

차량 뒷바퀴 윤곽선을 이용한 근거리 전방차량인식

Recognition of a Close Leading Vehicle Using the Contour of the Vehicle's Rear Wheels

노 광 현, 한 민 홍

(Kwang-Hyun Rho and Min-Hong Han)

Abstract : This paper describes a method for detecting a close leading vehicle using the contour of the vehicle's rear wheels. The contour of a leading vehicle's rear wheels in the front road image from a B/W CCD camera mounted on the central front bumper of the vehicle, has vertical components and can be discerned clearly in contrast to the road surface. After extracting positive edges and negative edges using the Sobel operator in the raw image, every point that can be recognized as a feature of the contour of the rear wheels is extracted. Clustering is performed to them and the closest contour of the leading vehicle wheel is determined. This process can detect the presence of a close leading vehicle, and it is also possible to calculate the distance to the leading vehicle and the lateral deviation angle. This method might be useful for developing an LSA (Low Speed Automation) system that can relieve driver's stress in the stop-and-go traffic conditions encountered on urban roads.

Keywords : close leading vehicle, contour of the vehicle's rear wheels, Sobel operator, clustering, stop-and-go

I. 서론

컴퓨터비전은 전자기술의 발달로 고해상도의 영상 데이터를 분석할 수 있을 뿐만 아니라, 실시간 동영상처리가 가능한 처리 속도를 얻게 됨에 따라 실세계 인식 관련 분야에서 폭넓게 사용되고 있다. 최근에는 운전자의 부주의로 인한 사고를 예방하기 위한 첨단장치 및 자율주행차량 개발을 위한 전방장애물인식, 차선인식, 측방 및 후방차량인식, 졸음운전감지 등에 컴퓨터비전이 적용되고 있다. 이 중에서도 컴퓨터비전을 이용한 전방장애물인식은 고가의 거리계인 Laser Radar, Millimeter-Wave Radar(MWR)에 비해 장애물까지의 거리 측정의 정확도 및 정밀도에서는 뒤지지만 거리계로 얻기 힘든 중요한 장애물 특징정보를 추출하여 장애물을 인식할 수 있고, 동시에 차선도 인식할 수 있으며, 훨씬 저렴하게 시스템을 구성할 수 있는 장점 때문에 많은 연구가 진행되고 있다[1]. 도로상의 장애물에는 여러 종류가 있지만 대부분 차량이 이에 속하므로 많은 연구가 전방차량인식 문제로 국한되어 진행되어 왔으며 현재 사용되고 있는 장애물을 인식을 위한 세가지 센서의 특징을 비교하면 표 1과 같다.

현재까지 컴퓨터비전을 이용한 전방차량인식에 관한 많은 연구가 진행되어 왔다. 전방차량인식은 일반적으로 전방차량의 존재 유무 판단 및 위치 측정을 의미한다. Graefe[2]는 영상내의 전방차량 외곽선이 한 쌍의

수직선으로 나타나는 특징을 이용하였고, Zielke[3]와 Kuehnle[4]은 전방차량 후방 모습이 차량 중앙을 기준으로 대칭성을 띠는 특징을 이용하였다. Burie[5]는 스

표 1. 센서들의 특징.

Table 1. Features of sensors.

	Laser Radar	MWR	Image Sensor
Obstacle Recognition	×	×	○
Weather Performance	△	○	△
Lane Recognition	×	×	○

○ : good △ : satisfactory × : poor

스테레오비전에서 왼쪽 영상을 이용하여 오른쪽 영상을 예측하는 방법을 사용하여 인식하였고, Broggi[6]는 스테레오비전에서 역투영(inverse perspective)된 두 영상을 비교하는 기법을 사용하여 전방차량을 인식하였다. Franke[7]도 스테레오비전을 이용하여 도심지 도로에서 Stop-and-Go 시스템에 적용하기 위한 차량과 보행자 인식 방법을 제안했다. Giachetti[8]와 Enkelmann[9]은 예측한 광류 흐름(Optical Flow)과 실제 계산된 광류 흐름을 비교하는 방법을 사용하였다. Betke[10]는 여러 형태의 차량 이미지를 이용하는 원형 정합(template matching) 방법을 사용하였다. 또한, 근거리 측방 차량의 검은 타이어 안쪽의 흰색 휠이 원의 형태를 띠고 측면을 바라보도록 설치된 카메라로 입력되는 영상에서 타원으로 보이는 휠의 특징을 Hough transform에 의한 방법으로 탐지하고 거리 및 방향을 측정하는 방법에 관한 연구도 수행된 바 있다[11].

접수일자 : 2000. 5. 29., 수정완료 : 2000. 12. 23.

노광현, 한민홍 : 고려대학교 산업공학과(khrho@dreamwiz.com /mhhan@korea.ac.kr)

※ 본 연구는 한국과학재단 국제공동연구(995-1000-004-2) 지원사업에 의하여 수행되었습니다.

최근 자율주행차량 실용화의 시작 단계로 도심지의 stop-and-go 교통 상황에서 차량 스스로 저속 자동 운전할 수 있는 LSA (Low Speed Automation) 차량 개발에 대한 연구가 진행되고 있다. 이 시스템 구현을 위해서는 근거리 전방차량을 탐지할 수 있으며 신뢰할만한 거리 계산이 가능한 방법이 필요하다. 하지만 기존의 컴퓨터비전을 이용한 방법들은 주로 전방에 소수의 차량이 존재하는 상황에서 원거리 전방차량을 탐지하는 방법에 대해 연구되어 왔다. 따라서 도심지 주행과 같이 전방과 측방에 다수의 차량이 근거리 존재하는 상황에서는 전방차량 종류도 다양하고 차량의 윤곽선도 매우 복잡하여 정확한 차량인식이 어렵고 많은 계산 시간이 필요하다.

본 연구에서는 전방차량인식을 위해 모든 차량의 특징 정보인 한 쌍의 전방차량 뒷바퀴 바깥쪽 윤곽선을 이용하였다. 원근 투영에 의해 카메라에 입력되는 영상내 차량의 특징 정보들은 전방차량의 위치 및 차종에 따라 형태가 일정치 않기 때문에 이용하기 쉽지 않다. 하지만 차량 뒷바퀴 윤곽선은 형태가 일정하고 차종에 관계없이 일반적으로 적용할 수 있는 장점을 갖는다. 주간시 차량 뒷바퀴의 바깥쪽 윤곽선은 수직 선분 형태를 띠고, 도로면과 gray-level에서 분명한 차이를 보이며 차종에 관계없이 바퀴 윤곽선의 아래 부분이 도로면에 위치하는 중요한 특징을 가지고 있다. 이 특징들을 이용하여 근거리 전방차량을 탐지하고 카메라가 장착된 차량에서 전방차량까지의 거리 및 방향을 계산할 수 있는 방법을 제안한다. 기존 연구 중 전방차량의 외곽선이 수직선 형태로 나타나는 특징을 사용한 방법들이 있지만 차량이 가까워짐에 따라 외곽선에 많은 굴곡이 발생하기 때문에 다양한 형태의 근거리 차량을 인식하기에 적당치 않다. 본 연구에서 제안하는 방법은 스테레오비전을 사용할 경우 차량의 후방특징정보를 이용하여 전방차량을 인식하는 다른 방법들에 비해 차종에 따라 다르게 나타나는 차폭 정보를 얻을 수 있는 장점이 있다. 그리고 단일 카메라를 사용하는 경우에도 대부분의 차량 특징 정보들은 차종에 따라 그 위치가 일정치 않으므로 거리 계산에 있어 일반화되기 힘들지만 차량 뒷바퀴의 경우는 두 차량이 동일 경사를 갖는 도로상에 존재한다면 전방차량까지의 거리도 측정할 수 있다. 이 방법은 근거리 전방차량 탐지를 위한 방법이기 때문에 측방 차량을 탐지하기는 어렵다. 그리고 전방 도로의 차선 정보를 안다면 전방차량인식에 도움이 될 수 있지만 논문에서는 차선 정보를 사용하지 않았다.

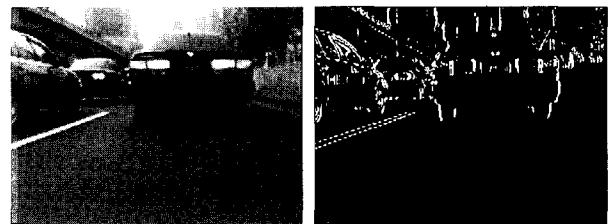
기존 연구에서는 전방차량인식을 위해 카메라를 차량 내 룸미러나 차량 지붕에 설치하는 것이 일반적이었다. 하지만 본 연구에서는 전방차량 뒷바퀴 윤곽선이 탐지하고자 하는 특징 요소이기 때문에 흑백 CCD카메라를 그림 1과 같이 차량 앞범퍼 중앙에 설치하였다. 이와 같이 설치한 이유는 카메라가 도로면과 가까이



그림 1. 카메라가 장착된 실험 차량.
Fig. 1. A vehicle mounted with a camera for imaging front views.

위치할수록 전방차량 뒷바퀴의 가시영역이 넓어지기 때문이다.

그림 1과 같이 장착된 카메라로 입력된 전방차량의 모습은 그림 2(a)와 같다. 차량 뒷바퀴가 명확하게 나타나고, 도로면과 gray-level의 차이가 분명함을 알 수 있다. 이와 같이 카메라를 설치하면 승용차뿐만 아니라 일반 트럭과 버스의 뒷바퀴도 입력 받을 수 있다. 대부분의 기존 연구들의 차량 특징 정보들이 주로 승용차를 대상으로 했던 점과 차이를 보인다. 그림 2(b)는 그림 2(a) 영상에서 수직 에지만을 추출한 영상이다. 차량 뒷바퀴의 바깥쪽 윤곽선이 선분 형태를 띠며 도로면에 대해 수직에 가깝게 나타남을 알 수 있다. 이 특징은 전방차량이 가까이 위치할수록 더욱 크게 나타난다. 따라서 차량이 가까이 위치할수록 탐지하기가 쉬워진다.



(a) 입력 영상 (b) 수직 에지 추출 영상

그림 2. 카메라에서 입력되는 전방 영상.
Fig. 2. A typical front image seen through the camera.

전방차량인식 과정은 전방차량탐지, 전방차량의 거리 및 방향 측정으로 구분되며 인식 절차는 그림 3과 같다. 입력영상에 대해 Sobel 연산자를 사용하여 에지를 구한 후 이들로부터 차량 뒷바퀴 윤곽선의 특징을 나타내는 특징점 추출 및 클러스터링을 수행한다. 여러 클러스터들 중에서 가장 가까이 위치하는 전방차량을 찾아낸 후 이 결과와 카메라, 차량, 도로의 기하학적 위치 관계를 이용하여 전방차량의 거리 및 방향을 측정한다. 모든 처리 과정은 처리 속도를 빠르게 하기 위해서 차량이 존재 가능한 영역에 대해서만 수행된다. 2장에서는 전

방차량을 탐지하는 방법에 대해 설명하고, 3장에서는 전방차량까지의 거리 및 방향을 측정 방법, 4장에서는 주행중인 다양한 전방차량에 대해 실험한 결과를 설명한다.

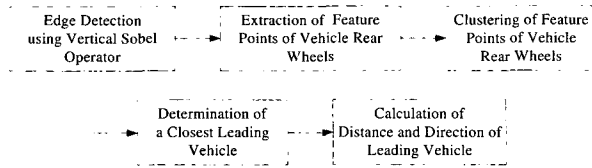


그림 3. 전방차량 인식절차.

Fig. 3. Procedure of a leading vehicle recognition.

II. 전방차량탐지

1. 에지 추출

차량 뒷바퀴의 바깥쪽 윤곽은 강한 수직 에지 성분을 갖고 있기 때문에 이를 찾기 위해서 우선 영상내의 모든 에지를 찾아내야 하고 이를 위해서 에지 추출 연산자로 Sobel 연산자를 사용한다. 뒷바퀴 윤곽의 에지는 수평 성분은 거의 없고 수직 성분만을 갖고 있기 때문에 Sobel 연산자의 마스크들 중에서 수직 마스크만을 사용하여 에지 추출 시간을 단축할 수 있다. 영상내의 x 축과 y 축 좌표가 각각 i, j 인 픽셀의 gray-level 값을 $f(i, j)$ 로 정의할 때, 3x3 크기를 갖는 수직 마스크를 적용한 계산식은 (1)과 같다.

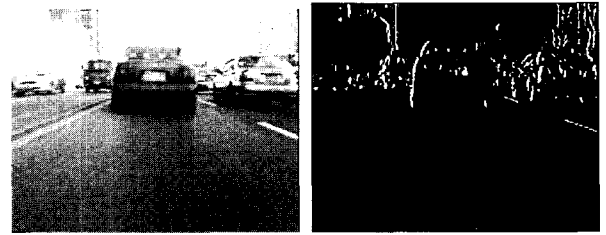
$$\Delta_x f(i, j) = [f(i+1, j-1) + 2f(i+1, j) + f(i+1, j+1)] - [f(i-1, j-1) + 2f(i-1, j) + f(i-1, j+1)] \quad (1)$$

탐색영역의 모든 픽셀에 대해 (1)을 수행한 결과값인 $\Delta_x f(i, j)$ 은 (2)에 의해 상승 에지(positive edge), 하강 에지(negative edge), 에지 없음(no edge)으로 구분된다. (2)에서 k_1, k_2, k_3 는 에지 종류를 구분하기 위한 서로 다른 상수이고, T 는 에지 임계값(edge threshold)이다.

$$g(i, j) = \begin{cases} k_1(\text{positive edge}) & \text{if } \Delta_x f(i, j) \geq T \\ k_2(\text{negative edge}) & \text{if } \Delta_x f(i, j) \leq -T \\ k_3(\text{no edge}) & \text{otherwise} \end{cases} \quad (2)$$

전방차량 뒷바퀴 한 쌍의 윤곽선을 살펴보면 왼쪽 바깥쪽 윤곽선에서는 $\Delta_x f(i, j)$ 이 큰 음수값을 갖는 하강 에지, 오른쪽 바깥쪽 윤곽선에서는 $\Delta_x f(i, j)$ 이 큰 양수값을 갖는 상승 에지가 발생한다. 따라서 한 쌍의 뒷바퀴 윤곽선은 하강 에지로 시작하여 상승 에지로 끝나는 선분들의 집합으로 표현될 수 있다. 뒷바퀴 안쪽 윤곽은 바깥쪽 윤곽과는 반대로 상승 에지로 시작해서 하강 에지로 끝나며 윤곽선이 차량의 위치에 따라 도로면과 명확하게 구분되지 않을 뿐만 아니라 바깥쪽 윤곽선에 비해 작게 나타나기 때문에 사용하지 않는다. 그림 4(a)의 영상에 (1)과 (2)를 수행한 결과가 그림 4(b)와 같이 나타난다. (2)에서 k_1, k_2, k_3 를 각각 128, 255, 0으로 설

정하여 흰색은 하강 에지, 회색은 상승 에지, 검은색은 에지가 없음을 나타낸다.



(a) 입력영상

(b) 에지 추출 영상

그림 4. 두 종류의 에지 추출.

Fig. 4. Extraction of two classes of edges.

2. 뒷바퀴 윤곽선 특징점 추출 및 클러스터링

2.1 특징점 추출

전방차량의 뒷바퀴 바깥쪽 윤곽선은 그림 4(b)와 같이 왼쪽 바퀴 윤곽은 하강 에지로 시작해서 오른쪽 바퀴 윤곽은 상승 에지로 끝나는 선분으로 이루어지는 한 쌍의 에지가 수직으로 일정 개수 이상 나타난다. 하지만 비슷한 특징을 갖는 다른 선분들이 존재하기 때문에 이들로부터 바퀴 윤곽선만을 찾아내는 과정이 필요하다. 우선 바퀴 윤곽선 좌우의 gray-level 값의 분포를 사용하여 적합하지 않은 에지들을 제거한다. 전단계에서 다수의 하강 에지와 상승 에지를 추출했고 같은 y 좌표에 위치하는 한 쌍의 에지로 선분이 구성되지만 많은 에지가 뒷바퀴 윤곽선 조건을 만족하지 않는다. 차량 바퀴의 특성상 바퀴 윤곽선의 안쪽 영역은 타이어 색인 검은색에 가깝게 나타난다. 따라서 추출된 에지가 뒷바퀴 윤곽선이 되기 위해서 하강 에지의 오른쪽 영역과 상승 에지의 왼쪽 영역은 각각의 반대쪽 영역에 비교하여 gray-level 평균값이 작게 나타나야 한다. 픽셀 (i, j) 가 추출된 에지를 이루는 한 점이면 이 점을 기준으로 각각 좌우 n 개 픽셀의 gray-level 평균값 avg_l 와 avg_r 은 (3)과 같고, 상승 에지와 하강 에지에서 이 값들의 비율은 (4)와 같이 일정 비율값과 대소관계를 유지해야 한다.

$$avg_l = \frac{1}{n} \sum_{k=1}^n f(i-k, j) \quad avg_r = \frac{1}{n} \sum_{k=1}^n f(i+k, j) \quad (3)$$

$$\frac{avg_l}{avg_r} < \text{gray-level ratio} \quad \text{if } g(i, j) \text{ is a positive edge.} \\ \frac{avg_r}{avg_l} < \text{gray-level ratio} \quad \text{if } g(i, j) \text{ is a negative edge.} \quad (4)$$

위의 조건을 통과한 한 쌍의 에지는 그림 5의 선분 \overline{AB} 로 나타나며, 이와 같은 모든 선분 $L_i(i=1, \dots, n)$ 를 추출한 후 각 선분의 중심점 x 좌표 x_i , y 좌표 y_i , 선분 길이 w_i 를 구하고, 이 세가지 값을 특징점 $P_i(x_i, y_i, w_i)(i=1, \dots, n)$ 로 정의한다.

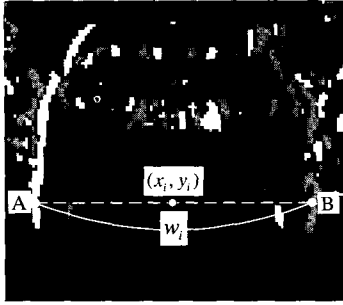


그림 5. 특징점 정의.
Fig. 5. Definition of a feature point.

이렇게 구해진 특징점들 중에서 전방차량의 뒷바퀴 윤곽에 해당하지 않는 많은 점들은 FOE (Focus of Expansion)를 이용하여 제거될 수 있다. 전방도로의 직선 차선을 연장하면 한 점을 향하게 되는데 이 점을 FOE라 하며 전방차량의 뒷바퀴 윤곽선은 FOE를 이루는 양쪽 직선 안에 위치한다. 따라서 임의의 특징점이 전방차량의 뒷바퀴 윤곽선을 나타내는 특징점에 포함되기 위해서는 특징점의 w_i 값은 y_i 값에 따라 일정 범위 내에 포함되어야 한다. 이는 (5)와 같고 함수 $F_{Max}(y)$, $F_{Min}(y)$ 는 임의의 y 좌표에서 특징점의 w_i 값이 가질 수 있는 최소값과 최대값을 의미하며, 카메라의 설치 높이와 기울어진 각도(tilt angle)에 따라 달라진다.

$$F_{Min}(y_i) \leq w_i \leq F_{Max}(y_i) \quad (5)$$

$F_{Max}(y)$ 는 FOE를 이루는 두 직선간의 거리를 정의하는 함수이고, $F_{Min}(y)$ 는 차선보다 급한 기울기를 갖고 FOE를 교차하는 두 직선의 거리로 정의된다. y_i 와 w_i 의 관계에 의해 (5)를 만족하지 않는 특징점들이 제거된다.

2.2 특징점 클러스터링

전 단계를 통과한 특징점 $P_i(i=1, \dots, m)$ 중에서 전방차량의 뒷바퀴 윤곽선에 해당하는 점들을 추출하기 위해 클러스터링을 수행한다. $P_i(x_i, y_i, w_i)$ 의 x_i, w_i 를 (6)에 의해 다운샘플링(downsampling)하여 $P'_i(x'_i, y'_i, w'_i)$ 로 변환한 후 클러스터링이 시작된다. 그림 5를 살펴보면 y_i 값에 대해 서로 다른 크기의 x_i, w_i 를 갖는 여러 개의 특징점들이 존재하며, 차량이 가까이 위치할수록 뒷바퀴 윤곽선 하단 부분이 상단부분에 비해 선분 \overline{AB} 의 크기가 작아짐을 알 수 있다. 따라서, x_i 값을 반으로 줄여 동일한 한 쌍의 뒷바퀴 윤곽선이지만 작은 x_i 값의 차이로 클러스터링 되도록 하고, w_i 값도 반으로 줄여 상단부분과 하단부분의 폭 차이를 반으로 줄여 클러스터링 수행시 동일한 클러스터에 소속되도록 한다. 뒷바퀴 윤곽선의 수직 길이는 영상 크기에 비해 작기 때문에 클러스터 크기 결정에 가장 큰 영향을 미치는 y_i 값은 그대로 사용한다.

$$x'_i = \frac{x_i}{2} \quad y'_i = y_i \quad w'_i = \frac{w_i}{2} \quad (6)$$

특징점들을 클러스터링하기 위하여 Nearest Neighbor Method로 알려진 Single-Linkage 알고리즘을 사용하였다[12]. 이 알고리즘에서는 C_i, C_j 를 각각 i, j 번째 클러스터라고 정의하면 두 클러스터간 거리를 (7)과 같이 두 클러스터에 포함된 점들간의 거리 중에서 가장 짧은 거리로 정의한다. 두 점간의 거리는 (8)을 사용하였고, 이 식은 n 차원상에 위치하는 두 벡터 \mathbf{a} 와 \mathbf{b} 의 각 요소들간의 절대거리합으로 정의되는 City Block Distance에 가중치를 적용한 거리합으로 정의하였다.

$$D(C_i, C_j) = \min_{\mathbf{a} \in C_i, \mathbf{b} \in C_j} d(\mathbf{a}, \mathbf{b}) \quad (7)$$

$$d(\mathbf{a}, \mathbf{b}) = k_1 |a_x - b_x| + k_2 |a_y - b_y| + k_3 |a_w - b_w| \quad (8)$$

(6)에 의해 변환된 특징점의 x', y', w' 값들은 클러스터링을 위한 거리 계산시 같은 기준으로 사용될 수 없다. 뒷바퀴 윤곽선을 나타내는 클러스터내의 특징점들은 뒷바퀴 양윤곽선의 중앙점이므로 x', y' 값을 고려한 2차원에서의 클러스터 모양은 수직선 형태이어야 하고 w' 값의 차이도 작아야한다. 이러한 클러스터를 얻기 위해서 특징점들의 요소들간 절대거리를 계산하는데 x', y', w' 순으로 중요성을 두었다. 이를 거리 계산식에 반영하기 위해 (8)에 사용된 가중치의 크기를 $k_1 > k_2 > k_3$ 와 $k_1 + k_2 + k_3 = 1$ 의 관계에 따라 결정하여 사용하는데 실험에서는 $k_1 = 0.5, k_2 = 0.3, k_3 = 0.2$ 로 설정하여 사용하였다. 클러스터링 시작 단계에서는 모든 특징점을 서로 다른 클러스터로 설정한 후 클러스터간 최대 허용 거리 값보다 작은 클러스터를 병합한다. 더 이상 클러스터링이 발생하지 않을 때까지 같은 과정을 반복한다. 클러스터 크기를 클러스터내에 포함된 특징점의 수로 정의하고, 클러스터링 종료 후 클러스터 크기가 일정값보다 작은 클러스터들은 탈락시킨다.

2.3 전방차량추적

클러스터링 결과 2개 이상의 클러스터가 발생할 때에는 우선 클러스터의 크기가 가장 큰 것을 선택하는데 이것은 가장 가까이 위치하는 전방차량을 탐지하기 위한 것이다. 또한 전방차량의 y 축값의 위치에 따라 클러스터의 크기가 달라지는 조건과 차량 바닥이 도로면 색깔에 비해 어두운 색을 띄는 특징을 이용하여 확인 절차를 거치도록 한다.

전방차량이 차선을 바꾸거나 차간 거리가 상당히 멀어지는 경우를 제외하고는 전방차량의 영상내 위치는 이전에 탐지된 위치로부터 크게 벗어나지 않는다. 따라서 다음 영상 처리 단계에서는 탐색 영역을 축소하여 처리 속도를 증가 시킬 수 있다. 차량 탐지 위치에 따라 차량의 크기가 변하므로 영상 전체에 대해 탐색할 필요가 없다. 따라서 이전에 탐지된 차량의 위치와 폭 정보를 이용하여 추적창(tracking window)를 설정하여 이 윈도우 내부에서만 차량을 탐지하도록 한다. 탐색 영역 결정을 위해 전방차량의 위치과 두 윤곽선의 거리를 도로면에 위치하는 뒷바퀴 윤곽선의 최하단 특징점

의 x, y, w 값으로 정의하고, 차량이 연속적으로 탐지되면 이동평균(moving average)기법으로 (9)와 같이 위치와 윤곽선 거리를 새롭게 결정한다. N_x, N_y, N_w 은 새로운 프레임에서 탐지된 전방차량의 위치 및 폭 정보이고, t 인덱스를 갖는 변수는 이전의 차량 정보에 해당하며 $t+1$ 인덱스를 갖는 변수는 이동평균기법에 의해 새롭게 설정되는 전방차량의 정보이다.

$$\begin{aligned} x(t+1) &= (1-\alpha_1)x(t) + \alpha_1 N_x \\ y(t+1) &= (1-\alpha_2)y(t) + \alpha_2 N_y \\ w(t+1) &= (1-\alpha_3)w(t) + \alpha_3 N_w \end{aligned} \quad (9)$$

where $0 < \alpha_1, \alpha_2, \alpha_3 < 1$

(9)에 사용된 α 값은 차량의 위치에 따라 다른 값을 사용한다. 근거리 차량일수록 차량 위치 변화에 따라 x, y, w 값의 변화가 크고 원거리 차량일수록 변화가 작다. 따라서 근거리 차량일수록 α 값을 크게 하고, 원거리 차량일수록 작게 한다. 탐지된 전방차량의 위치가 이전 위치와 비교하여 일정값을 넘어서면 차량 오인식으로 간주하고 새롭게 탐색을 시작한다. 이러한 과정이 일정 시간이상 반복되면 차량을 탐지할 수 없는 것으로 간주한다. 전방차량이 탐지된 이후에는 다음 프레임에 차량이 나타날 위치를 고려하여 탐색영역을 유동적으로 변경하도록 한다. 추적하던 차량이 탐지되지 않으면 탐색 영역을 전영역으로 확장하여 재탐색 한다.

III. 전방차량 거리 및 방향 측정

컴퓨터비전 시스템을 이용하여 영상내에 위치한 물체까지의 거리를 측정하는데 주로 스테레오비전(stereo vision)이 사용되며, 일반적으로 모노비전(mono vision)을 이용해서는 거리를 측정할 수가 없다. 하지만 거리를 측정하고자 하는 물체가 수평면에 위치하고 카메라 위치, 기울어진 각도(tilt angle), 초점거리(focal length) 등을 정확하게 알면 모노비전으로도 물체까지의 거리 및 방향을 측정할 수 있으므로 이 방법으로 전방차량의 거리 및 방향을 측정한다. 탐지된 두 뒷바퀴 윤곽선의 최하단점들의 중앙점을 전방차량의 후방 중심점으로 정의하고 이 점까지의 거리 및 방향을 구한다. 이 방법은 전방차량과 카메라를 장착한 차량이 같은 경사의 도로상에 위치하고, 차량에 장착된 카메라의 높이는 차량의 진동에 관계없이 항상 일정하다는 가정을 기본으로 한다.

카메라의 기하학적 모델링을 통하여 전방차량까지의 거리와 방향을 측정하기 위해서는 좌표계 설정과 좌표계간 변환과정이 정의되어야 한다. 사용되는 좌표계들은 직교좌표계(Euclidean coordinate system)로 화면좌표계 (X_s, Y_s) , 카메라좌표계 (X_c, Y_c, Z_c) , 월드좌표계 (X_w, Y_w, Z_w) 이다. 화면좌표계(screen coordinate system)는 CCD 카메라에서 입력된 전방도로영상에서 전방차량을 인식하는데 사용되는 좌표계로 도로상의 3차원 좌표점들이 원근 투영에 의해 맵핑된다. 영상화면의 중심점을 원점(O_s)으로 하고 수평축을 X_s , 수직축을

Y_s 로 설정한다. 카메라좌표계(camera coordinate system)는 렌즈 초점(focal point)을 원점(O_c)으로 하고, 카메라의 시축(optical axis)을 Z_c 로 한다. 월드좌표계(world coordinate system)는 차량의 위치를 기준으로 하여 설정되는 좌표계로서 차량의 진행방향을 Y_w 축이라 하고, 높이방향을 Z_w 축, 횡방향을 X_w 축으로 정의하며, $Z_w = 0$ 은 도로면을 의미한다. 그림 6은 설정된 좌표계와 차량간의 위치관계를 나타내고, 카메라는 차량의 앞범퍼 중앙에 전방 정면을 향하도록 설치된다. 카메라좌표계와 월드좌표계의 x 축은 평행하고, 카메라 시축 Z_c 는 월드좌표계의 $Y_w - Z_w$ 평면상에 위치한다. 또한, 카메라좌표계의 원점은 월드좌표계의 Z_w 축상에 위치한다. 전방차량의 후방 중심이 화면상의 좌표 (x_s, y_s) 에 나타날 때, 전방차량의 거리 D 와 방향 β 를 구하고자 한다. 그림 6의 변수들에 대한 정의와 계산 과정은 다음과 같다.

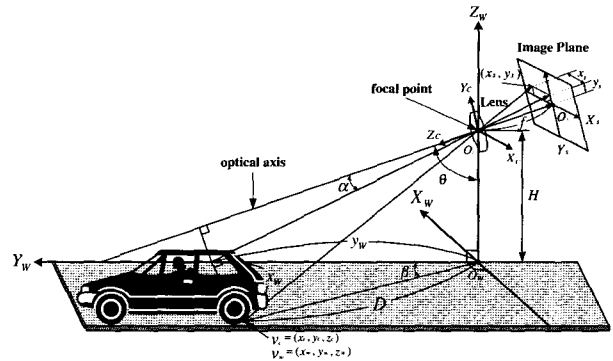


그림 6. 좌표계와 전방차량의 위치 관계.
Fig. 6. Geometric relation between coordinate system and a leading vehicle.

P 는 도로면에 위치한 두 뒷바퀴 바깥쪽 윤곽선의 중심점, f 는 카메라 초점거리(focal length), θ 는 카메라좌표계의 Z_c 와 실세계좌표계의 Z_w 가 이루는 각도, H 는 도로면에서 카메라 초점까지 높이, D 는 전방차량까지의 거리, β 는 전방차량의 방향이다.

도로면의 전방차량 후방 중심점을 월드좌표계에서 $v_w = (x_w, y_w, z_w)$, 카메라좌표계에서 $v_c = (x_c, y_c, z_c)$, 화면좌표계에 원근 투영된 점은 $v_s = (x_s, y_s)$ 로 정의하면, 카메라좌표계와 화면좌표계의 변환 관계는 (10)과 같고, 월드좌표계와 카메라좌표계의 변환 관계는 (11)과 같다. 카메라좌표계는 벡터 t 만큼 평행이동하고, X_c 을 기준으로 $\pi - \theta$ 만큼 시계방향으로 회전하는 R_x 와 Z_c 축을 기준으로 π 만큼 반시계방향으로 회전하는 R_z 의 곱인 행렬 R 에 의해 월드좌표계로 변환된다. 따라서, 전방차량의 후방 중심점까지의 거리 D 와 방향 β 를 구하는데 아래 관계식을 사용한다. 화면좌표계로 나타나는 전방차량의 후방 중심점의 좌표값으로부터 카메라좌표에서의 좌표값을 구한 후, 최종적으로 월드좌표계에서의 좌표값을 구하면 이로부터 거리와 방향을 측정할 수 있다.

$$\begin{pmatrix} x_s \\ y_s \end{pmatrix} = \frac{f}{z_c} \begin{pmatrix} x_c \\ y_c \end{pmatrix} \tag{10}$$

$$\mathbf{v}_w^T = \mathbf{R}\mathbf{v}_c^T + \mathbf{t}^T$$

where \mathbf{R} is a 3×3 rotation matrix and \mathbf{t} is a translation vector.

$$\mathbf{R} = \mathbf{R}_x \mathbf{R}_z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\pi - \theta) & \sin(\pi - \theta) \\ 0 & -\sin(\pi - \theta) & \cos(\pi - \theta) \end{bmatrix} \begin{bmatrix} \cos\pi & -\sin\pi & 0 \\ \sin\pi & \cos\pi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{11}$$

$$= \begin{bmatrix} -1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & \sin\theta & -\cos\theta \end{bmatrix}$$

$$\mathbf{t} = (0, 0, H)$$

앞에서 설명한 알고리즘으로 전방차량의 후방 중심점 \mathbf{v}_s 를 구하고, 초점 거리 f 를 알고 있으므로 (10)에서 z_c 를 알면 x_c, y_c 를 구할 수 있다. z_c 는 (12)에서 그림 6의 α 를 얻은 후 이 값을 사용하여 구해진다.

$$\alpha = \tan^{-1}\left(\frac{y_s}{f}\right) \tag{12}$$

$$z_c = \frac{H}{\cos(\theta - \alpha)}$$

카메라좌표계의 \mathbf{v}_c 좌표값을 구한 후에는 (11)로부터 월드좌표계에서의 전방차량 후방 중심점인 \mathbf{v}_w 좌표값을 구한다. 최종적으로 구하고자 하는 전방차량의 거리 D 와 방향 β 는 각각 (13)와 같다.

$$D = \sqrt{x_w^2 + y_w^2} \tag{13}$$

$$\beta = \tan^{-1}\left(\frac{x_w}{y_w}\right)$$

영상내에 있는 물체나 특정점까지의 거리를 측정하기 위해서는 카메라 캘리브레이션(camera calibration)을 하여야 한다. 카메라 초점으로부터 전방차량의 거리를 계산하기 위해 사용된 그림 6의 카메라 관련 파라미터들은 각각 $\theta = 89^\circ$, $f = 12\text{mm}$, $H = 52\text{cm}$ 로 설정되었다. 수평도로 전방의 미리 정해진 거리에 차량을 위치시키고, 후방 중심점까지의 거리를 측정한 후 모델에 의해 계산된 거리와 일치하는지를 확인하는 방법을 사용하였다.

IV. 실험 결과

실험은 그림 1과 같이 장착된 흑백카메라로 주간 도로영상을 녹화하여 실내에서 전방차량 탐지를 실험하였다. 알고리즘은 Pentium-II350에서 Visual C++로 구현되었고, 영상처리보드는 Matrox사의 Meteor-I, 영상신호는 NTSC 방식의 흑백신호, 입력된 영상의 해상도는 $320 \times 240 \times 8$ bits를 사용하였다. 실험 결과 실시간 처리에 가까운 평균 25frames/sec의 처리 속도를 얻을 수 있었다.

그림 7은 서울시 내부순환도로와 시내도로 주행상황에서 전방에 나타나는 다양한 차량에 대한 전방차량탐지, 전방차량의 거리와 방향을 측정한 실험 결과이다.

인식된 차량은 바퀴 윤곽선을 기준으로 검은색 사각형으로 표시하였고, 하단에 측정된 차량까지의 거리 및 방향을 나타내었으며 거리 단위는 미터(m)이고, 각도 단위는 도($^\circ$)이다. 방향값에서 ‘+’는 전방차량이 전방 우측에 위치하고, ‘-’는 전방 좌측에 위치함을 의미한다.

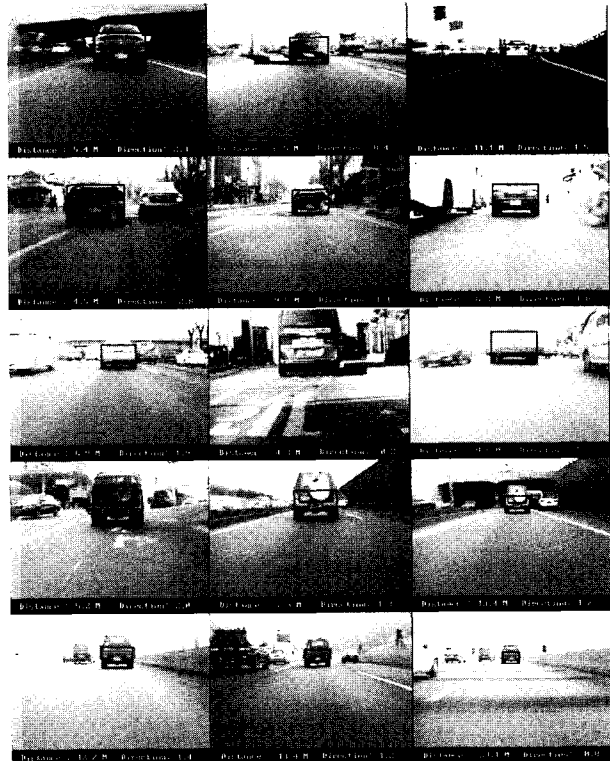


그림 7. 탐지된 전방차량을 직사각형으로 표현한 실험 결과.

Fig. 7. Some examples showing the detection of a leading vehicle by a rectangle shape.

결과에서 볼 수 있는 바와 같이 승용차, 승합차, 트럭, 버스 등 차종 및 차량의 색상에 관계없이 탐지할 수 있었고, 5m이내의 차량부터 25m 근처에 위치하는 차량까지 탐지가 가능했다. 햇빛의 영향이 심하지 않는 구간에서는 안정적으로 탐지할 수 있었고, 옆차선을 지나서 대형 차량의 그림자나 주행 방향 정면에 나타나는 햇빛으로 인해 번짐 현상이나 영상이 희미해지는 상황에서도 탐지가 가능했다. 하지만 터널안이나 카메라가 햇빛을 정면으로 바라보고 주행하는 경우에야간에는 적용하기 어려웠다. 카메라를 차량 앞범퍼에 장착했기 때문에 햇빛 입사각이 낮은 아침이나 저녁에 햇빛을 마주보고 주행하는 경우에는 입력 영상에 하얗게 변하는 현상이 발생한다. 따라서 햇빛을 가리기 위한 하우징을 장착하거나 입사 광량에 따라 카메라 조리개의 자동 조절이 필요하였다.

그림 8은 평평한 도로에서 전방차량을 5m부터 20m까지 1m 간격으로 이동시키면서 전방차량까지의 거리를 레이저 거리계와 본 논문에서 제안한 방법으로 측정

한 결과이다. 사용된 레이저 거리계는 100m 이상의 거리를 $\pm 5\text{cm}$ 이내의 오차로 측정할 수 있기 때문에 레이저 거리계로 측정한 거리를 기준으로 했을 때 제안한 방법의 거리 측정 오차는 평균 0.3219m이었다. 10m 이내에서의 오차는 0.1m보다 작았고 차량의 거리가 멀어짐에 따라 오차가 점점 커졌다. 20m 이상의 거리 측정 결과에서는 1m 이상의 오차가 발생하였기 때문에 제안한 방법으로 20m 이상의 거리에 위치한 전방차량의 거리를 측정하는 것은 적당하지 않았다.

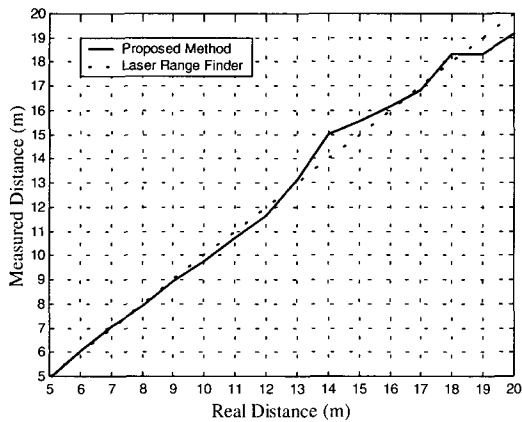


그림 8. 거리 측정 결과.

Fig. 8. Result of distance measurement.

전방차량의 거리 및 방향 측정은 두 차량이 동일 경사 도로면을 주행하고 진동이 없다는 가정하에 이루어진 것이기 때문에 실제 측정값에서는 도로의 굴곡 및 차량의 진동으로 오차가 발생한다. 모노비전의 거리 측정은 픽셀과 실제 거리의 비례 관계에 의해 이루어지기 때문에 원거리를 나타내는 픽셀이 근거리를 나타내는 픽셀에 비해 한 픽셀에 대응되는 실제 거리값이 크기 때문에 원거리 차량일수록 오차가 커지게 된다. 하지만 근거리 전방차량의 대략적인 거리 및 방향을 측정에는 효과적으로 적용될 수 있었고 향후 스테레오비전을 사용하면 좀더 정확한 측정값을 얻을 수 있을 것이다.

실험 결과 본 논문에서 제안한 방법은 도심지 도로에서와 같이 저속 주행을 하면서 전방차량이 10m 전후의 근거리에 위치하는 경우 효과적으로 사용될 수 있었다. 전방차량이 상당히 근거리에 위치하는 경우 차량 뒷모습의 윤곽선이 매우 복잡하여 기존의 에지 특징을 이용하는 방법으로는 차량 인식이 어렵지만 제안한 방법은 차량이 근접할수록 뒷바퀴 윤곽선을 탐지하기가 쉬워지므로 인식하기도 쉬워진다. 그리고 이 방법은 그림 7의 결과에 나타난 바와 같이 기존 방법과 비교하여 다양한 차종을 인식할 수 있는 장점이 있다. 즉, 제안한 방법은 차량의 특징 중 모든 차량이 동일한 형태로 가지고 있는 뒷바퀴를 사용하였기 때문에 전방차량이 승용차, 트럭, 버스 중 어느 것이든 상관없이 인식할 수 있었다. 반면에 단점으로는 제안한 방법은 동일한 차선에 있는 전방 차량만을 탐지하기 위한 방법이기 때문에 옆차선

차량을 인식하기는 어렵다는 점과 기존 방법과는 다르게 차량 뒷바퀴의 윤곽선이 인식 대상이기 때문에 원거리의 차량에서는 뒷바퀴 윤곽선만을 추출하기가 쉽지 않아서 25m 이상의 거리에 위치한 차량을 안정적으로 탐지하기 어렵다는 점이다.

V. 결론

본 논문에서는 단일후백카메라에서 입력되는 전방도 영상에서 차량 뒷바퀴 윤곽선을 이용하여 근거리 전방차량을 인식하는 방법을 설명하였다. 뒷바퀴 윤곽선은 모든 차량이 가지고 있는 특징이므로 이것을 이용하여 차종에 관계없이 인식이 가능했으며 특히 차량 간격이 짧은 도심지 도로에서 10m 이내의 근거리 차량을 탐지하는데 좋은 성능을 보였다. 향후에는 스테레오비전을 사용하여 거리 측정의 정확성을 향상시킬 수 있을 것이고, 뒷바퀴 윤곽선을 탐지한 후 뒷바퀴 상단에 위치하는 붉은색을 띄는 브레이크등도 탐지하여 전방차량의 가감속 상태도 판단할 수 있을 것이다. 본 논문에서 제안한 방법은 전방차량이 근거리에 위치할 때 유용하기 때문에 도심지 주행 상황과 같이 차량간 간격이 좁고 가다서다를 반복하는 교통상황에서 전방차량의 위치를 알려줄 수 있으므로 추돌경보에 적용할 수 있을 것이고, 전자식 거리계와 사용하여 Stop-and-Go 시스템을 구현하는데 적용될 수 있을 것이다.

참고문헌

- [1] H. Satonaka, Y. Hashimoto, Y. Yamada, and T. Kakinami, "A study of sensor fusion technology for collision avoidance system," *Proc. of Intelligent Transport Systems*, pp. 1108-1115, 1995.
- [2] U. Solder and V. Graefe, "Object detection in real time," *SPIE Symposium on Advances in Intelligent Systems*, vol. 1388, pp. 112-119, 1990.
- [3] T. Zielke, M. Brauckmann, and W. Seelen, "Intensity and Edge-Based symmetry detection with an application to Car-Following," *CVGIP: Image Understanding 58*, pp. 177-190, 1993.
- [4] A. Kuehnle, "Symmetry-Based recognition of vehicle rears," *Pattern Recognition Letters 12*, pp. 249-258, 1991.
- [5] J. C. Burie and J. G. Posraire, "Enhancement of the road safety with a stereovision system based on linear cameras," *Proc. of Intelligent Vehicles 96*, pp. 147-152, 1994.
- [6] Bertozzi, Massimo, Broggi, and Alberto, "Real-Time lane and obstacle detection on the GOLD system," *Proc. of Intelligent Vehicles 96*, pp. 213-217, 1996.
- [7] U. Franke and I. Kutzbach, "Fast stereo based object detection for Stop&Go traffic," *Proc. of IEEE Intelligent Vehicles 96*, pp. 339-344, 1996.

- [8] A. Giachetti, M. Campani, R. Sanni, and A. Succi, "The recovery of optical flow for intelligent cruise control," *Proc. of Intelligent Vehicles 95*, pp. 91-96, 1994.
- [9] W. Enkelmann, "Obstacle detection by evaluation of optical flow fields from image sequences," *Proc. of First European Conference on Computer Vision*, pp. 134-138, 1990.
- [10] Betke, Maegrit, Haritaoglu, Esin, Davis, and Larry, "Multiple vehicle detection in hard Real-Time," *Proc. of Intelligent Vehicles*, pp. 351-356, 1996.
- [11] C. C. Lai and W. H. Tsai, "Estimation of moving vehicle locations using wheel shape information in single 2-D lateral vehicle images by 3-D computer vision techniques," *Robotics and Computer-Integrated Manufacturing*, vol. 15, no. 2, pp. 111-120, 1999.
- [12] E. Gose, R. Johnsonbaugh, and S. Jost, *Pattern Recognition and Image Analysis*, Prentice Hall, 1996.



노 광 현

1995년 고려대학교 산업공학과(공학사). 1997년 고려대학교 산업공학과(공학석사). 1997년~현재 고려대학교 대학원 산업공학과 박사과정. 관심분야는 컴퓨터비전, 지능형 자동차.



한 민 홍

1964년 서울대학교 기계공학과(공학사). 1973년 University of Minnesota 산업공학과(공학석사). 1981년~1982년 숭실대학교 산업공학과 조교수. 1985년 Georgia Institute of Technology 산업공학과(공학박사). 1985년~1988년 Texas A&M University 산업공학과 조교수. 1988년~1990년 포항공과대학 산업공학과 부교수. 1991년~현재 고려대학교 산업공학과 정교수. 관심분야는 지능형 자동차, 지능 로봇, ITS, 컴퓨터비전.