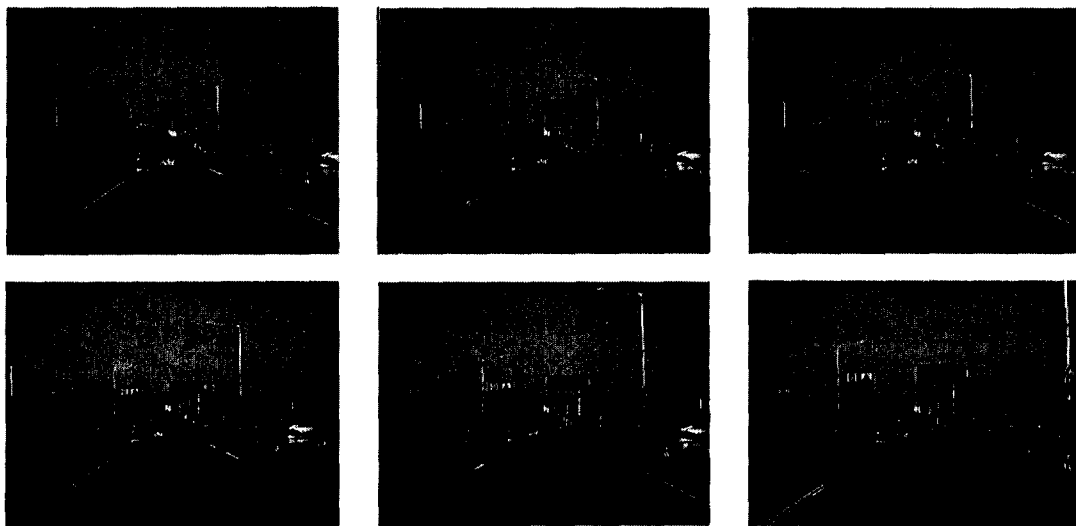


그림 21. 프레임 단위의 실험 결과

카메라가 계속 이동하기 때문에 주변의 환경이 계속 변하게 된다. 이런 상황에서 컬러 영상을 이용하는 다른 방법에 비해 주변 환경에 덜 민감하게 동작한다는 장점이 있는데 그 중 카메라로 들어오는 빛의 양에 따라서 영상의 컬러 값의 변화가 심해지거나 USB 방식의 PC카메라를 이용하기 때문에 빠른 속도로 영상을 입력받을 수 없고, 카메라의 속도에 따라서 영상의 선명도가 수시로 변하는 경우에 좋은 효과를 발휘하게 된다. 이러한 상황들은 Color blob을 이용하는 경우나 광류(Optical flow)를 이용하는 경우 많은 문제점을 노출시키지만, 본 연구에서는 이러한 문제점들을 해소하고 있으며, 도로의 바닥에 방향 표시자나 낮은 요철이 있는 경우에도 좋은 효과를 보이고 있다. 영상처리만을 이용하여 전방의 차량을 검출하는데 93%이상의 정확도를 보이고 있다.

카메라의 수평을 유지하도록 하는 장치와 같이 별도의 보조 장비를 사용하지 않고, 하나의 카메라를 이용하기 때문에 카메라 캘리브레이션(calibration)을 한다고 하더라도 차량의 거리를 정확하게 구할 수 없지만 차량이 멀어지고 가까워지는 정도는 충분히 파악할 수 있었다. 그리고 처리 속도면에서도 별도의 DSP 장비가 없어도 초당 약 3 프레임을 처리할 수 있으며, 여러 프레임의 정보를 이용하는 것이 아니라 단일 프레임의 정보만을 이용하여 차량의 위치를 검출하기 때문에 영상을 입력받는 속도에 무관하게 처리할 수 있다. 올림픽도로나 고속도로 등 자동차 전용도로에서 영상처리를 기반으로 저가의 하드웨어를 이용하여 전방의 차량을 검출 시스템을 구축할 수 있는 기반이 될 수 있으리라 본다.

향후 연구과제로는 현재는 검출된 여러 영역 중에 전방의 한 차량에 대해서만 처리를 하지만 여러 차량에 대한 정보를 검출하고, 추적하는 방법이 진행되어야 할 부분이다. 그리고 악천후에서 차량의 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.

참 고 문 헌

- [1] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner, W.v. Seelen, "An Image processing system for driver assistance", *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] 한 상훈, 조 형제, "전방의 차량포착을 위한 연속영상의 대상영역을 제한한 효율적인 차선 검출", *정보처리학회 논문지*, 제8-B권, 제 6 호, pp 705-716, 2001.
- [5] Nobuhiro TSUNASHIMA, Masato NAKAJIMA, "Extraction of the Front Vehicle using Projected Disparity Map", *Conference Visual Communications and Image Processing '99, califonia*, January, pp 1297-1304, 1999.
- [6] 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.
- [7] 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.
- [8] 정전익, 최성구, 노도환, "무한원점을 이용한 주행방향 추정과 장애물 검출", *전자공학회논문지*, 제34권, 제11호, pp 1302-1313, 1997.
- [9] H. Mori, N.M. Charkari, T. Matsushita, "On-line vehicle and pedestrian detections based on sign pattern", *Industrial Electronics, IEEE Transactions on*, vol. 41, pp 384-391, 1994.
- [10] Y. Du, N.P. Papanikolopoulos, "Real-time vehicle following through a novel symmetry-based approach", *Robotics and Automation, Proceedings.*, 1997 *IEEE International Con-*

[1] U. Handmann, T. Kalinke, C. Tzomakas, M. Werner, W.v. Seelen, "An Image processing

ference on , Vol. 4, pp 3160 -3165, 1997.

[11] 이용주, "그룹화 블록 스네이크 알고리즘을 이용한 차선추출", 멀티미디어학회 논문지, 제3권 제5호, pp 445-453, 2000.

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

[13] 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.

[14] 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.

[15] B. Heisele, W. Ritter, "Obstacle Detection Based On Color Blob Flow", Proceedings of the Intelligent Vehicles '95 Symposium., pp 282 -286 ,1995.

[16] 노광현, 한민홍, "저속 주행 환경에서 컬러비전 기반의 근거리 전방차량추적", 한국 정보처리학회 논문지, 제7권 제9호, pp 3037-3047, 2000.

[17] 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.

[18] Linda G. Shapiro, Georage C. Stockman, COMPUTER VISION, Prentice Hall, 2001.

[19] 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.

[20] Tsang, W.H., Tsang, P.W.M., "Edge gradient method on object color", IEEE TENCON. Digital Signal Processing Applications Vol. 1, pp 304-310, 1996.



한 상 훈

1990년 동국대학교, 전자계산학과 졸업(학사)
 1990년~1993년 공군 학사 장교 근무
 1995년 동국대학교 대학원 컴퓨터 공학과(공학석사)
 1998년 동국대학교 대학원 컴퓨터공학과(박사수료)

1998년~현재 : 동국대학교 시간강사
 관심분야 : 멀티미디어 정보처리, 컴퓨터비전, 형태인식, ITS, 게임
 E-mail: hansh@dongguk.edu



조 형 제

1973년 부산대학교, 전자 공학과 (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, 컴퓨터비전, 컴퓨터 그래픽스, 형태인식, 게임
 E-mail: chohj@dongguk.edu