

교통 정보 검지기 및 지능형 자동차 개발을 위한 영상 처리 알고리즘

문영수, 정상철, 이준웅, 강동중, 권인소

한국과학기술원

1. 서 론

최근 교통량의 폭발적 증가는 과도한 교통정체를 유발시켜 물류 비용의 증가 뿐만 아니라 국가의 전반적인 대외 경쟁력을 떨어 뜨리고 있다. 또한, 도로상에 주행하는 차량들의 교통사고 빈발에 따라, 운전자에 대한 보다 확실한 안전 확보가 요구되고 있다.

따라서 이런 문제점을 해결하기 위해, 선진 각국에서는 도로상의 교통 상황을 CCD 카메라를 이용, 감지하여 도로상의 교통 시스템을 자동 조정하는 교통 영상 시스템을 개발하고 있다. 또한, 주행차량의 안전도를 확보하기 위하여, 영상을 포함한 다양한 센서를 이용한, 지능형 자동차의 자율주행 시스템을 개발하고 있다.

문헌[1]에 발표된 대표적인 개발사례를 살펴보면 다음과 같다. 유럽 공동체에서는 프로메테우스(PROMETHEUS)라는 공동 프로젝트를 진행하고 있는데, 주로 차량의 자율주행 시스템 개발에 역점을 두고 있다. 미국에서는 IVHS(Intelligent Vehicle Highway Systems)개발을, 일본에서는 ASV(Advanced Safety Vehicle)개발을 위해 산학 연계를 통한 활발한 연구를 수행하고 있다. 도로 상의 교통상황을 감시하여 교통신호를 자동으로 조절하는 자동 교통 영상 시스템은 선진국에서는 이미 개발되어 실제 도로에 적용하고 있다.

국내에서는 초보적인 수준에서 교통 감시시스템을 개발하고 있는 실정이며 차량 자율주행 시스템의 개발은 몇몇 기업에서 시작단계에 불과하다. 따라서, 본 논문에서는 다양한 교통 정보 추출과 지능형 차량의 개발을 위해 본 실험실에서 개발한 영상 알고리즘을 소개 한다.

2. 교통 정보 검출용 영상 알고리즘

도로 영상에서 추출할 수 있는 교통 정보로서는 주행 차량대수, 차량 주행속도, 흐름량, 교통 밀도, 법규 위반차량 적발, 번호판 검지, 차량계적 추적 등이 있다. 이런 정보를 도로 영상에서 보다 쉽게 추출하기 위해서는 도로 영상을 잡는 CCD 카메라를 도로주변의 건물 옥상, 신호등, 또는 전용 고정대에 설치한다.

이렇게 고정된 CCD 카메라로 부터 얻은 도로 영상에서는 움직이는 차량을 제외한 모든 배경 영상이 변하지 않으므로, 도로 영상 속에서 움직이는 차량을 쉽게 검지할 수 있다.

교통 정보를 추출하기 위한 검지창을 도로 영상 전체 영역으로 설정 하느냐, 관심의 국소 영역만으로 설정하느냐는 추출할 교통정보의 종류, 시스템의 하드웨어 성능, 실시간 필요성 등에 따라 결정된다. 보다 단순한 교통 정보 추출(주행 차량대수, 차량 주행속도, 흐름량, 교통 밀도 등)과 실시간 요구시에는 계산량을 줄이기 위하여 도로 영상의 관심 영역을 검지창(detection window)으로 설정하여 그 영역에 대해서만 영상 알고리즘을 적용한다. 보다 복잡한 교통 정보(차종 판별, 법규 위반 차량 적발, 번호판 검지, 차량계적 추적 등)를 얻고자 할 때에는 도로영상 전 영역을 검지창으로 설정한다. 이 경우에는 아주 강력한 계산능력의 하드웨어를 필요로 한다.

영상에서 움직이는 물체영역과 움직이지 않는 배경영역을 분리하기 위한 다양한 방법들이 제안되었는데, 가장 보편적으로는 영상에서 영상 운동량(optic flow)를 계산한 후 유사한 운동량을 가지는 영역을 서로 합쳐, 도로 영상을 운동 물체 영역과 정적 배경 영역으로 분리하는 방법이다[2]. 이렇게 함으로써, 도로 영상에서 움직이는 차량을 검출해 낼 수 있다. 하지만, 영상 운동량을 계산하는 데는 시간도 많이 걸릴 뿐만 아니라 영상 노이즈에 매우 민감하다. 따라서, 차량이 존재하지 않는 도로 배경 영상을 저장해 놓고 영상 차

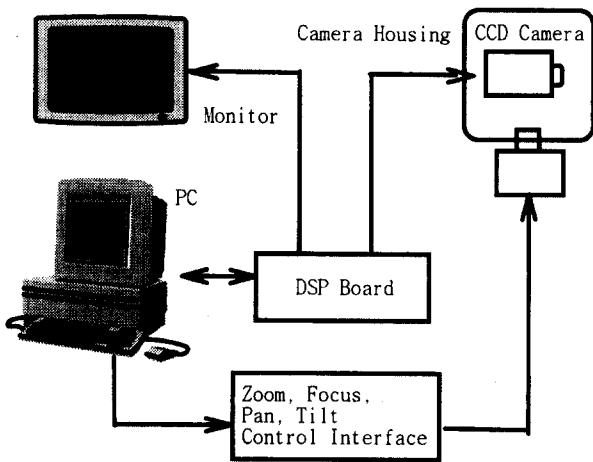


그림 1. 차량 검지용 영상 시스템 구성.

분법(image difference method)을 이용하여 현재 획득한 도로 영상에서 움직이는 차량을 간단히 분리해 낼 수 있다. 이런 알고리즘의 한 예로서, Koller et al.[3]은 고속 도로에서 주행하는 차량을 검지하고 차량의 주행 속도, 차량 궤적 등의 정보를 추출하기 위하여 영상 차분법을 기본적으로 사용하고 있다.

본 실험실에서는 건물 옥상에 고정된 카메라로 부터 실제로 영상을 획득한 후, 실험실의 PC에 설치된 영상 DSP 보드상으로 전송하여 영상처리 알고리즘을 실행시켜 주행 차량을 검지하는 시스템을 개발하였다. 여기에서는 국소 관심영역에 검지창을 설정하는 방식을 채택하여 주어진 하드웨어에 대하여 가능한 한 실시간에 근접하게 작동하도록 하였다. 본 실험실에서 구축한 차량 검지용 영상 시스템은 그림 1과 같다.

차량 검지용 영상 시스템 시스템에서는 팬, 틸트용 모터 구동시스템을 카메라에 부착하여 PC상에서 도로상의 관심 검지창을 자유롭게 설정하도록 하였다. 전체 영상에서 설정된 검지창 영역에 대하여 보다 정확하고 빠르게 차량을 검지하기 위해서는 어떤 영상특징을, 어떻게 이용하느냐에 달려있다. 따라서, 보다 간단하면서도 효율적인 영상 특징치를 얻기 위하여 다양한 실험을 하였다.

2.1 검지창에서의 차량검지 알고리즘

다음 세가지 알고리즘을 제안하고 이를 실제 영상에 적용한 후 결과를 비교하여 최적의 알고리즘을 선택하였다.

2.1.1 농담 정규화 상관법 (Grayscale-Normalized Correlation algorithm : GNC)

농담 정규화 상관법은 산업용 비전 시스템에서 사용하는 방법[4]으로 주어진 영상과 템플레이트 영상(Template image)간의 밝기값의 평균과 분산을 이용해서 두 영상간의 밝기값의 분포의 유사성을 검사하여 주어진 영상에서 템플

레이트 영상을 찾아내는 알고리즘이다.

검지창에 차량이 없을 때의 기준 배경영상(Background Reference Image)을 템플레이트 영상으로 하고 연속해서 들어오는 카메라 영상을 타겟 영상으로 하여 검지창 영역에서의 농담 정규화 상관 계수치(GNC)를 계산하여 차량의 검지 유무를 판단 한다.

$$r = a / (b * c) \quad (1)$$

여기서 a, b, c 각각은 다음과 같다.

$$a = [mn \sum_{i=0}^m \sum_{j=0}^n f_{ij}g_{ij} - \sum_{i=0}^m \sum_{j=0}^n f_{ij} \sum_{i=0}^m \sum_{j=0}^n g_{ij}]$$

$$b = \sqrt{[mn \sum_{i=0}^m \sum_{j=0}^n f_{ij}^2 - (\sum_{i=0}^m \sum_{j=0}^n f_{ij})^2]}$$

$$c = \sqrt{[mn \sum_{i=0}^m \sum_{j=0}^n g_{ij}^2 - (\sum_{i=0}^m \sum_{j=0}^n g_{ij})^2]}$$

r은 농담 정규화 상관 계수치(-1 ~ +1), g_{ij} 는 기준 배경 영상의 밝기값, f_{ij} 는 타겟 영상의 밝기값, mn은 검지창의 크기를 나타낸다. GNC값 r이 이미 정해진 임의의 문턱치보다 클 경우에는 기준 배경 영상과 타겟 영상간의 밝기값의 유사성이 크므로 타겟 영상의 검지창 영역에 차량이 없다고 판정할 수 있다. GNC값 r이 문턱치 보다 작을 경우에는 기준 배경 영상 과 타겟 영상간의 밝기값의 유사성이 작으므로 타겟 영상의 검지창 영역에 차량이 있다고 판정할 수 있다.

2.1.2 반사율의 비 변화 검지법 (Reflectance Ratio Change detection algorithm : RRC)

Nayar 등[5]이 제안한 공간영역(Spatial domain)에서 반사율의 비(Reflectance ratio)라 불리는 광학적 불변량(Photometric invariant)을 이용하여 시간 영역(Temporal domain)에서 반사율의 비(ratio)를 계산하여 차량을 검지하는 알고리즘을 제안한다. 두 영상간의 검지창 영역에서 밝기값의 비(Intensity ratio)는 두 영상의 조명항의 비와 두 Reflectance항의 곱으로서 (2)로 표현할 수 있다.

$$\frac{I_{p1}}{I_{p2}} = \frac{I_{l1}}{I_{l2}} \times \frac{R(N)_{p1}}{R(N)_{p2}} \quad (2)$$

여기서 I_{p1} 와 I_{p2} 는 카메라 영상의 밝기, I_{l1} 와 I_{l2} 는 조명의 밝기, $R(N)_{p1}$ 와 $R(N)_{p2}$ 는 반사율(reflectance), N은 물체 표면의 법선벡터를 나타낸다.

만약 물체가 움직이지 않는다면, 두 반사율은 $R(N)_{p1} = R(N)_{p2}$ 이 되고 두장의 영상간에 조명의 변화가 없다면 다음과 같이 밝기값의 비가 일정하게 된다.

$$\frac{I_{p1}}{I_{p2}} = \frac{I_{l1}}{I_{l2}} \quad (3)$$

따라서, 밝기값 비의 분산값이 거의 0에 가깝게 된다. 타

겟 영상과 기준 배경 영상에 대하여 위의 개념을 적용하면 다음과 같다.

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N \left(\frac{I_{t,i}}{I_{b,i}} - m \right)^2 \cong 0 \quad (4)$$

여기서,

$m = \frac{1}{N} \sum_{i=1}^N \frac{I_{t,i}}{I_{b,i}}$ 은 밝기값의 비의 평균값을 나타낸다.

아래첨자 I_t 는 타겟 영상을, I_b 는 기준 배경 영상을, 아래첨자 i 는 임의의 픽셀을, N 은 검지창의 크기를 나타낸다.

만약, 물체가 움직이면 두 영상에서의 반사율이 달라지므로 $R(N)_{p1} \neq R(N)_{p2}$ 이 되며, 각 픽셀마다 밝기값의 비가 일정치 않게 되어 위의 (3)식이 성립하지 않는다.

$$\frac{I_{p1}}{I_{p2}} \neq \frac{I_{t1}}{I_{t2}} \quad (5)$$

따라서 밝기값의 비의 분산값이 0보다 크게 된다. 타겟 영상과 기준 배경 영상에 적용하여 다음 관계식을 얻을 수 있다.

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N \left(\frac{I_{t,i}}{I_{b,i}} - m \right)^2 > 0 \quad (6)$$

위와 같이 기준 배경 영상과 타겟 영상간의 밝기값의 비의 분산을 계산하여 분산이 문턱치(Threshold)이상이면 차량을 검지했다고 판단할 수 있고, 그 미만이면 차량이 없다고 판단할 수 있다.

2.1.3 경계선 크기의 비 변화 검지법(Edge magnitude Ratio Change detection algorithm : ERC)

인간은 물체 인식 과정에서 밝기값의 변화가 뚜렷한 물체의 경계선을 아주 중요하게 이용한다는 사실에 착안하여, 두 영상의 검지창 내에서 각각 경계선(Edge) 영상을 얻은 다음 검지창 영역에서의 경계선 비의 분산을 계산하여 차량의 존재를 판정한다.

밝기값의 변화가 미소시간에 대해서 변화하지 않고 영상 속도장이 x, y 방향으로 선형이라고 가정하면, (7)을 이용하여 영상의 밝기값의 경계(gradient)는 시간에 대해서 변화하지 않는다는 (8), (9)를 유도할 수 있다[6,7].

$$\frac{dI}{dt} = 0, v_{xx} = v_{xy} = v_{yx} = v_{yy} = 0 \quad (7)$$

$$\frac{d(\nabla I)}{dt} = 0 \quad (8)$$

$$\nabla I(x, y, t) = \nabla I(x, y, t+1) \quad (9)$$

타겟 영상의 검지창에 차량이 없는 경우 두 영상간에 밝기값의 변화가 없어 경계선 크기(Edge magnitude)의 비가

다음처럼 일정하게 된다.

$$\frac{|\nabla I_{t,i}|}{|\nabla I_{b,i}|} = 1 \quad (10)$$

여기에서 $|\Delta I|$ 는 영상의 경계선 크기를, ∇ 는 구배(Gradient) 연산자를 나타낸다. 따라서, 검지창 영역에서의 경계선 크기 비의 분산값이 0에 근접하게 된다.

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N \left(\frac{|\nabla I_{t,i}|}{|\nabla I_{b,i}|} - m \right)^2 \cong 0 \quad (11)$$

$$m = \frac{1}{N} \sum_{i=1}^N \frac{|\nabla I_{t,i}|}{|\nabla I_{b,i}|}$$

윗식에서 N 은 검지창 영역의 픽셀수를, m 은 경계선 크기의 평균값을 나타낸다.

한편, 타겟 영상의 검지창에 차량이 있는 경우 두 영상간에 밝기값의 변화가 있으므로 경계선 크기의 비는 일정하지 않다.

$$\frac{|\nabla I_{t,i}|}{|\nabla I_{b,i}|} \neq 1 \quad (12)$$

따라서, 경계선 크기 비의 분산값은 다음처럼 0보다 크게 된다.

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N \left(\frac{|\nabla I_{t,i}|}{|\nabla I_{b,i}|} - m \right)^2 > 0 \quad (13)$$

위와 같이 기준 배경 영상과 타겟 영상간의 경계선 크기의 비의 분산을 계산하여 기준 Threshold 값보다 크면 차량을 검지했다고 판단할 수 있고, 작으면 차량을 검지하지 않았다고 판단할 수 있다.

2.2 검지 알고리즘의 비교 평가

각 알고리즘을 야간의 영상에 대해서 적용한 결과를 그림 2에 나타냈다. 차량 검지 영역은 검정색으로 움직임이 없는 배경 영역은 흰색으로 표시 하였다. 알고리즘 ERC를 이용한 경우가 가장 정확한 결과를 얻었다. 주간에는 RRC나 ERC 모두 높은 검지율을 보였지만 야간에는 밝기값이 어두워서 차량의 유무에 따른 구별이 어렵다. 따라서 경계선을 이용한 ERC의 경우에는 밝기값의 미분량을 이용하기 때문에 상대적으로 구별이 쉽게 된다. ERC 알고리즘의 적용결과 주간에는 98%, 야간에는 90%의 차량 검지율을 보였다.

3. 모델에 기반한 차량 인식

주어진 도로 영상에서 특정의 관심 차량을 검지하는 것은 그 도로의 교통 현황을 파악하여 적절한 교통 신호 체계를 세우는 데 매우 유효한 정보이다. 이를 위해 우선, 관심 차

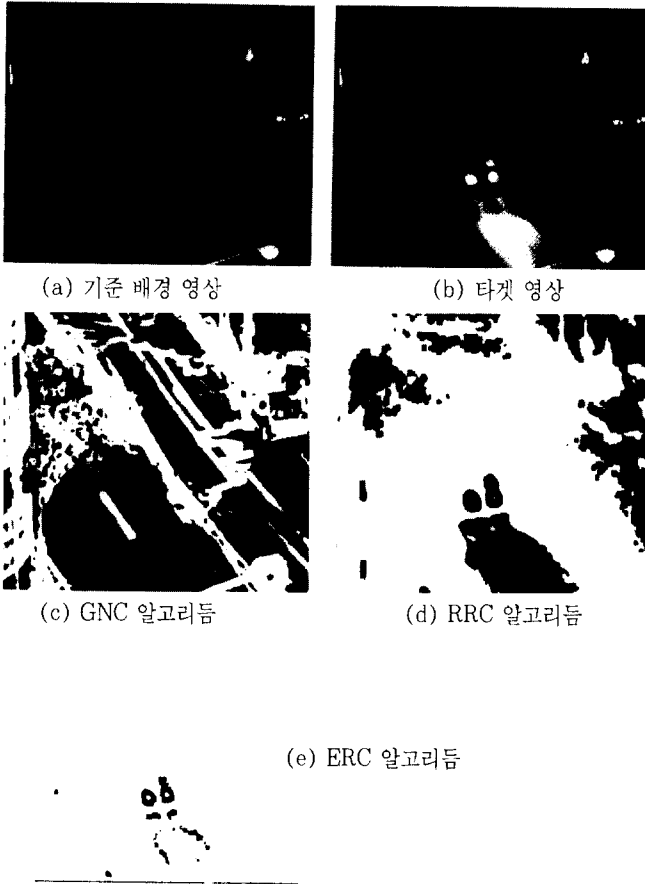


그림 2. 차량 검지 알고리즘의 적용 결과.

량의 외곽 형상에 대해 직선으로 구성된 차량 모델을 구축한다. 그 다음, 고정 카메라의 도로 영상에서 영상 특징을 추출하고 이를 바탕으로 영상에서 관심 모델 차량을 검지하게 된다. Koller[9]는 차량의 3차원 형상을 모델로 구축하고 이를 영상 평면상에 투영시킨 다음, 도로 영상에서 차량의 경계 직선을 뽑아내어 투영된 모델 직선과 매칭과정을 수행하여 모델 차량의 위치와 방향을 결정하였다.

본 연구에서는 매칭과정을 간략화 시키기 위하여 주어진 도로의 기하학적 조건을 바탕으로 2차원의 차량 직선 모델을 구축 하였으며, 매칭의 정확성을 높이기 위하여 새로운 매칭 알고리즘을 제안하였다. 이와같이, 주어진 도로 영상에서 특정 관심차량을 검지하고 추적하기 위한 영상 알고리즘을 살펴보고 실험을 통하여 그 가능성을 살펴 본다.

3.1 차량 경계 직선의 추출(edge line segment extraction)

도로 영상에서 차량의 외곽선에 해당하는 경계 직선(edge segment)을 얼마나 효과적으로 추출할 수 있는냐는 영상처리 분야에서 아주 중요한 기초 분야이다. 따라서, 보다 정확한 경계 직선을 추출하기 위한 알고리즘을 제안 한다[26].

3.1.1 경계의 정규화(Edge normalization)

주어진 영상에 대해, 경계 필터(edge filter)를 적용하여 경계 영상(edge image)을 얻는 데, 이때 Nonmaximum suppression과 Hysteresis thresholding[10]과정을 통하여, Thinning과정을 거치지 않고도, 1 픽셀(pixel)에 가까운 두께를 가진, 경계 크기 영상(edge magnitude image)과 경계 방향 영상(edge orientation image)을 얻을 수 있다. 경계 크기 영상은 이진화 영상(binary image)으로 연결 경계 리스트(connected edge list)를 얻는다. 이 리스트를 이용하면 다음 경계 방향 영상에서 경계 직선을 추출하는 데 있어, 탐색 영역(search space)을 줄일 수 있는 장점을 제공한다. 또한, 경계 방향 영상에서 보다 쉽게 경계 직선을 추출하기 위해, 전체 구배방향(gradient direction)을 미리 정해진 몇 개의 구배 방향 영역(편의상 구배블랍(gradient blob)으로 칭한다)으로 분리한다. 이렇게 하여, 보다 단순화된 경계 구배블랍 영상을 얻게 된다.

하나의 구배 방향만을 가진 영상위에 존재하는 각 픽셀들에 대해 위의 경계 구배 블랍 영상을 구하면 각 픽셀마다 비슷하거나 같은 구배 블랍을 가져야 하는데, 실제로는 랜덤 노이즈한 구배 블랍들이 발생한다. 이는 구배 방향의 블랍화에 따른 중간 구배 방향을 가지는 픽셀에서 주로 발생하며 원래 영상(original image)의 노이즈 픽셀 등에 의해서 발생하기도 한다. 따라서, 이런 문제점을 효과적으로 줄이기 위해서 다음과 같이 경계 정규화(Edge normalization) 과정을 거치게 된다.

앞에서 구한 연결 경계 리스트(connected edge list)상의 임의의 인접한 두개의 경계 픽셀 P_k 와 P_{k+1} 에 대하여, 각 픽셀의 구배블랍인 d_k 와 d_{k+1} 가 서로 다른 지를 체크한다. 만일 서로 다르다면, 앞의 n 개의 픽셀들과 뒤의 n 개의 픽셀들에 대하여 다음과 같이 정의되는 확률밀도 함수 $P(z|m_1)$ 와 $P(z|m_2)$ 를 계산한 후, 두 값의 비로 정의되는 likelihood ratio를 이용하여 (14)에서 처럼 비가 1보다 크면, d_{k+1} 를 d_k 로 바꾸어 주고 1보다 작으면, d_{k+1} 를 바꾸지 않는다.

$$\Lambda(z) = \begin{cases} m_1 & > 1 \\ & < 1 \\ m_2 & < 1 \end{cases} \quad (14)$$

여기서,

$$\begin{aligned} \text{likelihood ratio} &= \Lambda(Z) = \frac{P(z|m_1)}{P(z|m_2)} \\ P(z|m_1) &= \frac{N_1}{N}, \text{ where } N=2n+2, N_1 = \sum_{i=k-n}^{k+n+1} \{d_i = d_k\} \\ P(z|m_2) &= \frac{N_2}{N+1}, \text{ where } N_2 = \sum_{i=k+1}^{k+n+1} \{d_i = d_{k+1}\} \\ m_1 &= \{ \text{gradient label of } p_{k+1} = d_k \} \\ m_2 &= \{ \text{gradient label of } p_{k+1} = d_{k+1} \} \text{이다.} \end{aligned}$$

3.1.2 경계 직선의 추출

일반적으로 경계 필터를 적용하기 전의 밝기 영상에서는 경계선이 연결된 것 처럼 보여도 여러개의 작은 경계 직선으로 조각 나여지는 경우가 많이 발생한다. 따라서, 이런 문제점을 줄이기 위해 다음과 같은 개념을 정의하여 이를 동시에 만족시키는 인접 경계 직선들을 서로 연결 시키게 된다.

정의 1: 동일 선형성(co-linearity): 그림 3에 보이는 임의의 두 경계 직선 L_1 과 L_2 에 대하여, $\max(l_1, l_2) < \tau_1$ 이 성립하면 "colinear" 하다고 한다.

정의 2: 근접성(adjacency): 그림 3에서 두 경계 직선의 서로 가장 가까운 양 끝점 간의 거리 (l_3)가 미리 정해진 값 (τ_2)보다 작으면 두 경계선은 인접해 있다고 한다.

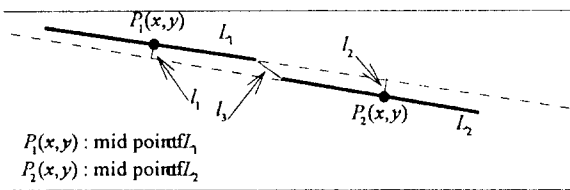


그림 3. 경계 직선의 연결.

3.2 모델에 기초한 표적 인식(Model-based target identification)

앞에서 추출된 경계 직선들(편의상, data segment라 칭함) 사이에서 관심있는 표적의 경계선을 찾아내는 알고리즘을 기술한다. 우선, 관심 표적에 대한 형상 모델(편의상, model segment라 칭함)을 보이는 면의 경계선을 이용하여 구축한다. 따라서 관심 표적은 매칭 과정(Matching process)을 통하여 모델의 경계선(model segment)과 제일 유사한 경계 직선(data segment)을 선택한다.

매칭은 다음과 같이 세 단계로 구성되는데, 첫째, 모델 세그먼트(model segment, M)와 데이터 세그먼트(data segment, D)사이의 유사성을 나타내는 지수로서 마할라노비스 거리(Mahalanobis distance[13])를 계산하며, 둘째, 각각의 모델 세그먼트에 대하여, 미리 정해진 문턱치보다 작은 마할라노비스 거리를 가지는 데이터 세그먼트를 선택하여 매칭 후보군(Matching candidate set : MCS)을 구성하고, 셋째, 모델 세그먼트와 데이터 세그먼트 사이의 기하학적 관계를 이용하여 각 모델 세그먼트에 가장 잘 부합하는 매칭후보군 중의 한 데이터 세그먼트를 선택하게 된다.

3.2.1 매칭 함수(Matching function)

데이터 세그먼트와 모델 세그먼트 간의 매칭은 통계학적 관점에서 수행된다. 여기에서, 데이터 세그먼트 집합과 모델 세그먼트 집합이라는 두개의 집합을 정의하는 데, 다음 (15)로 정의된다.

$$D = \{d_j\} j = 0, \dots, q-1 \quad (15)$$

$$M = \{m_i\} i = 0, \dots, n-1$$

$x = [x_1, x_2, \dots, x_n]^T$ 를 n 차의 가우시안 함수로 표시되는 랜덤 벡터라고 하자. 또한, 각 변수의 평균값과 분산을 μ, σ^2 라고 하자. 랜덤 벡터 x 의 확률 밀도 함수(probability density function)는 다음과 같이 표시된다.

$$f(x) = \frac{1}{(2\pi)^{n/2} \sqrt{|\Sigma|}} \exp\left[-\frac{1}{2}(x-\mu_x)^T \Sigma^{-1} (x-\mu_x)\right] \quad (16)$$

여기서 μ_k 는 평균 벡터(mean vector)를, Σ 는 분산 매트릭스(covariance matrix)를 나타낸다.

$r^2 = (x - \mu_k)^T \Sigma^{-1} (x - \mu_k)$ 은 x 에서 μ_k 까지의 "squared Mahalanobis distance"라 불리며, 윗식에서 r^2 을 최소화 함으로 써 $f(x)$ 를 최대화 할 수 있음을 알 수 있다. r^2 은 카이스퀘어 분포(Chi-square distribution)를 따른다는 사실은 잘 알려져 있다[40]. 다음 식은 에러 벡터(v)를 정의한다.

$$V = \tilde{y}_m - \tilde{y}_d, \quad i \in M \text{ and } j \in D \quad (17)$$

여기에서, y_m, y_d 는 각각 모델 세그먼트, 데이터 세그먼트의 특징치들로 구성된 특징벡터를 나타낸다. 이들은 서로 독립적(independent)이라고 가정하자. 그러면 다음 식과 같이 매칭함수를 정의할 수 있다.

$$r_{ij}^2 = V^T \Sigma_v^{-1} V \quad (18)$$

여기서, $E[v] = 0, \Sigma_v^{-1} = \Sigma_{y_d y_d}^{-1} + \Sigma_{y_m y_m}^{-1}$ 이다.

벡터 v 의 매칭함수 r_{ij}^2 은 χ^2 분포를 가진다.

3.2.2 불확실성 모델(Uncertainty modeling)

밝기 영상에서 경계선을 추출하고 그 경계선으로 부터 여러가지 특징치들을 추출하는데 있어, 영상속에 잠재되어 있는 에러와 경계선을 추출하는 과정에서의 에러 등에 의해서 경계선의 특징치들에 에러가 발생할 수 있다. 이와 같은 에러간의 관계를 규명하는 것을 불확실성 모델링을 한다고 한다. 즉, (18)의 $\Sigma_{y_d y_d}, \Sigma_{y_m y_m}$ 을 계산하는 과정이다. 여기서 사용하는 불확실성 모델링은 Deriche and Faugeras[14]의 방법을 참고하였다.

3.2.3 매칭 후보군의 결정(Construction of matching candidate set(MCS))

불확실성 모델을 기초로 하여, squared Mahalanobis distance, r_{ij}^2 를 계산한다. 계산된 이 문턱치 보다 작으면, 매칭 후보군에 편입시킨다. 이렇게 하여 각 모델 세그먼트에 대한 매칭 후보군(MCS)을 결정한다. 이를 관계식으로 표현하면 다음과 같다.

$$MCS_i = \{j | j \in D, r_{ij}^2 < d_x^2\}, i \in M \quad (19)$$

문턱치 d_x^2 은 100(1- α)%의 신뢰 구간으로 x^2 분포 테이블에서 결정한다[16].

3.2.4 라인 매칭(Line matching)

주어진 모델 세그먼트, m_i 에 매칭되는 데이터 세그먼트를 찾기 위해 기하학적 관계를 이용한다. 모델 세그먼트 m_i 에 대한 최적의 매칭 데이터 세그먼트를 선택하는 절차는 다음과 같다.

(1) 모델 변환(Model transformation)

모델 세그먼트 m_i 와 실제의 데이터 세그먼트 d_k , $k \in MCS_i$ 간의 기하학적 변환에 기초하여 모델 세그먼트를 데이터 세그먼트 영역으로 변환한다.

$$P_1 = \text{Trans}(-x_{si}, -y_{si})P, P = \begin{pmatrix} x_s & x_e & \dots \\ y_s & y_e & \dots \\ 1 & 1 & \dots \end{pmatrix} \quad (20)$$

$$P_2 = \text{Rot}(z, \theta_{ik})P_1,$$

$$P_3 = \text{Trans}(\Delta_x, \Delta_y)P_2$$

윗식에서 P는 모델 세그먼트 m_i 의 시작점, 끝점으로 표시되는 매트릭스를, θ_{ik} 는 그림 4.(a)에 나타난 d_k 와 m_i 사이의 경사 차(orientation difference)를, Δ_x, Δ_y 는 그림 4.(c)에 나타난 m_i 와 d_k 의 변환된 와 각각의 중점 간의 거리차를 나타낸다.

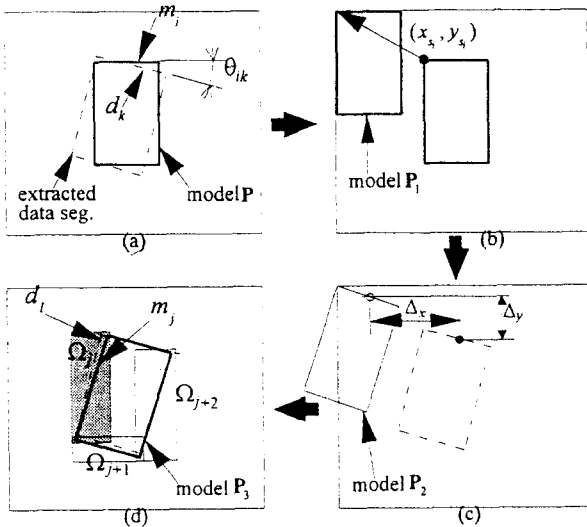


그림 4. 모델 세그먼트의 기하학적 변환.

(3) 매칭도와 최적의 매칭 후보군(Matching quality and best matching candidate set)

변환된 임의의 모델 세그먼트 m_l 에 대하여 데이터 세그먼트 $d_l, l \in MCS_i$ 이 다음 네가지 조건을 만족하는 지를 판정한다.

(a) 존재 조건(Existence condition)

$$\text{Exist}(j, l) = \begin{cases} 1 & \text{if } MBR_{d_l} \cap \Omega_j \neq \emptyset, j \in M \text{ and } j \neq i \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

여기에서 MBR_{d_l} 은 그림 4.(d)에 보이는 d_l 을 에워싸는 최소의 직사각형을 의미한다.

(b) 거리 조건(Distance condition)

$$\text{Dist}(j, l) = \begin{cases} 1 & \text{if } \text{dist}(m_j, p_{int}) \leq Z_{\alpha/2} \sqrt{v_x + v_y} \\ 0 & \text{otherwise} \end{cases} \quad (22)$$

여기서 p_{int} 는 데이터 세그먼트 d_l 와 데이터 세그먼트 탐색영역 Ω_j 와 만나는 교점이며 $\text{dist}(m_j, p_{int})$ 는 그림 5에 보이 듯이 m_j 와 p_{int} 사이의 거리이다. $Z_{\alpha/2}$ 는 100(1- α)%의 신뢰도를 지정하는 노말 분포값이고 V_x, V_y 는 모델 세그먼트의 끝점에서의 분산값을 나타낸다.

(c) 각도 조건(Angle condition)

$$\text{Ang}(j, l) = \begin{cases} 1 & \text{if } |\angle m_j - \angle d_l| < \delta \\ 0 & \text{otherwise} \end{cases} \quad (23)$$

여기에서, δ 은 두 세그먼트의 경사도 유사성을 판정하는 문턱치 이다.

(d) 길이 조건(Length condition)

$$\text{Len}(j, l) = \begin{cases} 1 & \text{if } |\text{len}(m_j) - \text{len}(d_l)| < \epsilon \\ 0 & \text{otherwise} \end{cases} \quad (24)$$

여기에서 $\text{len}(m_j), \text{len}(d_l)$ 는 m_j, d_l 각각의 길이를 나타내고 ϵ 는 두 세그먼트의 길이 유사성을 판정하는 문턱치 이다. m_i 와 d_l 사이의 기하학적 매칭도를 다음과 같이 정의 하자.

$$\text{Geo}(j, l) = \text{Exist}(j, l) + \text{Dist}(j, l) + \text{Ang}(j, l) + \text{Len}(j, l) \quad (25)$$

$\text{Geo}(j, l)$ 값이 4이면 매칭도가 만족되었다고 판단한다. 를 제외한 모든 모델 세그먼트에 대하여 기하학적 매칭도를 계산하고 체크한다. 만일 매칭도를 만족시키는 데이터 세그먼트가 존재한다면 최적 매칭 후보군 R_{ik} 을 다음처럼 형성 한다.

$$\forall i \in M, \exists k \in MCS_i R_{ik} = \{j | A[j], j \in M, j \neq i\}$$

$$\text{where } A[j] = \begin{cases} 1 & \text{if } \geq \alpha(j, l) = 4, l \in MCS_i \\ -1 & \text{elsewhere} \end{cases} \quad (26)$$

(3) 라인 매칭(Line matching)

최적 매칭 후보군의 크기 $\text{mag}(R_{ik})$ 는 매칭도를 만족시키는 데이터 세그먼트를 가지는 모델 세그먼트의 수로 정의 할 수 있다. 우리는 최대의 크기를 가지는 R_{ik} 를 R 이라고 하자. 여러 개의 조합이 발생할 때에는 각 조합의 Mahalano-bis distance를 합하여 최소의 값을 갖는 조합을 최상의 조

함으로 선택한다.

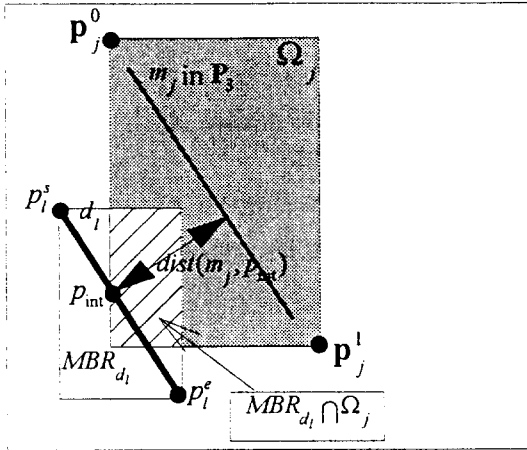
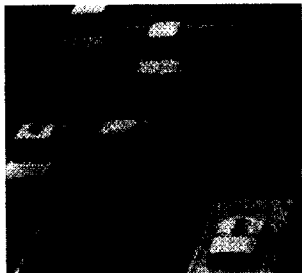


그림 5. 데이터 세그먼트의 매칭 조건.

3.3 실험결과

차량의 운동 모델없이, 그림 6.(a)에 나타난 타원에 둘러싸인 차량을 추적한다. 우선, 마름모 꼴의 차량 지붕을 표시하기 위해서 그림 6.(b)에 나타난 네개의 라인 세그먼트로 간단히 차량모델을 설정하였다. 따라서, 설정된 모델 세그먼트에 해당하는 라인 세그먼트를 찾음으로써 차량 인식을 수행하게 된다. 그림 7은 매칭과정을 거쳐 인식한 모델차량의 지붕을 보여주고 있다.



(a) 카메라 영상



(b) 설정 영상 모델

그림 6. 도로 차량 영상과 차량 모델.

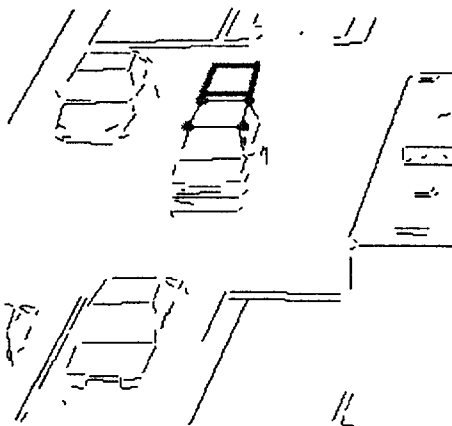


그림 7. 차량 인식 결과.

일단, 한 영상에서 관심차량을 인식한 후에는 다음의 연속 영상에서 관심차량을 추출하는 것은 비교적 쉬우며 상대적으로 시간도 적게 걸린다. 따라서, 관심차량의 연속 추적을 효율적으로 수행할 수 있다.

4. 지능형 자동차를 위한 선행차량 인식

주행하는 자동차에 장착된 카메라의 동적영상을 이용하여, 주행 도로상의 선행차량을 자동으로 검지하고 이의 위치와 형상을 파악하는 것은 차량의 자율주행 시스템을 개발하는데 있어서 아주 중요한 역할을 하며 이와 더불어 운전자의 안전확보 측면에서 많은 기여를 할 수 있다. 도로상의 동적영상에서 도로 배경과 선행차량을 분리하여 검지하고 추적하는 다양한 방법들이 제안되고 있는데 대략 다음과 같이 분류할 수 있다.

- (1) 영상 운동정보(optical flow)를 이용한 배경과 차량의 분리[17].
- (2) 선행차량 뒷면의 영상대칭을 이용한 선행차량 검지[18].
- (3) 스테레오 영상을 이용한 방법[19].
- (4) 템플릿 매칭(template Matching)을 이용한 선행차량 검지[20].
- (5) 신경 회로망(neural network)을 이용한 차량검지[21,22].

이와같이, 다양한 방법들이 존재하지만 비교적 영상 노이즈에 효과적이고 강건한 영상대칭을 이용한 알고리즘을 채택 하였다. 위의 개념을 도로영상에 적용하기 위해서는 다음의 가정들이 필요하다.

I) CCD카메라의 광축은 주행 도로면과 평행하다.

II) 도로와 하늘에 대한 밝기분포가 두장의 연속영상에 대해 충분히 유사한 밝기 분포를 가진다. 사실, 이사항은 도로 상에서는 신호표시, 그림자, 타차선의 차량들에 의한, 하늘에서는 구름, 일몰, 일출등에 의한 영상잡음으로 인해 잘 지켜지지 않지만 그럼에도 불구하고 다른 영역에 비하여 연속된 두장의 영상에 대해 근사한 밝기 분포를 보이고 있다.

III) 관심차량은 카메라 앞에 나타나는 선행차량이다.

IV) 도로와 관심차량 사이의 밝기분포는 다르다.

또한, 다음 두가지 제약조건을 두어 문제범위를 제한하고자 한다.

V) 관심차량의 전체모습이 영상안에 포함되어 있다.

VI) 도로부분도 영상에서 보여야 한다.

동적 카메라 도로영상에서 선행차량의 뒷면 영상이 나타나게 된다. 이때 차량 뒷면의 밝기값 분포는 차량뒷면 가운데의 임의의 수직선을 기준으로 대칭되어 나타난다. 이점을 고려하여 영상속에서 차량뒷면의 밝기값 대칭축 위치와 외곽 형상에 대한 정보를 얻을 수 있다.

4.1 차량 기저선(baseline) 추출

가정 II)에 근거하여 하늘과 도로부분은 연속의 두 영상에서 밝기값 차이가 크지 않으므로 두 영상에 대해 배타적 합연산(exclusive or operation)을 수행한다. 이 결과에 대하여 각 가로줄에 해당하는 영역을 모두 더해 수직 방향으로의 1차 함수 f_j 를 얻는다. f_j 를 두 영역으로 나누는 기준선 α 를 구하기 위하여 평균과 분산을 구한다.

$$\alpha = \mu_e + \lambda_e \sigma_e \quad (27)$$

$$\mu_e = \sum_j f_j / N, \quad \sigma_e^2 = \sum_j (f_j - \mu)^2 / (N - 1)$$

여기서 N 은 영상의 컬럼수이고 λ_e 는 노말(normal) 확률 분포의 상수값이다.

기준선 α 를 이용하여 전 도로영상을 크게 세부분, 즉 도로 영역, 하늘 영역, 관심차량 영역으로 구분할 수 있다. 즉 동적 도로영상을 수직방향으로 세 부분으로 나누는 2개의 기저선을 결정할 수 있다. 그림 8은 추출된 차량 기저선을 보여주고 있다.

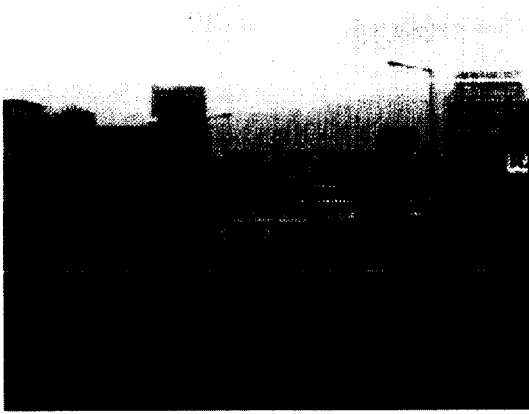


그림 8. 차량의 기저선 추출.

관심차량 영역에서 밝기값 분포가 상대적으로 균일한 부분을 설정할 수 있는데 이 영역을 유효영역(validation region, Ω)이라고 한다. 두장의 영상에서 차량의 움직임이 클 때에는 차량 기저선이 부정확 하게 나오는데 이를 수정하기 위하여 유효영역(Ω)을 이용한다. 유효영역 상에서 밝기값의 평균, 분산을 구하여 앞에서 구한 차량 기저선에 대한 채택 검증과정을 거쳐, 채택 거절인 경우에는 밝기값 분포가 갑자기 크게되는 유효영역 상의 수직위치를 차량 기저선으로 채택한다. 이렇게 하여 선행차량의 대략적인 수직위치를 얻을 수 있다.

4.2 차량의 영상 대칭축(mirror symmetry axis) 추출

도로에서 주행하는 차량뒷면의 영상은 밝기값의 대칭성을 가진다는 사실에 근거하여 선행차량의 영상 대칭축(mirror symmetry axis)을 찾는다. Zielke[18]가 제안한 기법을 기

본적으로 사용하여 선행차량 뒷면의 영상 대칭축을 찾는다. 그렇지만 유효영역에 대한 평균 밝기 분포와 국소 최대값 탐지기법의 이용은 그들의 방법과 다르며 관련 영상특성을 선택하기 위하여 MAP(maximum a posteriori)에 기초한 확률 추론을 이용한다. 이렇게 하여 검지된 영상 대칭축은 선행차량의 수평위치를 나타낸다.

임의의 함수 $f(x)$ 는 우함수(even function)와 기함수(odd function)로 나눌 수 있는데, 우함수는 원점의 세로축에 대하여 대칭성을 갖고 있다. 따라서 한 함수가 임의의 수직축에 대하여 대칭성을 갖고 있으면 상대적으로 우함수가 기함수보다 많은 부분을 차지하고 있다고 볼수 있다. $f(i)$ 는 유효영역의 각 컬럼에서의 밝기값의 합을 나타내는 평균 밝기 분포 함수를 나타낸다. ξ 는 유효영역의 수직폭을, (i, j) 는 픽셀의 좌표를, I 는 밝기값을 나타낸다.

$$f(i) = \frac{\sum_{j \in \Omega} I(i, j)}{\xi} \quad (28)$$

$f(i)$ 의 우함수를 $f_e(x_s + x)$ 로, 기함수를 $f_o(x_s + x)$ 로 나타낸다. 여기서 x_s 와 x 는 영상 대칭성을 계산하기 위한 대칭영역의 대칭축 위치와 대칭축을 기준으로한 수평위치를 각각 나타낸다.

식 (29)는 영상 대칭 영역의 대칭정도를 정량적으로 표시하는 함수이다. S 가 1에 가까우면 대칭인 영상을 가지고 있는 경우이며 -1에 가까운 경우에는 거의 대칭성이 없는 경우이다. 여기서 u 는 대칭영역의 너비를 나타낸다.

$$S(x_s, w) = \frac{\int_{-w/2}^{w/2} f_e(x_s + x)^2 dx - \int_{-w/2}^{w/2} f_o(x_s + x)^2 dx}{\int_{-w/2}^{w/2} f_e(x_s + x)^2 dx + \int_{-w/2}^{w/2} f_o(x_s + x)^2 dx}, -1 \leq S(x_s, w) \leq 1 \quad (29)$$

$$f_e^o(x_s + x) = f_e(x_s + x) - \frac{1}{w} \int_{-w/2}^{w/2} f_e(x_s + x) dx$$

우리가 찾는 대칭영역은 가능한 한 클수록 좋다. 따라서 이런 특성을 고려하여 대칭성의 신뢰도를 표시하는 함수를 다음과 같이 정의한다.

$$S_c(x_s, W, W_{\max}) = \frac{w}{2W_{\max}} (S(x_s, w) + 1),$$

$$W \leq W_{\max}, 0 \leq S_c(x_s, W, W_{\max}) \leq 1 \quad (30)$$

S_c 가 최대가 되는 x_s 는 영상 대칭축의 위치를 w 는 영상 대칭 영역, 즉 차의 유효폭을 나타냄을 알 수 있다.

4.3 차량의 가장자리 선(outmost lines) 추출

전 단계에서 구한 영상 대칭축과 다음에 살펴볼 세가지 종류의 구배정보를 통합하여 차량의 가장자리 선을 추출한다. 이 가장자리 선은 차량뒷면의 넓이를 결정한다. 결국, 시간에 따른 차량뒷면의 넓이 변화는 충돌시간(time-to-contact)을 계산하는데 기초가 된다.

세가지 구매정보는 두 영상의 배타적 합연산후 각 칼럼의 성분을 모두 합한 수직 축적분포의 수평방향 구매, 유효 영역에서의 밝기값 수직 평균 분포의 수평방향 구매, 경계선 요소의 수직 평균 축적분포의 수평방향 구매로 구성 된다. 차량의 가장자리 선에서는 두, 세번째 구매요소가 국소 최대값을 가진다는 사실에 근거하여 세가지 구매 분포에서 확률, 통계기법을 이용하여 가장 공통적으로 국소 최대값이 발생하는 위치를 차량의 가장자리 선으로 결정한다. 이 모든 구매계산영역은 앞에서 언급한 동적 영상의 유효영역 상에서 이루어 진다.

그림 9는 위의 세가지 구매정보를 이용하여 차량의 가장자리 선을 추출한 예를 보이고 있다. 차량의 가장자리 선이 정확하게 검출됨을 알 수 있다.

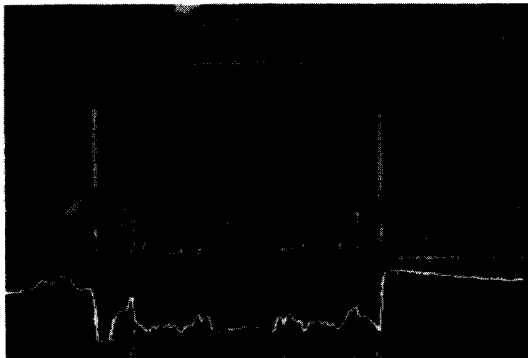


그림 9. 차량의 가장자리 선 추출.

4.4 차량뒷면 유리창의 경계선 검출

차량 표면의 방향을 계산하기 위해서 뒷면 유리창의 경계선을 추출한다. 이를 효율적으로 수행하기 위해, 먼저 차량 상단인 지붕(roof)에 해당하는 후보 수평선을 찾아낸다. 이때 앞에서 설정했던 유효영역과 차량 기저선 그리고 가장자리 선을 이용하여 탐색영역을 설정한다. 이때 앞에서 설정했던 유효영역과 차량 기저선 그리고 가장자리 선을 이용하여 탐색영역을 설정한다.

탐색 영역에서 5군데의 수직선을 지정하고 각각의 수직선을 따라 경계치 분포를 얻은 후 각 경계분포 구매의 국소최대(local maximum) 위치를 찾아낸다. 이렇게 하여 5군데의 수직선에서 공통으로 나타나는 국소최대(local maximum) 위치를 MLD(Maximum Likelihood Decision)기법을 이용하여 차량지붕에 대한 수평 후보선을 찾아 낸다. 다음은 추출한 지붕 후보선 중에서 제일 상단의 선과 유효영역 상단과 두개의 가장자리 선으로 구성하는 유리창 경계선 탐지영역을 설정한다. 이 영역에서 경계치에 의한 선을 검출한다.

이렇게 추출된 선들을 감독 분리(supervised clustering) 기법을 사용하여 미리 설정한 유리창 경계모델의 각 3가지 경계선 그룹으로 분류 한다. 즉 각 3가지 경계선 그룹에 대한 패러메터들의 평균, 분산값을 미리 학습시킨 다음 탐색

영역의 검출된 선 각각의 마할라노비스 거리(mahalanobis distance)를 계산하여 각 그룹으로 분류시키거나 없애 버린다. 최종의 유리창 영역에 대한 경계선을 검출하기 위하여 확률적 추론기법에 의한 선들의 재 분류가 이루어 져야 한다. 이런 과정을 거쳐 최종의 결과를 얻게 된다. 그림 10은 이의 결과를 보여주고 있다. 이런 과정을 연속의 동적 영상에 적용하여 연속의 유리창 경계선을 검출하면 차량의 방향 변화에 대한 정보를 얻을 수 있게 된다.

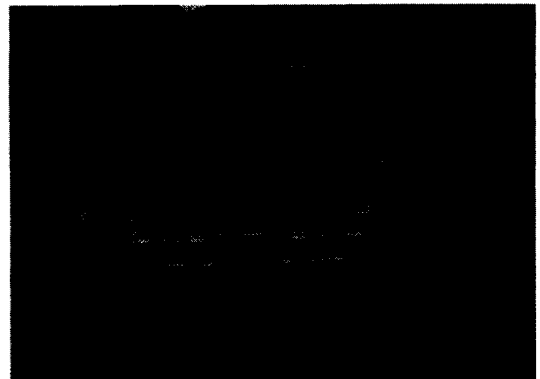
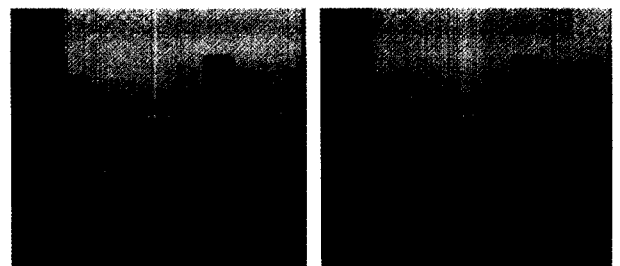


그림 10. 차량 유리창의 경계선 검출.

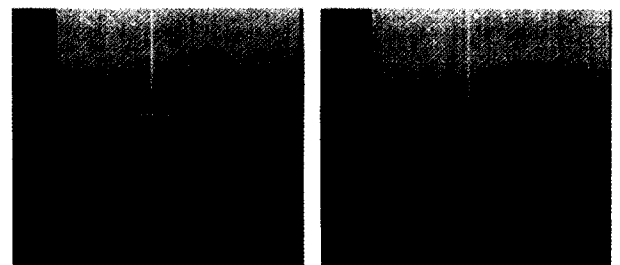
4.5 연속 영상에서의 차량의 추적

그림 11은 연속의 동적영상에서 차량의 영상 대칭축과 차량 외곽선을 추출한 결과를 보여주고 있다. 전 영상에서의 추출결과와 칼만필터(Kalman Filter)를 이용하면 현 영상에서의 영상 대칭 축과 차량 외곽선을 보다 신속하게 찾을 수 있다.



(a) frame 8

(b) frame 9



(c) frame 10

(d) frame 11

그림 11. 차량의 연속추적.

5. 지능형 자동차를 위한 차선 검지

일반적으로 도로에서의 차량 진행방향을 계산하기 위해서는 도로의 차선(lane mark)에 대한 정보가 필요한데, 이러한 차선의 검출은 경계선 검출 연산자(edge gradient operator)에 의존한다. 이러한 경계선 정보로부터 차선을 검출하기 위해, 하프 변환(Hough Transform)에 의한 직선 추출, simulated annealing에 의한 곡선의 검출(fitting)등과 같은 경계선에 기반한 차선 검출 알고리즘[1]이 주로 사용되고 있다.

그러나, 전통적인 경계선 기반 방법으로는 검출을 위한 임계값(threshold value)의 결정이 어렵다. 차선 이외의 아주 복잡한 경계선 존재, 주기적인 표식으로 인한 중간 표식 경계선의 분실, 그림자나 진흙(shadow or skid marks)등, 여러 영상 노이즈 성분들의 존재는 차선의 경계선 검출을 어렵게 만든다. 또한, 차선 추출과 더불어 연속 영상에서의 연속 추적 문제는 또 다른 하나의 문제로 취급된다. 미리 알려진 도로의 기하학적인 제한 조건 등을 이용하여 새로운 입력 영상에서, 차선의 방향과 위치에 대한, 강한 제한 조건을 이용하여 보다 추적을 쉽게 한다[1].

일반적인 환경에서 차선의 검출과 추적은 어려운 문제로 취급되어 왔고 신뢰성 있는 검출 방법에 많은 노력이 집중되어 왔다. 본 차선 검출 방식에서는 능동 직선 모델(Active Line Model)을 전통적인 직선 검출 방법인 하프 변환과 결합하여 이용한다. 경계선 크기에 기반한 하프 변환시 발생하는 임계값 문제를 보다 쉽게 취급하기 위해, 제안된 방법은 기본적으로 노이즈가 존재하는 경계선 영상을 허용한다. 낮은 경계선의 임계값은 도로의 안내선 뿐만 아니라, 그림자, 타이어 마크, 기름 자국 등의 성분도 동시에 검출되는 것을 허용하게 된다.

5.1 능동 직선 모델(Active Line Model)

능동 윤곽 모델은 영상의 경계선(edge)이나 직선, 밝기값 등에 능동적으로 반응하는 곡선으로 구성된 최소 에너지 곡선(Minimum Energy Contour)이다. 이러한 모델은 영상면의 한 위치에서 다른 위치로 움직일 수 있으며, 여러 가지 영상 특징치에 반응하면서 정의된 에너지 모델의 값을 최소화시키는 위치를 찾는다. 능동 윤곽 모델의 변형된 형태를 가지는 능동 직선 모델로 상호 연결된 직선군들로 구성되어 있으며, 이러한 직선들로 정의된 에너지를 최소화 시키면서 영상의 직선을 나타내는 특징치에 근접한다.

먼저 전체 영상 영역을 수직 방향으로 균등하게 분할한 다음, 각각의 분할 영역에서 직선 추출을 위한 하프 변환이 수행된다. 이렇게 검출된 노이즈한 직선 요소들 중에서 도로의 차선을 나타내는 적절한 하나의 연결 직선군들을 찾는 것이 능동 직선 모델(ALM)이다[29].

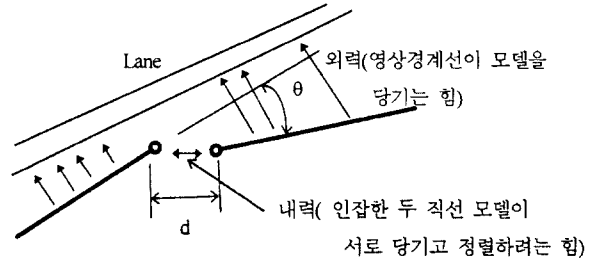


그림 12. 능동 직선 모델

정의된 모델의 에너지는 그림 12에서 볼 수 있는 것처럼, 내력(internal energy)과 외력(external energy)으로 구성되어 있으며 내력은 인접하는 두 직선의 상호 배치 관계에 의해 결정되는 에너지이고, 외력은 주어진 위치에서 영상 특징치와 반응하는 강도의 세기를 나타낸다. 즉, (31)에서 알 수 있듯이 내력은 두 직선의 위치가 가까울수록, 그리고 두 직선이 이루는 사잇각이 작을수록 선호되며 정의된 에너지 모델을 최소화 할 수 있도록 구성된다.

$$E_{int} = \alpha \cdot d(L_i^e, L_{i+1}^s) + \beta \cdot \theta(L_i, L_{i+1})$$

$$\alpha \cdot (L_{i+1}^s - L_i^e)^2 + \beta \cdot \cos^{-1} \left(\frac{(L_i^e - L_i^s) \cdot (L_{i+1}^e - L_{i+1}^s)}{|L_i^e - L_i^s| \cdot |L_{i+1}^e - L_{i+1}^s|} \right)$$

$$E_{ext} = \gamma \cdot \sum_{i=s}^{i=e} |\nabla G(\sigma) * I(i)| \quad (31)$$

5.2 차선 검출

전체 도로 영상을 그림 13과 같이 수직으로 분할한다. 이렇게 분할된 각각의 영역에서 하프변환을 수행하여 차선 후보 직선을 설정한다(그림 13.(a) 참조). 이런 각 분할영역의 차선후보 직선에 대해, 전 영역에서 연결된 정확한 차선을 검출하기 위하여 다음과 같이 정의된 ALM 에너지 함수(32)를 최소화하기 위해, 전형적인 최적화 해법인 Dynamic Programming기법[24]을 이용하여 상태변수(V_1, \dots, V_n)들을 결정한다.

$$E(v_1, v_2, \dots, v_n)$$

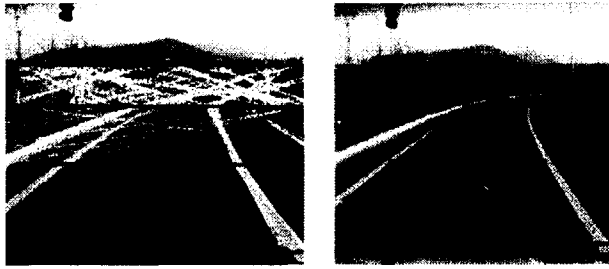
$$= E_1(v_1, v_2) + E_2(v_2, v_3) + \dots$$

$$+ E_{n-1}(v_{n-1}, v_n) \quad (32)$$

여기서, $E_i(\cdot)$ 는 $E_{int,i}(\cdot) + E_{ext,i}(\cdot)$ 이다.

이렇게 하여 검출된 차선은 그림 13.(b)와 같다. 도로 차선이 그림자나 기타 노이즈에 의해 차선에 해당하는 차선 후보 직선이 존재하지 않을 수 있다. 이런 경우에는 앞에서 언급한 알고리즘을 적용하면 각 분할영역의 차선 후보 직선들이 모두 연결되지 않고 어떤 영역에서 갑자기 벌어지는 현상이 발생한다. 따라서 이런영역에서는 에너지 값이 갑자기 증가하므로 실제 차선에 해당하는 차선 후보 직선이 없다고 판정하고 이 영역 직선의 영역에서 정확하게 추출된 차선 후보 직선의 기하학적 정보를 이용하여 현 영역의 차선 후보 직선을 예측한다.

이렇게 하여 일부영역에서 실제의 차선에 해당하는 후보 직선이 없더라도 전체 영역에서의 차선을 정확하게 검출할 수 있다.



(a) 차선 후보 직선 (b) 검출된 차선

그림 13. 하프변환과 ALM에 의한 차선검출.

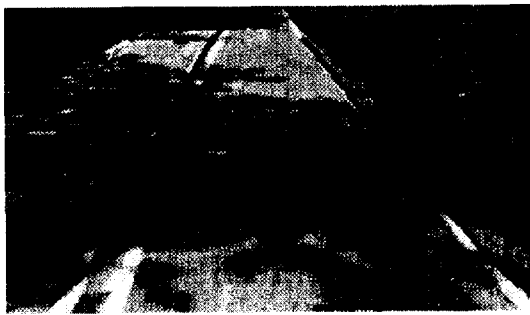


그림 14. 강한 영상 잡음하에서의 차선 검출.

그림 14는 그림자 등의 영상 잡음이 심한 경우에도 추론 기법을 보장하여 차선을 정확하게 추출할 수 있음을 보여주고 있다.

5.3 차선의 추적

연속 영상속에서 연속적으로 차선을 검출하기 위해서는 첫 영상에서 앞에서 언급한 알고리즘을 적용하여 차선을 찾을 수 있다. 다음 영상이 들어올 때에는 앞에서 얻었던 차선 위치를 근거로 역시 각 영역의 차선 후보 직선을 얻는다. 이때에는 하프 변환을 사용하지 않고 다음과 같은 기법에 의해 후보 직선을 얻게 된다.

그림 15에서의 S_0, e_0 은 이전 영상에서 검지된 차선인데, 현 영상에서의 후보 차선으로서 그림 15.(a)에서의 S_0, e_0 과 좌, 우 네 개씩 총 9개의 평행 직선을 선택한다. 또한, 그림 15.(b)에서의 start단 S_0 와 end단의 각 후보점 ($e_{-n/2}, \dots, e_0, \dots, e_{n/2}$)으로 구성된 직선에 대해서 가장 경계 구배치가 큰 직선의 end단 e_p 를 선택한다. 그 다음, e_p 와 start단의 각 후보점 ($S_{-m/2}, \dots, S_0, \dots, S_{m/2}$)으로 구성된 직선에 대해서 가장 경계 구배치가 큰 직선의 start단의 S_p 를 선택한다.

이렇게 하여, 경계구배가 큰 두 개의 차선 후보를 선택할 수 있다. end단의 e_0 를 시작으로 start단의 각 후보점

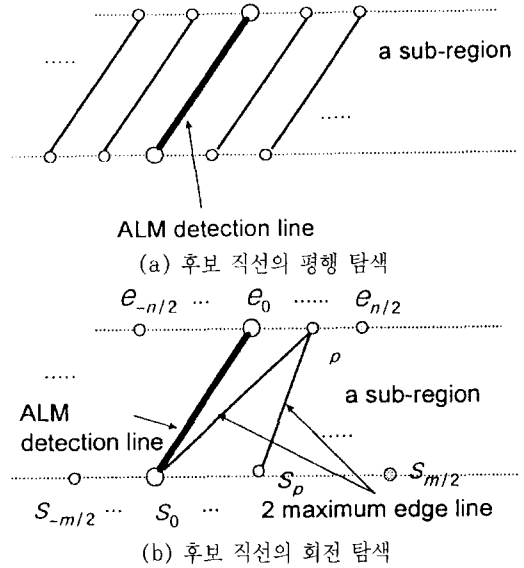


그림 15. 현재 영상에서의 후보 차선 탐색.



(a) frame 14



(b) frame 100



(c) frame 117

그림 16. 연속 추적된 차선.

계 구매치가 큰 S_p 를 선택하는 앞에서의 방법으로, 역시 경제구매가 큰 두 개의 차선 후보를 선택할 수 있다.

결과적으로 현 분할 영역에서 이전 영상의 차선 위치를 기준으로 13개의 후보 차선을 얻게 된다. 주어진 분할영역의 현 영상에서 차선후보 직선을 얻은 다음, 앞에서 설명한 최적화 기법을 이용하여 현 영상에서의 차선을 정확하게 추출할 수 있다. 이를 반복 수행함으로써, 연속 영상에 대해 정확한 차선을 추적할 수 있다. 그림 16은 연속 영상에서 차선을 연속 추적한 결과이다.

6. 결 론

지금까지, 본 실험실에서 개발한, 교통 정보 추출을 위한 차량검지 알고리즘과 차량 추적 알고리즘, 모델에 기반한 차량의 인식과 추적을 위한 알고리즘, 지능형 차량을 위한 선행차량 인식과 차선 인식을 위한 영상 알고리즘에 대하여 살펴 보았고 이들 알고리즘을 직접 도로영상에 적용하여 그 가능성을 살펴 보았다. 교통 정보 추출용 차량 검지 알고리즘은 실험실에서 구성한 영상 검지기 시스템에 실제 적용되어 실시간으로 작동하고 있다. 그렇지만, 아직까지는 대부분의 알고리즘이 실시간 구현과 시스템 적용성에서 미흡한 상태이다. 따라서, 본 실험실에서는 이미 개발된 영상 알고리즘의 신뢰성 제고와 실시간 처리를 위해 알고리즘 자체의 성능 향상 뿐만 아니라 이를 수행하는 하드웨어에 대한 연구도 병행하고 있다.

참고문헌

[1] I. Masaki (ed.), Vision-based Vehicle Guidance : A Parallel Architecture for Curvature-based Road Scene Classification, by Amy Polk & Ramesh Jain, Springer-Verlag New York, 1992.

[2] G. Adiv, Determining three-dimensional motion and structure from optical flow generated by several moving objects, IEEE Trans. PAMI, 7, pp. 384-410, July, 1985.

[3] D. Koller, J. Weber, J. Malik, Robust Multiple Car Tracking with Occlusion Reasoning, 4th ECCV, 1994.

[4] William M. Silver, Alignment & Gauging Using Normalized Correlation Search, Cognex Corporation, Technical Report.

[5] Shree K. Nayar and Ruud M. Bolle, Computing reflectance ratio from an image, Pattern Recognition, vol. 26, no. 10, pp. 1529 - 1542, 1993.

[6] B. K. P. Horn, Robot Vision, MIT Press, McGraw-

Hill Book Company, 1986

[7] Mats Gokstorp et al, A generalized model for optical flow, ICARCV '94, Singapore, November 9-11, 1994.

[8] D. J. Kang, I. S. Kweon, A Visual Tracking Algorithm by Integrating Rigid Model and Snake, Proc. IEEE Intelligent Robotic System (IROS'96), Osaka, Japan, Nov., 1996.

[9] D. Koller, K., Daniilidis and H. H. Nagel, Model-Based Object Tracking in Monocular Sequences of Road Traffic Scenes, International Journal of Computer Vision, vol. 10 no. 3, June, 1993.

[10] O. Faugeras, Three Dimensional Computer Vision-A Geometric Viewpoint, The MIT Press, England. 1993.

[11] E. Horowitz, and S. Sahni, Fundamentals of data structure in pascal, Computer Science Press, U.S.A. 1984.

[12] J. L. Melsa and D. L. Cohn, Decision and Estimation Theory, McGraw-Hill Book Company, U.S.A. 1978.

[13] O. Duda and P. E. Hart, Pattern Classification and Scene Analysis, John Wiley & Sons, U.S.A. 1973.

[14] R. Deriche and O. Faugeras, Tracking line segments, Image and Vision Compt., vol. 8, no. 4, pp. 261-270, 1990.

[15] E. M. Mikhail, Observations and Least Squares, University press of America, U.S.A. 1976.

[16] K. Rohatgi, "An introduction to probability theory and mathematical statistics", Wiley, 1976.

[17] S. M. Smith and J. M. Brady, ASSET-2 : Real time motion segmentation and shape tracking, IEEE Trans. Pattern Anal. Mach. Intell., vol. 17, no. 8, pp. 814-820, 1995.

[18] T. Zielke, M. Brauckmann and W. V. Seelen, Intensity and Edge-Based Symmetry Detection with an Application to Car-Following, CVGIP: Image Understanding, vol. 58, no. 2, pp. 177-190, 1993.

[19] Q. T. Luong, J. Weber, et al., An integrated stereo-based approach to automatic vehicle guidance, ICCV '95, pp. 52-57, 1995.

[20] Yoshiki Ninomiya, et al., A real time vision for intelligent vehicles, In Proceedings of the Intelligent Vehicles Symposium, pp. 101-106, Detroit, 1995.

[21] M. Dougherty, and H. Kirby, The Use of Neural Networks to Recognize and Predict Traffic Congestion, Traffic Engineering and Control, June, 1993.

- [22] C.L. Wan and K.W. Dickinson, Computer Vision and Neural Networks for Traffic Monitoring, IEEE Conference Publication, Institution of Electrical Engineers (U.K.), no. 355, pp. 143, 1993.
- [23] M. Kass, A. Witkin, and D. Terzopoulos, Snakes : Active Contour Models, *IJCV*, pp. 321-331, 1988.
- [24] Amini A.A., Weymouth T.E., and Jain R.C., Using dynamic programming for solving variational problems in vision, *IEEE Trans. Pattern Analysis and Machine Intelligence*. vol. 12, no. 9, pp. 855-867, 1990.
- [25] J. W. Lee and I. S. Kweon Image Point Feature Matching by Triangulation, *Int. J. Intell. Auto. and Soft Comp.*, 1996, To appear.
- [26] I. S. Kweon and J. W. Lee, "Extracting line features in a noisy image", To appear in *Pattern Recognition*, 1997,
- [27] D. H. Ballard and C. M. Brown, *Computer Vision*, Prentice Hall Publishing, 1982.
- [28] 정상철, 영상 시퀀스 처리에 의한 차량의 검지와 차량의 윤곽 검출, 한국과학기술원 자동차 및 설계 공학과 석사 학위논문, 1996.
- [29] D. J. Kang, C. W. Choi, I. S. Kweon, "Finding and tracking road lane using line-snakes", *IEEE Proc. Intelligent Vehicle '96*, Tokyo, Japan, 1996.

저 자 소 개



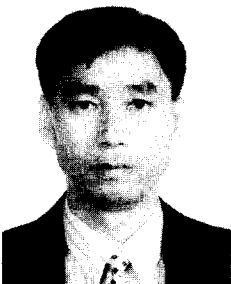
문 영 수

1966년 2월 20일생.
 1988년 서울대 기계공학과 졸업(학사).
 1991년 KAIST 기계공학과 졸업(석사).
 1991-93년 삼성전자 근무.
 1994-현재 KAIST 자동차 및 설계공학과 박사과정.
 관심 분야는 computer vision이다.



이 준 응

1962년 10월 9일생.
 1984년 전남대 산업공학과 졸업(학사).
 1986년 KAIST 산업공학과 졸업(석사).
 1986년~현재 기아자동차 근무.
 1992년~현재 KAIST 자동차 및 설계공학과 박사과정.



권 인 소

1958년 3월 5일생.
 1981년 서울대 기계설계학과 졸업.
 1983년 동 대학원 졸업(석사).
 1990년 카네기멜론대학 로보틱스 졸업(박사).
 1991년~92년 일본 도시바 중앙연구소 연구원.
 1992년~현재 한국과학기술원 부교수.

정 상 철

1970년 3월 15일생.
 1994년 광운대 제어계측 공학과 졸업(학사).
 1996년 KAIST 자동차 및 설계공학과 졸업(석사).
 1996-현재 기아 자동차 근무.



강 동 중

1965년 4월 25일생.
 1988년 부산대 정밀기계과 졸업(학사).
 1990년 KAIST 기계공학과 졸업(석사).
 1990년~93년 현대전자 근무.
 1993년~현재 KAIST 자동차 및 설계공학과 박사과정.