

블럽칼라링 기반의 횡단보도와 정지선 검출

Stop-Line and Crosswalk Detection Based on Blob-Coloring

이 준 웅*
(Joon-Woong Lee¹)

¹Chonnam National University

Abstract: This paper proposes an algorithm to detect the stop line and crosswalk on the road surface using edge information and blob coloring. The detection has been considered as an important area of autonomous vehicle technologies. The proposed algorithm is composed of three phases: 1) hypothesis generation of stop lines, 2) hypothesis generation of crosswalks, and 3) hypothesis verification of stop lines. The last two phases are not performed if the first phase does not provide a hypothesis of a stop line. The last one is carried out by the combination of both hypotheses of stop lines and crosswalks, and determines the stop lines among stop line hypotheses. The proposed algorithm is proven to be effective through experiments with various images captured on the roads.

Keywords: autonomous vehicle, stop line detection, crosswalk detection, edge analysis, blob coloring

I. 서론

최근, 스스로 주위환경을 인지하여 이를 기반으로 도로를 자율주행하는 차량에 대한 연구가 관심을 끌고 있다. 차량의 자율주행을 실현하기 위해서는 여러 일들이 수행되어야 한다. 예를 들어 차량의 위치, 차량의 주행방향 및 속도, 차량의 주행가능 공간 등의 인지는 포장도로이건 비포장도로이건 상관없이 필수적 요소다. 반면에 자율주행이 포장도로에서 이루어질 경우 노면객체의 인식이 필요하다. 대표적인 노면객체로는 차량의 경계(lane mark)와 횡단보도(crosswalk) 그리고 정지선(stop line) 등을 들 수 있다. 차로 경계의 인식은 차량이 자기 차로 유지를 위해 필요한 기능이다[1,2]. 또한 도심지 도로의 횡단보도에서 정지신호등이 켜지거나 보행자가 있을 경우 차량은 정지선 앞에서 정지를 해야 한다. 이때 횡단보도와 정지선 인식이 필요하다.

노면객체는 차량과 같은 여타의 장애물들과는 달리 검출 과정에서 컴퓨터비전에 대한 의존성이 크다. 그 이유는 노면객체는 노면에 매우 얇게 색칠되어 있으면서 많은 잡음에 노출되어 있어서 카메라로 촬영된 영상정보가 아닌 다른 센서로 검출한다는 것이 쉽지 않기 때문이다[2].

본 논문은 차량의 실내에 장착된 단안 카메라로 촬영된 도로영상에서 횡단보도와 정지선을 검출하는 알고리즘을 제안한다. 문제는 노면에는 매우 다양한 잡음들이 혼재해 있기 때문에 횡단보도의 줄무늬 표식과 정지선 표식을 안정적으로 검출하기 어렵다는 점이다. 횡단보도와 정지선 설치규정[3]을 보면 이들은 대개 도로의 진행방향에 수직인 방향으로 설치되도록 되어 있다. 그렇다고 차량에 설치된 카메라의 광축이 차량의 종방향 중심축과 일치한다고 가정할 수 없다. 더욱이 영상에서 이들 객체들의 검출 범위도 카메라로부터

객체들까지의 거리와 무관하게 정해져야 한다. 따라서 제안하는 횡단보도와 정지선 표식 검출알고리즘에 이들의 검출 방향이나 검출범위에 대해 강한 제한을 둘 수 없다.

횡단보도는 그림 1에 보인 바와 같이 노면에 흑백이 교차되는 줄무늬 형태의 표식으로 도색 되어 있다. 이러한 횡단보도 검출에 컴퓨터비전은 매우 자연스러운 방법일 수 있는데도, 실제 알려진 검출 알고리즘은 많지 않다. 최근 시각장애인을 위한 검출 알고리즘이 발표되고 있는데[4], 이 알고리즘들은 명료하게 단순한 영상에 초점을 맞추고 하프변환을 통해 횡단보도 표식들의 경계를 추출한다. 또한 카메라와 객체 사이에 위치 변동도 없다고 가정하고 있다. 그러나 본 연구에서 대상으로 하는 횡단보도 및 정지선 검출목적이 차량의 자율주행을 지원하는 데에 있기 때문에 검출대상과 카메라 사이의 거리변화가 크고, 방향변화도 크다. 또한 노면은 본질적으로 자연환경하에 있어서 조명과 그림자의 영향을 많이 받고, 서로 구별이 쉽지 않는 유사표식도 많다. 더욱이 백색인 정지선과 횡단보도 표식들은 시간의 흐름에 따라 닳아지고 퇴색이 되며, 차량과 같은 장애물에 의해 가려지기도 한다. 이러한 상황에서 횡단보도 표식과 정지선 검출에 하프변환은 적절한 알고리즘이라 할 수 없다.

정지선은 그림 1에 보인 바와 같이 노면상에 일직선으로 30~60cm의 폭으로 백색으로 그려지는데, 차량이 주행하는

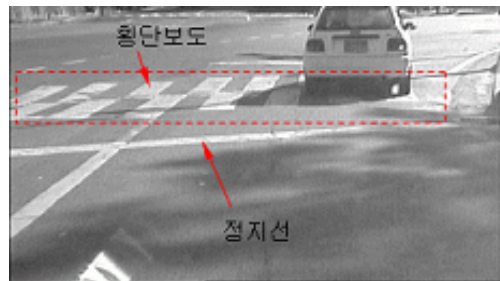


그림 1. 정지선과 횡단보도.
Fig. 1. Stop line and crosswalk.

* 책임저자(Corresponding Author)

논문접수: 2011. 1. 4., 수정: 2011. 3. 14., 채택확정: 2011. 4. 19.

이준웅: 전남대학교 산업공학과(시스템자동차 연구소)

(joonlee@chonnam.ac.kr)

※ 이 논문은 2010년도 전남대학교 학술연구비 지원에 의하여 연구되었음.

도로의 좌우 양방향 가운데 도로 중앙을 기준으로 좌측 또는 우측 한쪽에만 그려진다. 또한 정지선은 횡단보도가 없는 곳에서도 단독으로 그려지는 경우도 많다[3]. 정지선을 영상에서 육안으로 식별하는 것은 어렵지 않지만 컴퓨터로 식별하는 데는 까다로운 점이 많다. 첫째 정지선이 노면의 소실점에 가까이 있는 경우 매우 희미하게 보이며, 둘째 노면상에 백색으로 쓰여진 글씨, 고가도로나 다리 같은 곳의 구간과 구간의 이음쇠, 그림자에 의해 만들어지는 유사특징, 차량후면의 범퍼부 등이 실제 정지선과 구별이 어렵다. 이런 이유로 본 논문에서는 횡단보도와 정지선 표식 후보가 동시에 검출되었을 때 검출된 정지선 후보를 정지선으로 인정한다.

제안한 알고리즘은 횡단보도를 나타내는 줄무늬 표식이나 정지선 표식이 여러 잡음에 의해 퇴색되거나 손상되어 희미하게 보일 수 있지만 배경보다는 밝다고 가정한다. 이 가정을 토대로 에지정보[5,6]를 이용하여 횡단보도나 정지선 표식들의 경계에 있을 것으로 기대되는 픽셀들을 추출한다. 그리고 추출된 픽셀들에 블롭칼라링(blob coloring) [7]을 적용하여 픽셀들을 블롭별로 그룹핑함으로써 표식들의 경계를 개별 픽셀이 아닌 영역 단위로 처리를 할 수 기반을 마련한다. 블롭 하나하나의 개별 픽셀이 제공할 수 없는 정보추출을 가능하게 함으로 이들 정보를 이용하여 보다 정확히 표식의 경계를 추출할 수 있도록 한다.

논문의 구성은 다음과 같다. II 장에서는 에지분석과 에지정보 구성의 개요를 설명한다. III 장에서는 논문에서 제안한 블롭칼라링 알고리즘에 대해 기술한다. IV 장에서는 정지선 후보 추출, V 장에서는 횡단보도 후보 추출 과정을 제시한다. VI 장에서 실험결과, 그리고 VII 장에서 결론을 맺는다.

II. 에지분석과 에지정보의 구성

에지는 수학적으로 밝기함수의 그래디언트(gradient)로 정의된다[6]. 밝기영상 $f(x,y)$ 의 임의의 점 (x,y) 에서 그래디언트는 다음과 같이 벡터로 표시된다.

$$\nabla f = [G_x \ G_y]^T = \left[\frac{\partial f}{\partial x} \ \frac{\partial f}{\partial y} \right]^T \quad (1)$$

본 논문에서는 3x3 소벨연산자[6]를 사용하여 식 (1)의 G_x 와 G_y 를 구한 후 에지의 크기 ∇f 와 방향 $\alpha(x,y)$ 을 다음과 같이 계산한다[5].

$$\nabla f = \text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2} \approx |G_x| + |G_y| \quad (2)$$

$$\alpha(x,y) = \tan^{-1} \left[\frac{G_y}{G_x} \right] \quad (3)$$

여기에서 $\alpha(x,y)$ 의 범위는 영상의 수평축을 기준으로 0°~180°가 되도록 한다. 또한 G_x 와 G_y 의 부호에 따라 에지 픽셀들을 다음과 같이 8가지로 분류한다.

$$C(G_x, G_y) = i, \quad i = 1, 2, \dots, 8 \quad (4)$$

이 분류 기준은 표 1에 보인 바와 같다. 그림 2는 식 (4)와 표 1의 결과를 영상에 표기한 것이다.

표 1. G_x 와 G_y 에 의한 에지픽셀의 분류.

Table 1. Categorization of edge pixels by G_x and G_y .

$C(G_x, G_y)$	G_x 와 G_y 의 범위
1	$G_x > 0$ and $G_y > 0$
2	$G_x < 0$ and $G_y < 0$
3	$G_x > 0$ and $G_y = 0$
4	$G_x < 0$ and $G_y = 0$
5	$G_x > 0$ and $G_y < 0$
6	$G_x < 0$ and $G_y > 0$
7	$G_x = 0$ and $G_y > 0$
8	$G_x = 0$ and $G_y < 0$

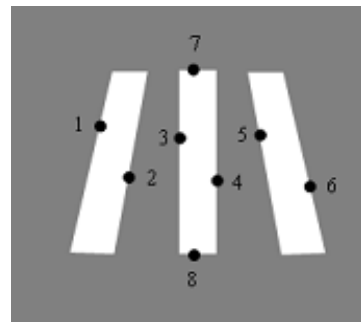


그림 2. G_x 와 G_y 에 의한 에지픽셀의 분류.

Fig. 2. Categorization of edge pixels by G_x and G_y .

영상에서 노면객체의 경계에 해당될 가능성이 큰 픽셀들을 추출하기 위해 다음 두 가지 처리를 한다. 첫째, ∇f 가 작은 픽셀들을 제거한다. 이 때 ∇f 에 대한 임계값(threshold)을 선택하는 것이 어려운 문제인데, 그 이유는 임계값에 따라 제거되지 않아야 할 픽셀까지 제거될 수 있기 때문이다. 본 논문에서는 참고문헌 [2]에 소개된 적응형 임계값 설정법을 사용한다. 둘째, 임계값 적용후 노면객체의 경계일 가능성이 더욱 큰 픽셀들을 남기고, 아울러 세선화(thinning) 효과까지 얻기 위해 NLMS (Non-Local Maximum Suppression) [8]을 적용한다.

III. 블롭칼라링

블롭칼라링은 0과 1로 이루어진 2진 영상이 주어졌을 때 값이 1인 픽셀들을 대상으로 서로 연결된 픽셀들을 블롭단위로 모으는 기법으로 L모양의 템플릿이 사용되고 있다[7]. 이 템플릿에 의해 추출된 블롭은 상하, 좌우의 4방향 연결성(four connectivity)은 성립되지만 좌우 대각선 방향까지 포함한 8방향 연결성은 만족되지 않는다. 또한 스캔(scan) 방향도 위에서 아래로 그리고 좌에서 우로 이루어진다. 그러나 본 연구에서 추출대상인 노면표식들의 경계를 블롭단위로 추출하기 위해서는 8방향 연결성을 만족하는 템플릿이 필요하다. 따라서 본 논문에서는 8방향 연결성을 만족하고 아래서 위로, 그리고 좌에서 우로 스캔할 수 있는 그림 3에 보인 바와 같은 템플릿을 제안한다. 아래에서 출발하여 위로 스캔하려는 이유는 영상의 하단부가 차량에서 가깝기 때문에 영상의 하단부에 있는 정지선이나 횡단보도 표식들 경계의 블롭을 상단부보다 우선적으로 추출해야 할 필요가 있기 때문이다.

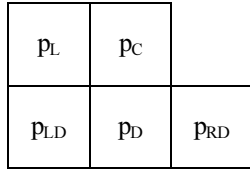


그림 3. 블러프칼라링 템플릿.

Fig. 3. Template for blob coloring.

그림 3에서 p_C 가 현재 고려하는 픽셀의 위치로서 $f(p_C)$ 는 픽셀 p_C 의 값이다. p_C 를 기준으로 p_L 은 좌측 픽셀, p_{LD} 은 좌하측 픽셀, p_D 은 하측 픽셀, p_{RD} 은 우하측 픽셀을 가리킨다. 제안된 블러프칼라링 알고리즘의 유사코드는 다음과 같다.

```

초기 칼라 k=1, 영상을 아래서 위로, 좌에서 우로 스캔한다.
if f(p_C) = 0 then continue
if f(p_D) and f(p_L) = 0 then color(p_C) = color(p_D)
elseif f(p_L) and f(p_D) = 0 and f(p_{RD}) = 0 then color(p_C) = color(p_L)
elseif f(p_L) = 0 and f(p_D) = 0 and f(p_{RD}) = 0 and f(p_{LD}) then
color(p_C) = color(p_{LD})
elseif f(p_L) = 0 and f(p_D) = 0 and f(p_{LD}) = 0 and f(p_{RD}) then
color(p_C) = color(p_{RD})
elseif f(p_D) and f(p_L) then
    if (color(p_D) = color(p_{LD})) color(p_C) = color(p_D)
    else { color(p_C) = color(p_D)
          color(p_L) is equivalent to color(p_D)
        }
}
elseif f(p_L) and f(p_{RD}) then
    if (color(p_L) = color(p_{RD})) color(p_C) = color(p_L)
    else { color(p_C) = color(p_L)
          color(p_L) is equivalent to color(p_{RD})
        }
}
elseif f(p_{LD}) and f(p_{RD}) then
    if (color(p_{LD}) = color(p_{RD})) color(p_C) = color(p_{LD})
    else { color(p_C) = color(p_{LD})
          color(p_{LD}) is equivalent to color(p_{RD})
        }
}
else
    color(p_C) = k
    k = k + 1
    
```

영상의 모든 픽셀에 대해 스캔이 끝나고 나면 동일 칼라 (equivalent color)로 지정된 픽셀들의 경우 이들의 칼라를 단일 칼라로 일치시킨다.

블러프칼라링은 블러프에 대한 기하학적인 정보를 얻기 위해 행한다. 하나의 블러프는 서로 연결된 픽셀들을 분리한 것으로서 본 연구에서는 블러프단위로 블러프의 길이, 기울기, 중점 등과 같은 정보를 산출한다. 블러프의 중점 (\bar{x}, \bar{y}) 는 다음 식 (5)에 의해 계산된다.

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}} \quad (5)$$

여기에서 $m_{pq} = \sum x_i^p y_i^q$ 이다. 블러프에 선형회귀[2,5]를 적용하면 회귀 파라미터인 기울기 b 와 y 축 절편 a 을 다음과 같이 계산할 수 있다.

$$b = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2} \quad (6)$$

$$a = \bar{y} - b\bar{x} \quad (7)$$

IV. 정지선 후보 추출

1. 정지선 경계픽셀 후보 선정

정지선 검출은 “정지선 표식은 배경보다 밝다.”는 가정을 토대로 정지선 표식의 상단경계와 하단경계가 될 가능성이 있는 픽셀추출로부터 시작된다. 그림 4의 p_1 이 정지선 표식의 하단 경계픽셀 후보가 되려면

- 조건 1: $f(p_1)$ 이 배경보다 밝고,
- 조건 2: $C(G_x, G_y)$ 가 2이거나 5이거나 8이고,
- 조건 3: 그림 4의 A영역보다 B영역이 밝아야 하며,
- 조건 4: $\alpha(p_1)$ 도 30°이하거나 150°이상이어야 한다.

반면에 그림 3의 p_2 가 정지선 표식의 상단 경계픽셀 후보가 되려면

- 조건 1: $f(p_2)$ 가 배경보다 밝고,
- 조건 2: $C(G_x, G_y)$ 가 1이거나 6이거나 7이고,
- 조건 3: 그림 4의 C영역보다 B영역이 밝아야 하며,
- 조건 4: $\alpha(p_2)$ 도 30°이하거나 150°이상이어야 한다.

여기에서 조건 4에 사용된 각도는 정지선 경계픽셀이 지나치게 많아지는 것을 막기 위해 실험적으로 선정된 값이다.

이러한 조건을 적용하여 정지선 표식의 상·하단부의 경계상에 있을 것으로 기대되는 후보픽셀들을 검출한 후 나 홀로 있는 픽셀들은 제거한다. 그리고 남은 픽셀들을 대상으로 제안된 블러프칼라링(blob coloring)을 적용하여 서로 연결된 픽셀들을 블러프들로 분리해 낸다.

2. 정지선 후보 결정

2.1 블러프 필터링

블러프칼라링에 의해 추출된 정지선 상·하한경계 블러프들 가운데 블러프의 좌우 길이(W)보다 상하 높이(H)가 큰 것들은 정지선 경계일 가능성이 매우 낮으므로 제거한다. 이 때 블러프 제거를 위해 사용된 폭과 높이의 비(H/W)는 1.5로서 실험적으로 설정한 값이다. 제거되지 않고 남은 블러프들 가운데 식 (6)과 (7)의 회귀 파라미터인 기울기와 y 축 절편이 유사하고, 중첩되어 있지 않고, 서로 떨어진 정도가 크지 않은 것들은 연결하여 단일 블러프로 만든다.

2.2 정지선 후보 결정

추출된 정지선 하한경계 블러프 각각에 대해 영상의 수직방향 위쪽으로 정지선 설치매뉴얼에 규정된 정지선폭 60cm에

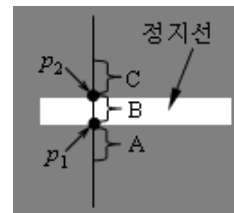


그림 4. 정지선 표식의 상·하 경계 픽셀 조건.

Fig. 4. The condition to be a candidate pixel of upper and lower boundaries of a stop line.

해당된 픽셀거리내에 정지선 상한경계 블럽이 존재하고, 이때 하한 블럽이나 상한 블럽 가운데 최소한 하나의 수평길이 영상의 수평폭의 1/3 이상이면 이 정지선 상·하한경계 블럽쌍을 정지선 후보로 결정한다. 이 때 정지선의 기준은 하한 블럽이 된다. 여기에서 공간상의 일정거리에 대한 픽셀 거리는 카메라 캘리브레이션[9] 파라미터를 이용하여 구한다. 이에 대한 계산방법은 참고문헌 [10]에 소개되어 있다.

만약 정지선 후보가 하나도 검출되지 않을 경우 횡단보도 검출은 수행하지 않는다. 반면에 정지선 후보가 검출되면 이 정지선 후보위치를 기점으로 수직 위쪽에 횡단보도 줄무늬 표식이 있는지 탐색한다.

V. 횡단보도 후보 추출

1. 횡단보도 표시 설치 규정

우리나라의 횡단보도 설치매뉴얼[3]을 보면 횡단보도는 “백색으로 폭원은 4m 이상이고 노면의 전폭을 가로질러 표시하는 지브라식으로 설치하여야 한다.”로 규정되어 있다. 또한 횡단보도의 폭원이 4m를 초과하는 경우는 2m 단위로 확폭하되 6m 이상인 경우는 도로폭원을 2등분하여 설치하도록 되어 있다. 이를 그림으로 표현하면 그림 5와 같다. 그림 5에 보인 바와 같이 정지선 전방으로 2~5m 정도 떨어진 곳으로부터 최대 8m의 폭원을 가진 횡단보도 표시가 설치된다. 따라서 IV장에서 제시된 방법에 의해 영상에서 정지선 후보가 검출되면 정지선 후보가 있는 위치로부터 위쪽으로 2~10m 범위에 대응되는 픽셀거리내에서 횡단보도 표시가 있는지 탐색을 한다.

2. 횡단보도 표시의 경계후보 픽셀 추출

횡단보도 표시의 줄무늬 검출은 “표식은 배경보다 밝다.”는 가정을 토대로 줄무늬 표시의 좌우 경계가 될 가능성이 있는 후보 픽셀들의 추출로부터 시작된다. 그림 2에 보인 하얀색의 표식들이 영상에 나타난 횡단보도 줄무늬 표식들이라 가정을 해보자. 이 때 영상의 어떤 픽셀 $p(x, y)$ 가 줄무늬 표시의 좌측 경계가 되기 위해서는

- 조건 1: $f(p)$ 가 배경보다 밝고,
- 조건 2: $C(G_x, G_y)$ 가 1이거나 3이거나 5이고,
- 조건 3: p 의 좌측보다는 우측이 더 밝아야 하며,
- 조건 4: $\alpha(p)$ 도 6°보다 크고 174°미만이어야 한다.

반면에 $p(x, y)$ 가 줄무늬 표시의 우측 경계가 되기 위해서는

- 조건 1: $f(p)$ 가 배경보다 밝고,
- 조건 2: $C(G_x, G_y)$ 가 2이거나 4이거나 6이고,
- 조건 3: p 의 우측보다는 좌측이 더 밝아야 하며,
- 조건 4: $\alpha(p)$ 도 6°보다 크고 174°미만이어야 한다.

여기에서 조건 4에 사용한 각도는 수평선에 가까운 픽셀들의 선택을 막기 위해 실험적으로 정한 값이다. 이와 같은 조건을 적용하여 횡단보도를 나타내는 줄무늬 표시들의 좌·우 경계상에 있을 것으로 기대되는 후보픽셀들을 검출한 후 나 홀로 떨어져 있는 픽셀들은 제거한다.

줄무늬 표시는 45~50cm의 일정한 폭을 갖는다는 설치규정을 적용하여 추출된 후보픽셀들에 대해 다음과 같은 필터링을 실시한다. 줄무늬 표시 폭 50cm에 해당되는 픽셀거리를 이용하여 좌측 후보 픽셀의 경우는 우측으로 이 픽셀거리만큼 떨어진 곳에 우측 후보 픽셀이 존재하는지 확인한다. 우측 후보 픽셀의 경우도 동일한 방법으로 이 픽셀거리만큼 좌측으로 떨어진 곳에 좌측 후보 픽셀이 존재하는지 확인한다. 만약 좌·우 짝꿍 픽셀이 존재하면 이 두 픽셀 사이의 평균 밝기와 밝기값의 분산을 계산하여 현재의 후보 픽셀이 정말 줄무늬 표시의 경계일 가능성이 큰지 판단한다. 즉 짝꿍으로 고려되는 두 픽셀 사이의 평균 밝기가 배경보다 밝고, 분산이 정해진 임계값보다 작으면 고려하고 있는 픽셀은 표시경계의 후보픽셀로서 인정이 되지만, 짝꿍 픽셀이 존재하지 않거나 평균 밝기가 배경보다 어둡거나 분산이 크면 현재 고려 중인 픽셀은 후보픽셀에서 탈락한다. 이 과정에서 사용하는 분산에 대한 임계값은 실험적으로 결정하는데, 본 연구에서는 600을 택하였다.

후보픽셀로 선정된 픽셀들에 대해 블럽칼라링 알고리즘을 적용하여 좌·우 경계픽셀들의 블럽을 추출한다. 추출된 블럽들 가운데 블럽의 상하 높이가 그림 5의 파라미터 d_1 의 거리인 8m의 픽셀거리보다 큰 블럽 또는 블럽의 기울기가 γ 보다 작거나 173°보다 큰 블럽은 제거한다. 왜냐하면 이들은 줄무늬 표시의 경계일 가능성이 매우 낮기 때문이다. 남아 있는 블럽들 가운데 식 (6)과 (7)의 기울기와 y-절편의 값이 유사하고 떨어진 거리가 그림 5의 파라미터 s 의 픽셀거리보다 짧은 것들은 하나의 블럽으로 통합한다.

3. 횡단보도 표시 후보 결정

먼저 횡단보도 줄무늬 표시의 좌측 블럽 b_L 에 대해 자신의 우측에 자신과 짝을 이룰 우측 블럽 b_R 이 존재하는지 탐색한다. 이 때 짝의 선택조건은 다음과 같다.

- 조건 1: 영상의 수직 방향을 기준으로 두 블럽은 조금이라도 겹치는 영역이 있어야 한다. 즉 그림 6의 ξ 가 0보다 커야 한다.
- 조건 2: 두 블럽의 양쪽 끝(그림 6의 $(s_L, e_L), (s_R, e_R)$)이나 식 (5)에 의해 구한 중점(그림 6의 (m_L, m_R)) 가운데 어느 한 부분이라도 그림 5의 파라미터 l_1 의 50cm의 픽셀거리의 범위 내에 존재해야 한다.
- 조건 3: 그림 6에 보인 b_L 의 기울기 α 가 90°보다 크고 b_R 의 기울기 β 가 90°보다 작은 경우 사이각, $\phi(=\alpha-\beta)$ 는 100°보다 작아야 한다. 여기서 100°는 실험적으로 선정한 것이다. 조건 3이 없을 경우 노면에 새겨진 화살표식이나 글씨 등이

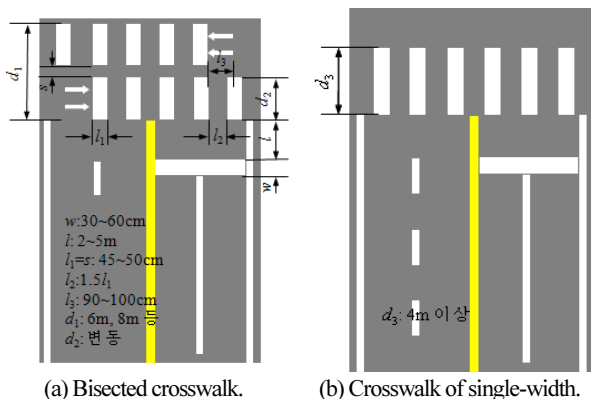


그림 5. 횡단보도 표시 설치 예시.

Fig. 5. Illustrations of crosswalk installation.

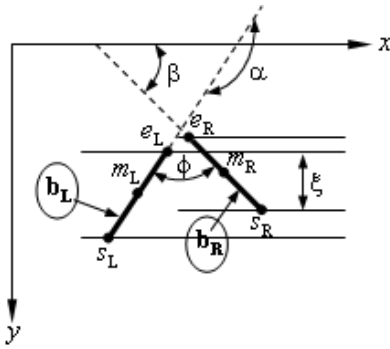


그림 6. 좌·우 블러블의 정보와 사이각.
Fig. 6. Data of left and right blobs and angle between them.

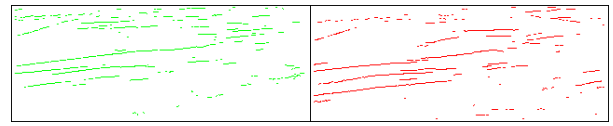
횡단보도의 줄무늬 표식으로 오인될 수 있다. 만약 b_L 의 짝으로 선정된 우측 블러블이 둘 이상일 경우 이들 가운데 블러블의 길이가 가장 긴 것을 b_L 의 짝으로 선택한다. 상기 과정을 통해 줄무늬 표식의 좌측 블러블의 짝으로 뽑힌 우측 블러블들 각각에 대해서 현재 자신의 짝인 좌측 블러블보다 더 좋은 짝이 있을지도 모르므로 짝을 찾지 못한 좌측 블러블들 가운데 자신의 짝이 될만한 블러블이 존재하는지 위에서 제시한 동일한 조건들을 적용하여 탐색한다. 만약 그런 블러블이 여럿 존재하면 그 중에 길이가 가장 긴 것을 택하고, 이것과 이미 짝으로 뽑혀 있는 좌측 블러블과의 길이를 비교해서 길이가 더 긴 것을 최종 짝으로 선택한다. 이와 같은 방법으로 횡단보도 표시의 줄무늬 표식의 좌·우 경계 블러블의 쌍들이 결정이 되면, 쌍의 수에 따라 횡단보도의 존재유무를 결정한다. 본 연구에서는 쌍의 수가 4개 이상일 경우 횡단보도가 존재하는 것으로 판정을 한다.

VI. 실험결과

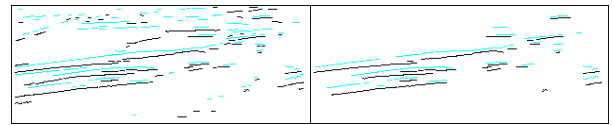
실험은 차량의 실내에 설치된 전방 주시 카메라에 의해 촬영된 320x240 크기의 도로영상으로 이루어졌다. 그림 7(a)에 제시된 영상에서는 육안으로 멀리 영상의 좌측부분에 횡단보도가 흐릿하게 보이지만, 우측 부분은 차량에 가려져서 보이지 않는다. 정지선은 횡단보도 앞으로 그려져 있었지만, 너무 멀리 있고 또 차량에 가려져서 보이지 않는다. 또한 영상의 하단부에는 그림자에 의해 어두운 부분과 밝은 부분이 뚜렷하게 교차되어 나타나 있다. 그림 7(b)는 정지선 상·하경계상에 있을 것으로 기대되어 추출된 픽셀들을 표현한 것이다. 그림 7(c)는 정지선 상·하경계 후보픽셀들에 블러블칼라링을 적용하여 추출한 블러블을 나타낸 것이다. 그림 7(c)의 오른쪽 그림은 왼쪽 그림의 블러블들에 대해 정지선 경계의 블러블들로서는 적절치 않은 것들을 제거한 후의 블러블들을 표현한 것이다. 이들 가운데 정지선 조건을 만족하는 상·하경계블러블이 한 쌍 존재하였고, 이 블러블들의 위치가 그림 7(d)에 보인 것처럼 정지선 후보가 되었다. 정지선후보가 검출됨에 따라 횡단보도의 줄무늬 표식이 존재하는지 탐색이 이루어졌고, 그 결과 추출된 줄무늬 표식의 좌·우경계 후보픽셀들이 그림 7(e)에 표시되었다. 이들 픽셀들에 대해 블러블칼라링을 적용하였고, 이 때 추출된 블러블들은 그림 7(f)에 보인 바와 같다. 그러나 이들 블러블들 가운데는 어느 것도 줄무늬표식의 조건을 만족하는 좌·우쌍이 없었다. 따라서 그림 7(a)에 보



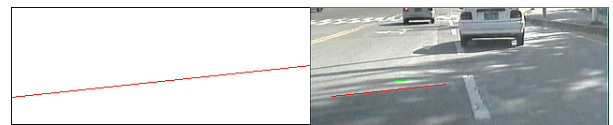
(a) Image.



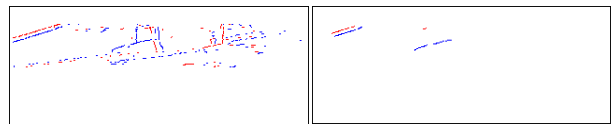
(b) Pixels on top and bottom boundaries of stopline candidates. (left: top boundary, right: bottom boundary)



(c) Blobs of boundaries of stopline candidates. (left: before filtering, right: after filtering)



(d) A detected stopline candidate.

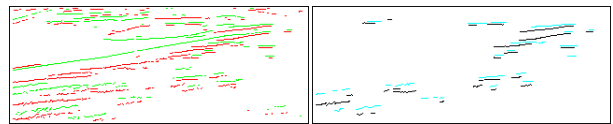


(e) Pixels on left and right boundaries of probable crosswalk stripes. (f) Blobs of left and right boundaries of probable crosswalk stripes.

그림 7. 정지선과 횡단보도 추출과정.
Fig. 7. Extraction process of stop line and crosswalk.



(a) Image.



(b) Pixels on top and bottom boundaries of stopline candidates. (c) Blobs of top and bottom boundaries of stopline candidates.

그림 8. 정지선이 없는 영상.
Fig. 8. An image without a stop line.

인 영상에서는 비록 그림자로 인해 정지선후보가 검출되었어도 횡단보도에 관련된 표식이 검출되지 않았기 때문에 정지선과 횡단보도는 없다고 결정되었다.

그림 8은 그림 7에 보인 영상에 이어진 영상에 적용된 결과로 영상에도 정지선이나 횡단보도가 없고 알고리즘 적용 결과에서도 정지선이 검출되지 않았다. 그림 8(b)에 보인 바와 같이 정지선의 상·하경계상에 있을 것으로 기대되는 픽셀들을 추출하였고, 이 픽셀들에 블러블칼라링을 적용하여 그림 8(c)에 보인 바와 같은 블러블을 추출하였지만 정지선 조건을 만족하는 상·하경계 블러블쌍은 존재하지 않았다.

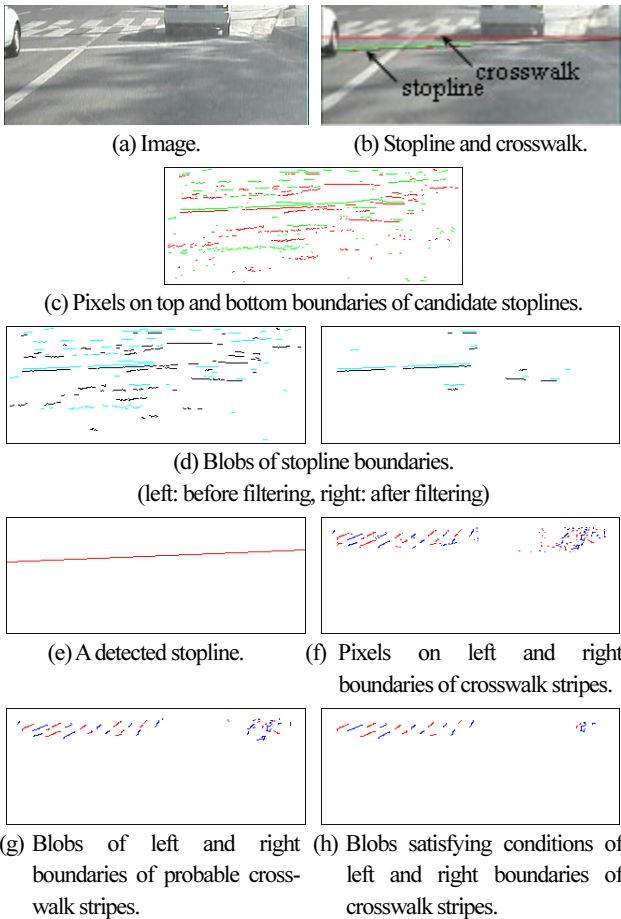


그림 9. 정지선 및 횡단보도가 보인 영상에서 이들의 추출.

Fig. 9. Extraction of stop line and crosswalk from an image with them.

그림 9에 보인 실험결과는 그림 9(a)의 영상에 보였듯이 횡단보도의 오른쪽이 차량에 가려져 있지만 정지선과 횡단보도가 성공적으로 검출된 경우이다. 그림 9(b)와 (e)에 보인 바와 같이 정지선후보가 추출되었고, 이 결과에 의해 횡단보도 줄무늬 표식의 경계픽셀 검출이 그림 9(f)에 보인 바와 같이 이루어졌다. 이 경계픽셀들에 대해 블롭칼라링이 적용되어 그림 9(g)에 보인 블롭들이 추출되었다. 이들 블롭들 가운데 10쌍이 횡단보도 줄무늬표식 조건을 만족하였고, 이 결과에 의해 추출된 정지선과 횡단보도를 그림 9(b)에 표시하였다.

그림 10은 그림의 좌측에 보인 다양한 영상에 대해 정지선과 횡단보도를 추출한 결과를 그림 우측에 보인 것이다. 그림에 보인 바와 같이 도심지 영상에서는 제대로 되어 있는 정지선이나 횡단보도 표식이 없다. 모두 닳아지고(그림 10(a)-(g)), 가려지고(그림 10(a), (e), (g), (h)), 또 역광이 비추기도 하는(그림 10(c), (d)) 등 손상이 되어 있다. 또한 이들 표식들은 차량에 가까이 있을 수 있고(그림 10(b)-(c), (f)) 또 멀리 있기도(그림 10(e), (h)) 하다. 차량진행방향에 직각일 수도 있고 기울어져 있기도 하다. 이러한 다양한 상황에서 본 논문에서 제안된 알고리즘은 성공적으로 정지선이나 횡단보도 표식을 검출하였다.

그림 11의 실험결과는 횡단보도 표식은 없이 정지선만 있거나(그림 11(a)) 정지선과 유사한 표식이 있는 영상(그림

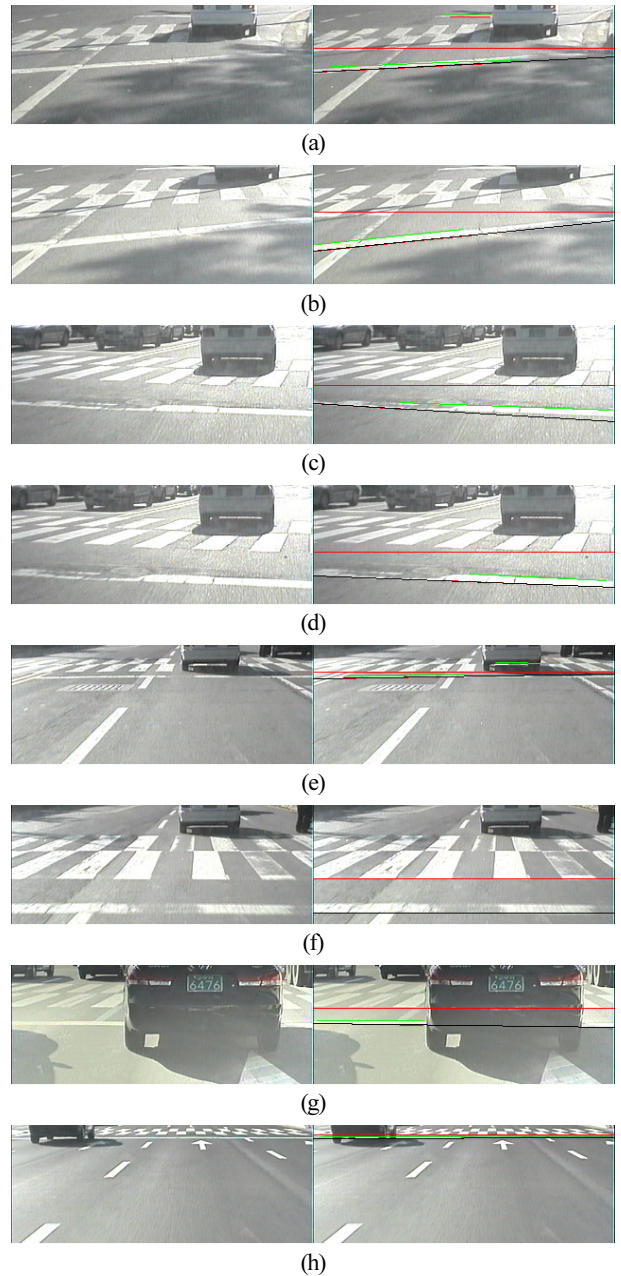


그림 10. 다양한 영상에서 정지선과 횡단보도 추출.

Fig. 10. Extraction of stop line and crosswalk from various images.

11(b)-(h))을 대상으로 제안된 알고리즘을 적용한 것이다. 대부분의 경우 실제로는 정지선이 아닌데 육안으로 보아도 정지선처럼 보인다. 그림 11(b)-(e)는 노면에 새겨진 글씨가 정지선조건을 만족한 경우이다. 그림 11(e)와 (f)는 도로의 이음매가 정지선처럼 보인 경우이다. 그림 11(g)는 차선표식이 정지선처럼 보인 경우이다. 그림 11(h)는 그림자가 정지선처럼 보인 경우이다. 이 실험 예에서 왜 정지선과 횡단보도 검출을 동시에 수행하는지 알 수 있다. 정지선과 유사한 표식이 노면에는 너무 많다. 따라서 정지선만 검출하여 차량의 주행을 제어하기에는 분명히 한계가 있다.

그림 12에 제시한 실험결과는 많은 글씨가 근접된 공간에 새겨진 경우로서 다행히 제안된 알고리즘이 기대한 대로 결과를 도출하였다. 그러나 그림 12(f)에 보인 바와 같이 블롭

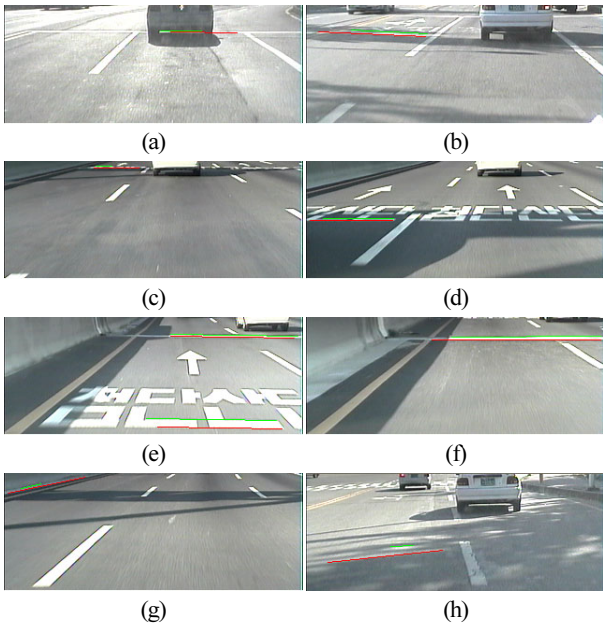


그림 11. 다양한 영상에서 정지선 추출.
Fig. 11. Extraction of stop lines from various images.

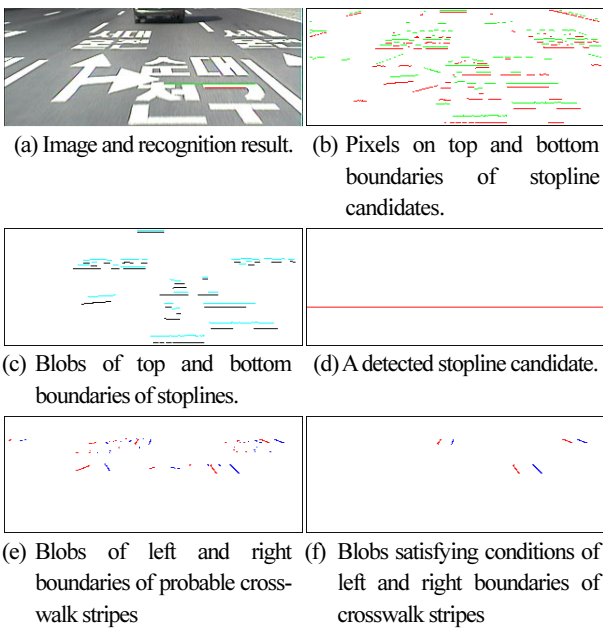


그림 12. 글씨를 정지선과 횡단보도로 오인할 가능성이 있는 예.
Fig. 12. An example with possibility to recognize letters as stop line and crosswalk.

세 쌍이 줄무늬표식 경계조건을 만족하였으므로 글씨를 횡단보도로 오인할 가능성도 있다.

그림 13은 정지선과 횡단보도가 카메라로부터 멀리 있는 경우의 실험결과를 보인 것이다. 횡단보도 왼쪽의 줄무늬 표식들이 잘 보이지 않고, 정지선과 줄무늬 표식들의 훼손 정도가 심하지만 정지선과 횡단보도가 성공적으로 검출되었다.

그림 14는 정지선과 횡단보도 대부분이 카메라에 근접된 전방 차량에 의해 가린 경우의 실험결과를 보인 것이다. 횡단보도 줄무늬 표식과 정지선이 일부만 보였지만 정지선과

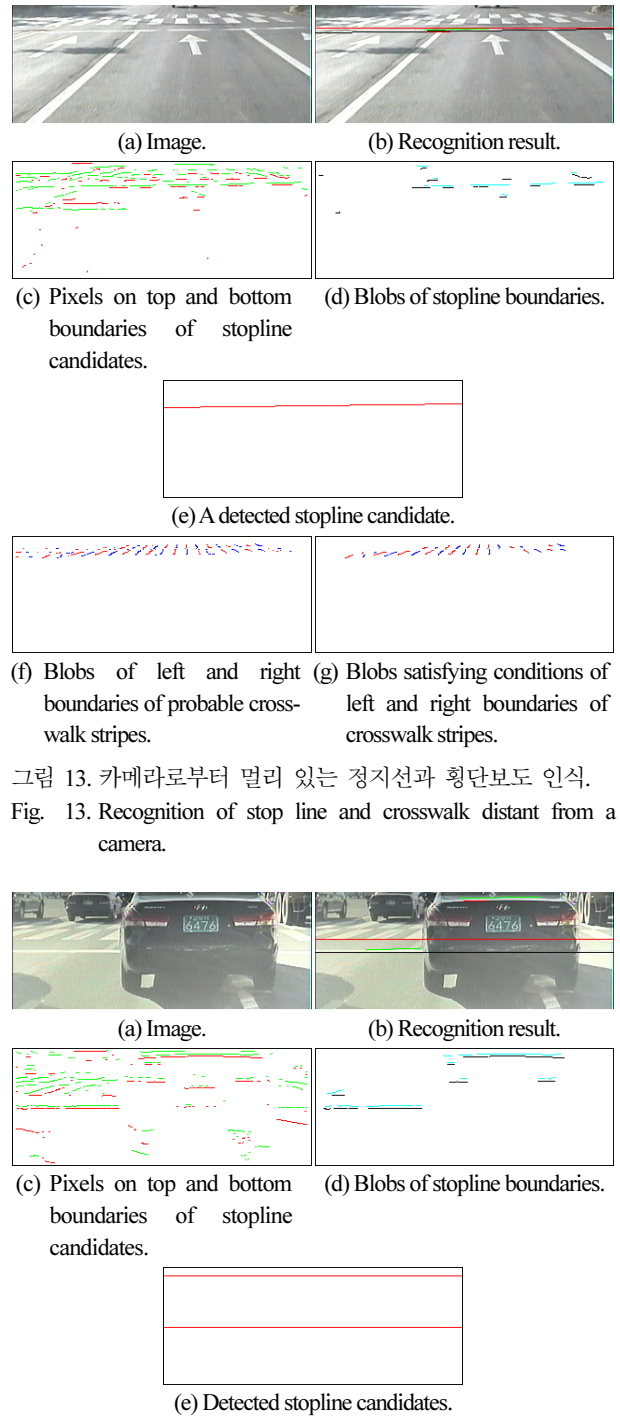


그림 13. 카메라로부터 멀리 있는 정지선과 횡단보도 인식.
Fig. 13. Recognition of stop line and crosswalk distant from a camera.

그림 14. 카메라로부터 멀리 있는 정지선과 횡단보도 인식.
Fig. 14. Recognition of stop line and crosswalk distant from camera.

횡단보도가 성공적으로 검출되었다. 이 실험결과에서 한 가지 주목할 점은 정지선 후보가 실제 정지선과 거짓 정지선 이렇게 두 군데에서 검출되었다는 것이다. 거짓 정지선은 그

림 14(b)와 (e)에 보인 바와 같이 차량의 후미에서 검출되었다. 그러나 이 거짓 정지선은 이 윗 부분에 횡단보도와 관련된 픽셀들이 검출되지 않았기 때문에 최종 실험결과에 아무런 영향을 미치지 않았다.

제안한 알고리즘은 램프로나 십자로와 같은 곳에서 차량이 우회전할 때 갑자기 나타나는 정지선과 횡단보도 등을 제외하면 대부분의 정지선과 횡단보도를 검출한다. 그러나 앞의 실험결과에서 밝혔듯이 횡단보도가 영상에 보인다고 해서 다 검출할 수는 없다. 소실점 가까운 곳에서는 육안으로는 보여도 횡단보도나 정지선 표식의 윤곽정보가 너무 흐릿하기 때문에 검출에 실패할 수 있다. 또한 차량들에 의해 횡단보도나 정지선 표식이 70-80% 이상 가려진 경우도 검출에 실패할 때가 있다. 이런 경우를 제외한다면, 즉 정지선 표식과 횡단보도 표식이 카메라에 20-30m 이내로 근접하고, 차량 등에 의해 심하게 가려져 있지 않는다면 정지선과 횡단보도 검출은 성공적으로 이루어졌다.

VII. 결론

본 논문에서 제안된 정지선과 횡단보도 검출법은 크게 두 가지 장점이 있다. 첫째, 이들 객체의 표식이 노면보다 밝다는 점에 착안하여 간단한 소벨 에지연산자를 사용하여 얻은 정보를 효율적으로 활용했다는 점이다. 둘째, 블립칼라링 기법을 도입하여 픽셀 중심이 아닌 영역중심으로 접근함에 따라 사용할 정보의 범위를 넓힘과 동시에 검출의 신뢰성을 높였다는 점이다. 이외에도 검출에 소요되는 시간이 길지 않다는 점도 장점이다. 인텔 코어 2쿼드 PC에서 320×240 크기의 영상을 대상으로 초당 20프레임 이상 처리가 가능하였다.

다양한 영상을 이용한 실험을 통해 제안된 알고리즘이 노면으로부터 정지선과 횡단보도를 정확하게 검출할 수 있음을 보였다.

참고문헌

[1] D. A. Pomerleau and T. Jochem, "Rapidly adapting machine vision for automated vehicle steering," *IEEE Expert Intelligent Systems and Their Application*, vol. 19, no. 27, 1996.

- [2] J. W. Lee and U. K. Yi, "A lane-departure identification based on LBPE, Hough transform, and Linear regression," *CVIU*, vol. 99, no. 3, pp. 359-383, 2005.
- [3] 교통노면표시 설치·관리 매뉴얼, 경찰청교통기획과, 2005.
- [4] J. M. Coughlan and H. Shen, "A fast algorithm for finding crosswalks using figure-ground segmentation," *Proc. 2nd Workshop on Applications of Computer Vision*, 2006.
- [5] J. W. Lee and U. K. Yi, "A Lane-Departure identification based on linear regression and symmetry of lane-related parameters," *Journal of Control, Automation, and Systems Engineering*, vol. 11, no. 5, pp. 435-444, 2005.
- [6] R. G. Gonzalez and R. E. Woods, *Digital image processing*, Addison-Wesley, 1992.
- [7] D. H. Ballard and C. M. Brown, *Computer Vision*, Prentice Hall, 1982.
- [8] O. Faugeras, *Three-Dimensional Computer Vision - A Geometric Viewpoint*, the MIT Press, 1993.
- [9] R. Y. Tsai, "A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. Robotics Automation*, vol. RA-3, no. 4, pp. 323-344, 1987.
- [10] G. Y. Song and J. W. Lee, "Detection of preceding vehicles based on a multistage combination of edge features and horizontal symmetry," *Journal of Institute of Control, Robotics and Systems (in Korean)*, vol. 14, no. 7, pp. 679-688, 2008.



이 준 응

1984년 전남대학교 산업공학과(공학사).
1986년 KAIST 산업공학과(공학석사).
1997년 KAIST 자동차 및 설계공학과
(공학박사). 1986년~2000년 기아자동차
연구소 및 현대기아연구개발본부 전자
개발센터 근무. 2000년~현재 전남대학교
산업공학과 교수. 관심분야는 지능형안전자동차, 차량자율주
행, 컴퓨터비전, 센서퓨전.