

Haarlike 기반의 고속 차량 검출과 SURF 를 이용한 차량 추적 알고리즘

유재형*, 한영준**, 한헌수**

Fast Vehicle Detection based on Haarlike and Vehicle Tracking using SURF Method

Jaehyoung Yu*, Youngjoon Han**, Hemssoo Hahn **

요약

본 논문에서는 단일 카메라를 이용하여 차량의 위치를 검출하고 연속적인 프레임에서의 차량의 움직임을 추적하는 알고리즘을 제안한다. 차량의 특징을 검출하기 위해 Haar-like 에지 검출기를 사용하고, 카메라의 캘리브레이션 정보를 이용하여 차량의 위치를 추정한다. 신뢰도를 높이기 위해 k 개의 연속적인 프레임에서의 누적된 차량 정보를 추출한다. 최종 검출된 차량을 템플릿으로 지정하고 SURF (Speeded Up Robust Features) 알고리즘을 통해 연속적으로 입력되는 프레임에서 동일한 차량을 추출한다. 이를 통해 동일 차량으로 추출된 차량 정보를 새로운 템플릿으로 업데이트한다. 비교 검출을 위한 수행 시간을 줄이기 위해 이전 프레임에서 검출된 차량의 범위를 확장한 영역만을 관심 영역으로 지정한다. 이 과정은 공통된 대응점을 찾지 못할 때까지 검출과 추적 과정을 반복하여 진행한다. 실험으로 상에서 얻어진 영상에 대해 적용함으로써 제안된 알고리즘의 효율성을 보였다.

▶ Keyword : Haarlike 특징 검출기, 차량 검출, SURF 알고리즘, 차량 추적

Abstract

This paper proposes vehicle detection and tracking algorithm using a CCD camera. The proposed algorithm uses Haar-like wavelet edge detector to detect features of vehicle and estimates vehicle's location using calibration information of an image. After that, extract accumulated

• 제1저자 : 유재형 • 교신저자 : 한헌수

• 투고일 : 2011. 11. 09, 심사일 : 2011. 11. 28, 게재확정일 : 2011. 12. 12

* 송실대학교 전자공학과 (Dept. of Electronic Engineering, Soongsil University)

* 송실대학교 정보통신전자공학부 (Dept. of Information Communication & Electronic Engineering, Soongsil University)

vehicle information in continuous k images to improve reliability. Finally, obtained vehicle region becomes a template image to find same object in the next continuous image using SURF(Speeded Up Robust Features). The template image is updated in the every frame. In order to reduce SURF processing time, ROI(Region of Interesting) region is limited on expended area of detected vehicle location in the previous frame image. This algorithm repeats detection and tracking progress until no corresponding points are found. The experimental result shows efficiency of proposed algorithm using images obtained on the road.

▶ Keyword : Haarlike Feature Extractor, Vehicle Detection, SURF, Vehicle Tracking

1. 서 론

최근 자동차 산업에 있어서 카메라의 활용도가 매우 높아짐에 따라서 안전 주행을 위한 시스템에 대한 연구와 개발이 활발해지고 있다. 특히 차선 인식을 비롯한 전 후방 물체 인식, 주차 보조 시스템 등의 분야에서는 이미 기업에서 상용화를 시작하고 있다. 점차 하드웨어의 성능이 높아지고 연산속도를 위한 최적화 알고리즘이 연구됨에 따라서 실시간으로 처리되는 시스템들이 나타나고 있다. 자동차의 경우 운전자의 순간적인 부주의로 인해 사고가 발생할 경우 재산 및 인명에 대한 피해가 크게 발생하기 쉽다. 따라서 이러한 사고를 미리 예측하여 신속하게 반응할 수 있는 스마트한 안전 시스템의 연구와 개발이 필요하다. [1, 2]

영상을 이용하여 차량을 검출하는 연구가 많은 관심을 받고 있으며 다양한 검출 알고리즘이 제안되고 있다. Sun [3]은 차량의 특징 정보 중에서 가장 두드러지는 수평, 수직 에지를 검출하기 위해 가버필터를 사용하였으며 SVM(Support Vector Machine)을 이용하여 분류 함으로써 차량의 존재 여부를 검증하였다. Wang [4]은 차량의 후면에 대한 특징들을 통해 고유벡터를 얻고 이를 훈련한 결과를 저장한다. 이러한 훈련 결과를 통해 차량과 비차량의 클래스에 대한 확률모델을 생성하고 베이저안 룰을 적용하여 새로 입력되는 이미지가 차량인지 아닌지를 판단하였다. 차량의 특징 분석을 위해서 차량의 후면을 특정 영역으로 나누고 각각의 영역에 대해 PCA(Principal Component Analysis)를 적용함으로써 차량의 여부를 판단하는 알고리즘을 제안하였다. 하지만 이러한 알고리즘은 연산량이 많아 실시간성이 보장되지 않으며 검출결과가 훈련되는 데이터의 형태에 영향을 받기 때문에 다양한 환경에서 입력되는 데이터에 적용할 수 없다는 단점이 있다.

물체의 트래킹의 경우 연속되는 영상에서 동일한 물체의 대응점을 찾는 일이 매우 중요하다고 할 수 있다. 또한 영상

처리에서 대응점을 찾는 일은 필수적인 일의 하나로 카메라 보정, 다수의 카메라를 이용하는 3D 비전, 물체인식과 같은 분야에 활용되고 있다. 물체의 크기나 조명, 시점이 변화된 다양한 영상에서 대응점을 찾는 것은 매우 어려운 일이다. 따라서 다양한 환경에서 지역적인 특징을 통해 대응점을 찾기 위해서는 특징점 추출과 표현자를 생성하는 두 가지 과정을 필요로 한다. 특징점 추출 과정은 영상에서 환경이 변해도 강인하게 찾아질 수 있는 특징의 위치를 찾는 것이다. 이러한 특징점은 대응되는 영상에서도 물리적으로 동일한 곳에 위치하여야 하며, 다른 영상에서 해당 특징점이 찾아질 확률이 높아야 한다.

이전 과정에서 찾은 특징점을 설명할 표현자를 만드는 과정이 두번째 과정이다. 이러한 표현자는 환경 변화에 불변하는 특징을 가져야 하며, 다른 특징점과 구분되어야 한다. 최종적으로 찾아진 특징점의 표현자를 비교하여 대응점을 찾을 때 속도를 향상시키기 위해 최소한의 데이터를 이용하여 표현자를 만들어야 한다.

위에서 설명한 조건을 만족시키며 특징점을 찾고, 표현자를 만드는 방법들이 여러 논문들에서 소개되었다[5~7]. 가장 대표적인 방법으로 Lowe의 스케일불변 특징 변환 SIFT(Scale Invariant Feature Transform) 알고리즘[8]과 Bay의 고속의 강인한 특징 추출 SURF(Speeded Up Robust Features) 알고리즘[9]을 들 수 있다. 두 알고리즘은 흑백 영상으로부터 스케일, 회전 변환에 불변하는 특징점을 찾고, 표현자를 구성하는 공통점을 갖는다.

본 논문에서는 연속적인 영상에서 나타나는 차량의 공통점을 추출하기 위해 SURF 알고리즘을 이용하여 전방에 존재하는 차량을 효과적으로 추적 할 수 있는 알고리즘을 소개한다. 차량을 검출하기 위해서 차량의 외곽 수직 에지 성분을 추출하고 각 성분들에 대한 너비 정보를 이용하여 차량을 검출하고 연속적인 영상에서 SURF 알고리즘을 적용하여 이전에 검출되었던 차량의 위치를 추적하도록 한다. 이는 차량의 검출

과 동시에 동일한 차량여부를 판단함으로써 관심 차량을 추적할 수 있기 때문에 관심 차량의 움직임을 판단할 수 있고 차량의 움직임을 예측하거나 변화에 대한 위험성 판단을 위해 유용하게 사용될 수 있다.

그림 1 은 본 논문에서 제안하는 차량 검출 및 추적 방법에 대한 전체적인 흐름도를 간략화하여 보여준다. 크게 검출 부분과 추적 부분으로 나눌 수 있으며 차량 검출 부분은 매 프레임 새로운 차량의 출현 여부를 판단하여 추적 부분에 전달하고 추적 부분은 한번 검출된 차량의 특징을 연속적인 이미지 프레임에서 추출하여 반복적으로 추적을 하도록 되어 있다.

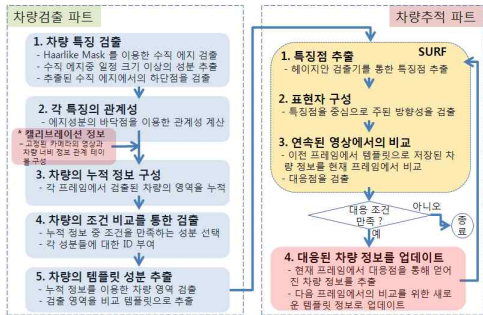


그림 1. 차량 검출 및 추적 알고리즘의 흐름도
Fig. 1. Flowchart for vehicle detection and tracking algorithm

2장에서는 차량의 기본적인 수직 에지 성분의 추출 방법에 대해 설명하고 3장에서는 후보 위치에서의 템플릿을 추출하여 SURF 를 이용한 대칭 성분의 추출 방법을 설명한다. 4장에서는 도로주행을 통해 얻은 연속적인 영상에 적용한 실험결과를 보여주며 5장에서는 결론을 통해 제안한 알고리즘을 정리한다.

II. 차량 위치 검출

1. 에지 기반 차량 위치 추정

차량의 검출 및 검증 단계는 크게 전처리 단계, 후보영역 검출 단계, 차량 검증 단계의 세 개의 영역으로 구성된다. 전처리 단계에서는 수직 및 수평 에지 성분을 검출하고 x 축에 대해서 프로젝션 하여 이중에서 가장 강하게 검출되는 피크 위치를 검출한다. 이것은 차량의 양쪽 외곽 부분을 쉽게 추출할 수 있는 정보를 나타낸다. 검출된 수직 에지 중에서 강하게 나타나는 수직 에지들의 하단 점을 각각 추출한다.

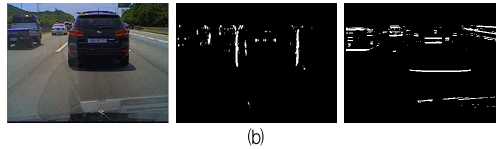
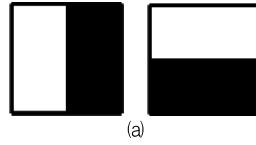


그림 2. 에지 추출을 위한 Haarlike Feature Mask 와 수직, 수평 에지 검출 결과

Fig. 2. Horizontal and vertical edge detection result using Haar-like Feature Mask.

차량의 후면에서 가장 두드러지게 나타나는 특징으로는 외곽선 에지 성분을 들 수 있다. 외곽선 에지 성분은 크게 수직 성분과 수평 성분의 두 가지로 구분할 수 있는데 그 중 수직 성분의 에지 정보는 차량의 기본적인 위치를 고려하기 위해 가장 중요한 성분으로 볼 수 있다. 그림 2 는 차량의 수직 및 수평 에지 성분을 빠르게 검출하기 위해서 Haarlike Feature 마스크를 이용해 주행 영상에서의 차량의 수직, 수평 에지 성분을 검출한 결과를 보여준다. (a) 는 Haarlike Feature 마스크 이고 (b) 는 입력 영상 및 수직, 수평 에지의 검출 결과를 보여준다.

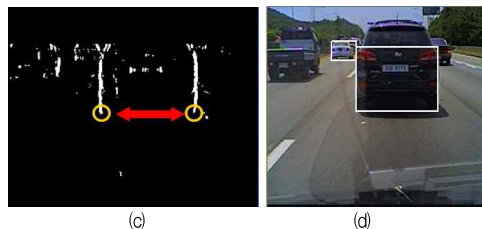


그림 3. 수직에지 추출 결과 및 각 에지의 관계를 이용한 차량 영역의 추출

Fig. 3. Detection result using edge detection result and relation of each edge.

차량 양쪽의 외곽 부분에 해당하는 위치에서 수직 에지 성분이 가장 강하게 나타나기 때문에 이러한 위치들을 관심 대상으로 지정할 수 있다. 수직 에지 영상에서 연속적으로 길게

이어져 있는 에지를 검출하여 크기 순으로 10개를 선정한다. 선정된 수직 에지 성분이 위치하는 하단 점 부분을 검출하고 미리 정의되어 있는 차량 너비 추정 테이블을 이용하여 해당 영역에서의 차량을 검출한다. 이때 추정 테이블은 영상에서의 위치에 대한 차량의 너비 정보에 대한 관계식으로 표현될 수 있다. 그림 3은 수직 에지 성분을 이용한 차량 검출 단계를 보여준다. (a)는 입력영상이고 (b)는 수직 에지 (c)는 수직 에지 성분에 대한 거리 비교이며 (d)는 최종 검출 결과를 보여준다.

이러한 차량 검출이 가능한 이유로, 첫 번째 조건은 차량이 평평한 도로를 주행하고 카메라는 특정한 각도로 고정되어 있기 때문에 영상의 특정 점이 도로의 한 지점과 일대일 매칭이 될 수 있다는 것이다. 두 번째 조건은 매칭이 되어 있는 지점에서의 차량의 크기를 미리 추정할 수 있으며 이 정보를 특정 테이블 형태로 구성해서 사용할 수 있다는 것이다.

2. 연속적인 영상의 누적 정보를 이용한 차량의 위치 검출

도로의 환경, 조명의 변화에 따라서 순간적으로 차량의 특징을 잃게 될 경우 차량의 올바른 위치를 검출하지 못할 수 있다. 다만 이러한 경우 순간적으로 발생하는 경우가 많기 때문에 이전 프레임에서 검출된 차량의 정보를 이용하여 현재 프레임에서의 차량의 위치를 추정할 수 있다.

이를 위해서 이전 20개의 프레임에서 검출된 차량의 위치 및 크기 정보를 누적시키고 해당 영역에서 일정한 프레임 이상 검출된 위치를 검사하여 일정 횟수 이상으로 출현한 것을 유효한 차량의 정보로 인정한다.

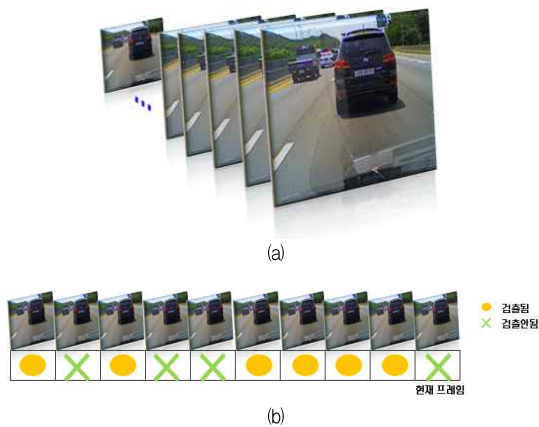


그림 4. 연속된 프레임에 대한 차량 정보의 처리 예시
Fig. 4. Example of vehicle information in continuous frames

그림 4는 연속적인 프레임에서의 차량 정보에 대한 예시를 보여준다. 그림 4(a)는 연속적으로 들어오는 입력 영상에 대한 차량 검출의 예시를 보여주며, 그림 4(b)는 각 프레임에서 검출된 차량의 정보를 저장한 결과 예시를 보여준다. 이러한 정보를 통해 20 프레임에서 검출된 정보와 미 검출된 정보를 저장하여 특정 프레임 이상 존재하는 것을 최종 검출된 결과로 지정한다. 또한 기억되어 있는 차량의 정보를 그림 5와 같이 누적하여 차량의 위치 정보를 쉽게 찾을 수 있도록 한다.

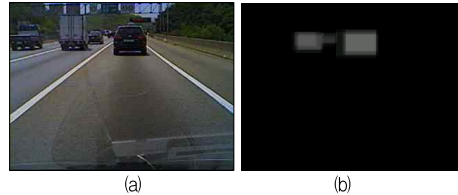


그림 5. 차량의 누적된 정보의 결과
Fig. 5. Accumulated result image of detected vehicle information

그림 5는 누적된 차량 정보의 결과를 보여주는 것으로 각 프레임에 대해 일정 가중치를 주고 각 프레임에서 검출된 차량의 위치에 해당하는 값을 누적시킨 값을 이미지로 표현한 것이다. 그림 5(a)의 경우는 현재 프레임에 대한 입력 영상을 보여주는 것으로 의미상 계속적으로 입력되는 영상을 의미한다. 그림 5(b)의 경우는 20개의 프레임에서 검출된 차량의 영역 정보를 누적한 결과를 보여주는 것으로 차량의 검출 횟수에 대해서 밝기 성분을 누적하여 보여주는 결과를 나타낸다. 이러한 결과를 이용하여 순간적인 오검출 및 미검출에 대한 예외 처리를 할 수 있다.

III. SURF 기법을 이용한 추적

1. SURF 알고리즘

대응점을 찾기 위한 절차를 특징 점 추출과 표현자 구성의 2가지 단계로 구분할 수 있다. 첫 번째로 흑백 영상을 이용하여 특징 점을 찾기 위한 대표적인 방법들로 헤리스 코너 검출기[10], 헤이시안 검출기[11], 헤리스-라플라시안 검출기[12], 가우시안 차분 검출기[8], 고속 헤이시안 검출기[9]등을 들 수 있다.

Harris가 제안한 헤리스 코너 검출기는 가장 널리 알려진 특징추출 방법 중 하나로 코너 점이 양방향으로 곡률이 높다는 점에 착안하여 고유 값과 코너응답함수를 이용하여 회전에

불변하는 특징을 찾는다. 헤이시안 검출기는 얼룩 특징과 나선 특징 검출에 강한 헤이시안 행렬식을 이용하여 특징 점을 추출한다. 그리고 가우시안 2차 미분검출기(Laplacian of Gaussian)는 기존의 헤리스 코너 검출기가 스케일에 따라 특징 점이 변하는 특성을 해결하기 위해 가우시안2차 미분을 통해 스케일공간을 구성하고 코너 점을 찾는다.

가우시안 차분 검출기(Difference of Gaussian)는 LoG 를 근사화하여 속도를 향상시킨 방법으로 스케일 공간에서 이웃하는 픽셀 중 극 값만 검출한다. 그리고 에지 특성과 명암 대비도를 고려하여 불필요한 성분을 제거함으로써 특징 점을 검출한다. 마지막으로 Bay가 제안한 고속 헤이시안 검출기는 적분영상[13]과 가우시안 2차 미분을 근사화한 사각필터를 이용하여 수행시간을 획기적으로 단축하였다. 두 번째 단계로 흑백 영상의 정보를 이용하여 표현자를 구성하는 알고리즘에는 SIFT, GLOH[7], SURF를 들 수 있다.

Lowe의 SIFT 표현자는 특징 점을 중심으로 4×4의 세부영역을 구성하고, 각 세부영역에서 픽셀들의 기울기 방향 히스토그램을 8개의 방향으로 구분하여 총 128차원의 벡터를 구성한다. Mikolajczyk가 제안한 기울기 위치 방향 히스토그램은 SIFT를 확장한 알고리즘의 하나로 찾아진 특징 점을 중심으로 로그폴라 형태의 3개의 반경과 8개의 방향에 대해 총 17개의 세부영역을 나누고, 각 세부영역에서 16방향의 기울기 히스토그램을 통하여 총 272개의 표현자 벡터를 얻는다. 그리고 얻어진 고차원의 표현자 벡터를 주성분 분석 방법으로 128차원 표현자 벡터로 재구성한다. 마지막으로 SURF 표현자는 특징 점을 중심으로 4×4의 세부영역을 구성하고 각 세부영역에서 Haar 웨이블릿을 사용하여 2, 4, 8개의 특징을 구함에 따라 각각 32, 64, 128차원의 표현자 벡터를 구성한다.

2 특징점 추출

영상에서 모든 화소를 구분하는 표현자를 구성하거나 모든 화소의 표현자를 비교하여 대응점을 찾는 작업은 쉽지 않다. 따라서 환경 변화에 강인한 특징 점을 찾고, 다른 특징 점과 구분할 수 있는 표현자가 필요하다. 제안하는 확장 SURF 알고리즘에서 특징 점은 고속 헤이시안 검출기[9]를 사용하여 추출한다. 고속 헤이시안 검출기는 적분영상과 근사화된 헤이시안 검출기[11]에 기반을 둔다.

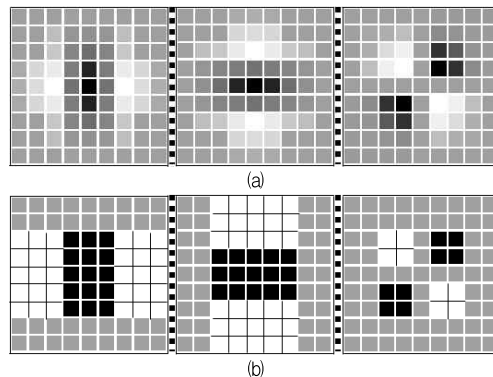


그림 6. 특징 점 검출을 위한 근사화된 헤이시안 검출기
Fig. 6. Approximated Hessian detector to extract feature points

헤이시안 검출기는 식 (1)에 정의된 헤이시안 행렬식에 기반을 둔 특징 점 추출 알고리즘으로 속도와 정확도면에서 좋은 성능을 보인다.

$$H(x, y, \sigma) = \begin{bmatrix} LI_{xx}(x, y, \sigma) & LI_{xy}(x, y, \sigma) \\ LI_{xy}(x, y, \sigma) & LI_{yy}(x, y, \sigma) \end{bmatrix} \quad (1)$$

식 (1)에서 $LI_{xx}(x, y, \sigma)$ 는 x, y 위치의 입력영상과 의 분산을 갖는 가우시안의 x방향 2차 미분 값 $\frac{\partial^2}{\partial x^2} g(\sigma)$ 과의 컨벌루션 값을 의미하고, 나머지 $LI_{xy}(x, y, \sigma)$ 와 $LI_{yy}(x, y, \sigma)$ 도 x y방향으로의 미분 그리고 y방향으로의 2차 미분된 가우시안 필터와 컨벌루션 값을 의미 한다. 근사화된 헤이시안 검출기는 헤이시안 행렬식을 사용하는 대신 그림 6과 같이 표현된 사각필터를 이용한 근사화된 헤이시안 행렬식을 사용하는 방법이다. 참고로, 그림 6(a) 의 경우는 x, y, xy 방향에 대한 가우시안 2차 미분 필터를 보여주는 것이며, 그림 6(b) 의 경우는 x,y, xy 방향의 근사화된 사각필터를 보여준다. 그리고 스케일에 불변하는 특징을 얻기 위해 스케일링된 영상을 사용하지 않고 사각필터의 크기를 변화시켜 특징을 추출한다. 결과적으로 사각필터의 컨벌루션을 계산하는데 앞서 얻어진 적분영상을 이용함으로써 사각 영역의 크기에 상관없이 빠르게 헤이시안 행렬식을 구성하고 특징 점을 찾아낸다.

3. 표현자 구성

표현자를 구성하기에 앞서 첫 단계로 회전에 불변하는 특성을 갖게 하기 위해 찾아진 특징점에 대해 방향 정규화를 수행한다. 방향 정규화에는 주방향을 찾기 위해 특징점을 중심으로 특징점이 찾아진 스케일공간의 스케일정보 's'를 참조하여 6s반경 내의 픽셀들에 대하여 가로방향 Haar 웨이블릿 특징(d_x)과, 세로방향 Haar 웨이블릿 특징(d_y)을 계산한다. 그리고 찾아진 d_x 와 d_y 를 더하면 특정 방향을 가리키는 합벡터를 구할 수 있다. 특징점의 주방향은 합벡터가 가장 많이 가리키는 방향으로 정해지며, 이는 그림 7에서 보이는 것과 같이 60°범위내의 포함되는 합벡터의 수를 더하고 가리키는 벡터의 총 수를 비교하게 된다.

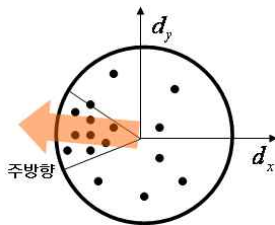


그림 7. 주된 방향성 검출
Fig. 7. Main orientation detection

지역적 특성을 표현할 특징벡터를 계산하기 위해 Haar 웨이블릿 특징의 조합을 사용한다. 특징벡터를 계산하기에 앞서 회전 불변 특성과 스케일 불변 특성을 고려하기 위해 특징점의 주방향과 스케일 정보 's'를 이용하여 특징점을 중심으로 가로, 세로 20s 범위를 참조하고, 20×20(pixels)의 정규화된 부분영상을 구성한다. 회전과 스케일에 정규화된 부분영상이 구성되면 d_x, d_y 에 해당되는 가로, 세로 방향의 Haar 웨이블릿을 계산한다. d_x, d_y 가 계산되면 부분영상을 4×4의 세부영역으로 나누고 각 영역에서 $V(\sum d_x, \sum d_y, |\sum d_x|, |\sum d_y|)$ 의 계산을 통해 4개의 특징벡터를 각각 만들어낸다. 여기서 지역적 특성을 갖는 특징벡터 4개를 사용하게 된 이유는 SURF 알고리즘을 2, 4, 8개의 특징벡터를 사용하여 대응점을 찾을 때 얻은 결과의 분석을 통하여 얻어졌다. 4개의 Haar 웨이블릿 특징 조합으로 구성된 특징벡터를 사용했을 때와 8개를 사용하였을 때 정확도 측면에서는 큰 차이를 보이지 않았고, 낮은 차원의 표현자를 구성하여 정합 시 비교속도를 높일 수 있기 때문이다.

4. SURF 기법 기반의 차량 추적

앞에서 설명한 것과 같이 SURF 알고리즘은 다양한 기하학적 변환과 광학적인 변환에 대해 강건한 특징점을 추출하고 기술자를 생성해서 정합을 하는 방식으로 특징 영상 내에 존재하는 물체를 검출한다. 차량의 추적에 SURF를 적용하기 위해 이전 프레임에서 검출된 차량의 이미지를 비교 템플릿으로 저장하고 현재 프레임의 영상 내에서 템플릿과 정합되는 위치를 검출함으로써 동일한 차량을 추적한다. 우선 Fast-Hessian 검출기를 이용하여 차량의 템플릿 이미지와 현재의 검색 프레임 이미지의 특징 점들을 구하고 SURF 서술자를 이용하여 관심 포인트에서의 방향과 크기를 구한다. 그림 8은 차량의 템플릿과 현재 검색 프레임에 대해 특징점을 구하고 각점에서의 방향과 크기를 구한 결과를 보여준다.

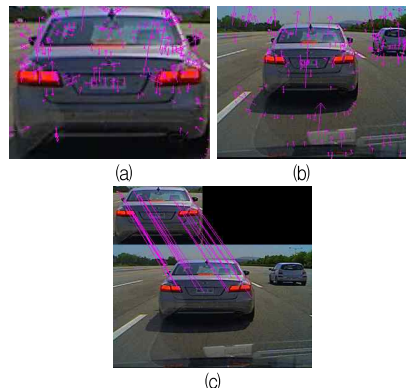


그림 8. 차량 템플릿에 대한 특징점 추출 결과 및 대응점 검출 예시
Fig. 8. Feature extraction result for vehicle template image and corresponding points

특정 점 및 서술자를 통해 얻어진 결과를 이용하여 현재 프레임에 존재하는 이전 차량 정보를 검출한다. 그림 8은 이러한 처리에 대한 결과를 보여준다. 이렇게 찾아진 결과를 이용하여 현재 프레임에서의 새로운 차량의 위치를 분석하는데 이때 정확한 차량의 위치를 검출하기 위해 매칭된 차량의 영역을 새로운 검출 차량으로 인식한다. 그림 8(a)는 이전영상에서 추출된 차량의 템플릿에서의 특징점 추출 결과를 보여주며, 그림 8(b)는 현재 프레임에서의 특징점 추출 결과를 보여준다. 그림 8(c)의 경우는 (a)의 차량 템플릿과 (b) 현재 프레임에 대한 매칭 결과를 보여준다.

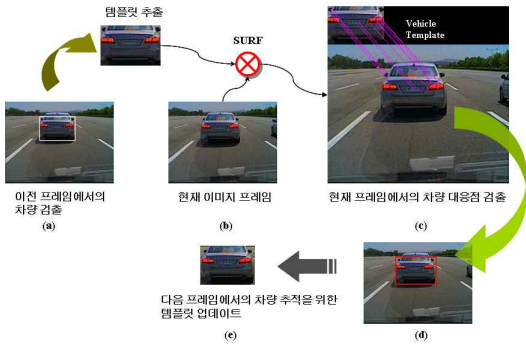


그림 9. 차량 추적을 위한 과정
Fig. 9. The vehicle tracking progress

그림 9 는 이전 프레임에서 검출된 차량의 정보를 이용하여 SURF 알고리즘을 적용해 현재 프레임에서 동일한 차량의 위치를 검출하는 단계를 순서적으로 보여준다. 그림 9(a)는 이전 프레임에서의 차량 검출 결과를 비교를 위한 템플릿 정보로 저장하는 과정을 보여주고 그림 9(b)는 현재 프레임에서 이전에 추출된 차량 템플릿과의 매칭 과정을 보여준다. 그림 9(c)는 SURF를 이용한 매칭 결과를 보여주며 이렇게 최종적으로 검출된 차량의 정보는 그림 9(d)와 같이 다음 프레임에서 비교 할 수 있도록 차량 템플릿으로 갱신된다. 이렇게 갱신된 차량의 템플릿은 다음 프레임에서 반복적으로 차량의 위치를 검출하기 위해 사용된다. 추가적으로 동일 차량을 빠르게 비교하기 위해서 이전에 검출된 차량의 위치정보에 대해서 일정한 비율로 확대한 영역을 관심영역으로 두어 실시간 처리가 가능하도록 한다. 검출된 동일 차량의 정보를 이용하여 최종적으로 차량의 움직임을 판단할 수 있으며 이러한 이동 정보를 통해 관심 차량에 대한 위험성 여부를 판단할 수 있다. 예를 들면 차량이 계속적으로 다가오는 경우, 멀어지는 경우, 측면으로 이동하거나 측면에서 이동해 들어오는 경우들에 대한 상황 판단을 통해 차량과의 충돌에 대한 가능성을 판단하고 이를 운전자에게 알려줄 수 있도록 한다.

IV. 실험 및 평가

본 실험에서는 제안한 알고리즘의 성능을 평가하기 위해 차량의 앞면 상단 중앙에 단일 CCD 카메라를 장착하여 서울 시내 및 외곽 고속도로에서 주행하면서 획득한 연속적인 영상을 이용하였다. 이렇게 획득한 도로 영상으로부터 1 프레임 단위로 각 1586 개의 정지영상을 연속적으로 추출하여 차량 검출 및 추적 기법을 적용하였으며 차량은 소형 및 중형 차량

에 대해서만 적용하였다. 실험에 사용된 영상은 320 x 240 크기를 가지는 그레이 영상을 사용하였으며 2.67 GHz i5 CPU 기반의 PC 에서 개발되어 평가하였다.

그림10 은 제안한 알고리즘에서 검출된 이전 프레임의 차량 템플릿을 저장하고 이를 통해 다음 프레임에서의 동일한 특징을 가지는 차량을 검출함으로써 차량의 위치와 크기를 계속적으로 추적해 나가는 과정을 보여주고 있다. 이 결과는 연속적으로 입력되는 영상에 적용한 결과를 보여주는 것으로 차량이 시간의 변화에 따라 근접함에 있어서도 잘 검출되는 것을 알 수 있으며 연속적인 프레임을 통해 차량의 템플릿이 업데이트 되는 것을 알 수 있다. 이 과정에서 현재 프레임의 전체 영상을 대상으로 SURF 알고리즘을 적용하는 경우 연산량이 증가하기 때문에 이전 프레임에서 검출된 차량의 위치 정보를 이용하여 일정 영역만큼 확장된 영역을 관심영역으로 지정하였다.

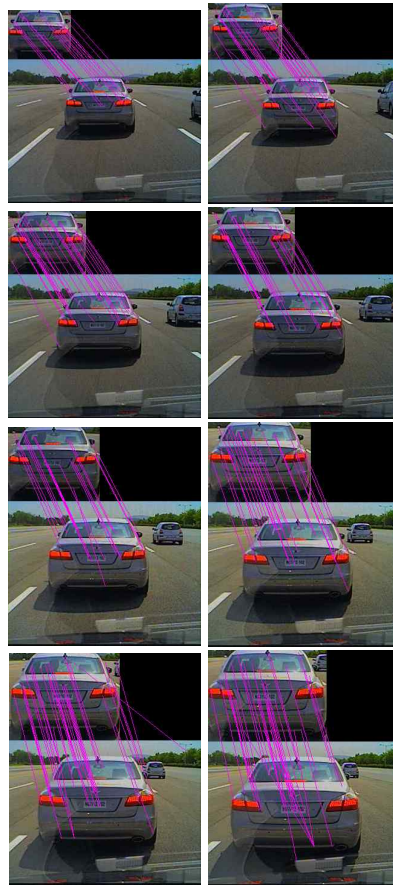
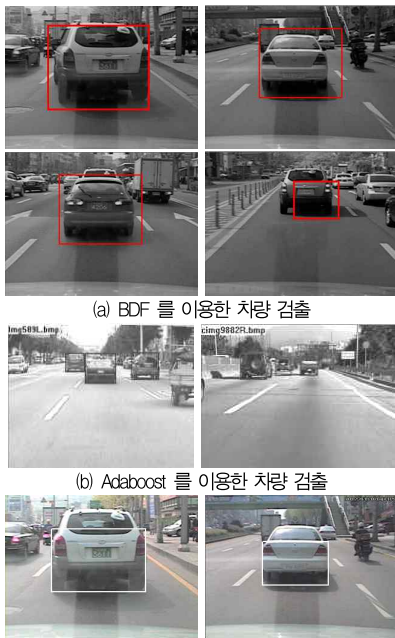


그림 10. 연속적인 차량 추적 결과
Fig. 10. Vehicle tracking results continuously

그림 11은 제안된 알고리즘을 이용하여 차량이 검출된 결과를 기존의 BDF(Bayesian Discriminating Features)[14] 기법을 이용한 차량 검출 방식과 Adaboost 기법을 이용한 차량 검출 방식[15] 과 비교하여 보여준다. 특히 얼굴 검출에서 많이 사용되는 Adaboost 방법을 이용하여 차량을 검출하는 방법의 경우 수평 에지를 통해 차량의 후보 영역을 검출하고 이를 Adaboost 를 통해 차량 유무를 판별하는 방식을 사용하고 있다. 그림 11 에서 상단 4개의 이미지는 BDF 방식을 이용한 차량 검출 결과를 보여주며 하단 줄은 제안한 알고리즘을 이용한 차량 검출 결과를 보여주고 있다.

BDF 방식의 차량 검출 알고리즘은 차량을 검출하기 위해 하나의 영상 이미지를 특정 사이즈 단위로 변환해 가며 차량 클래스의 확률모델에 적합한 위치들을 선정하기 때문에 차량의 정확한 하단 면과 측면 위치를 검출하기가 어려운 경우가 많이 발생하기 때문에 차량과의 거리 측정에 오차가 커지는 것을 알 수 있다. 또한 프레임간의 연속성을 고려하지 않기 때문에 동일한 차량인지 여부를 판단할 수 없다. 반면 제안한 알고리즘의 경우는 BDF 를 이용한 방식 보다 정확한 차량의 하단 면을 검출하여 거리를 측정할 수 있음을 알 수 있으며 연속되는 프레임에서의 동일한 차량을 검출할 수 있기 때문에 관심 차량에 대한 위치의 변화를 획득할 수 있다.



(c) 제안하는 방법을 이용한 차량 검출
그림 11. 기존 방법과 제안한 방법을 이용한 차량 검출 결과
Fig. 11. Vehicle detection result between previous method and proposed method

그림 11은 BDF 를 이용한 차량 검출 결과 (a) 와 Adaboost 를 이용한 차량 검출 결과 (b), 그리고 제안한 방법을 통한 결과 (c) 를 비교하여 보여준다. BDF 의 경우 다양한 크기의 차량을 검출해야 하기 때문에 비교적 정확한 위치를 추출할 수 없다는 단점이 있는 반면 제안한 알고리즘의 경우 정확한 차량의 외곽 위치를 추출할 수 있다는 점을 비교를 통해 알 수 있다.

표 1 은 BDF 알고리즘을 이용한 차량 검출 방식과 Adaboost 방식을 이용한 차량 검출 알고리즘을 제안한 알고리즘을 이용한 차량 검출 방식의 검출 율과 비교한 결과를 보여준다. Adaboost 를 이용한 차량 검출의 경우 기존 논문에서의 검출 결과를 참조하였다. 영상은 실제 도로 주행을 하면서 획득한 영상을 통해 1586 개의 연속적인 프레임에 적용한 결과를 보여준다.

표 1. 기존 방식과 제안한 검출 알고리즘의 비교
Table 1. Comparison proposed method with previous algorithm

	BDF	Adaboost	Proposed
정상	1322 / 1586	1379 / 1586	1453 / 1586
오류	264 / 1586	207 / 1586	133 / 1586
검출율	83.35 %	86.95 %	91.62 %

표 1 에서 알 수 있듯이 BDF 를 이용한 경우는 1586 개의 프레임에서 1322 개의 프레임에서 차량을 검출하였고 294 개의 오류가 발생하여 검출율이 83.35 % 이며, Adaboost 알고리즘의 경우 1586 개의 프레임 중 1379 개의 정상을 보여줌으로써 86.95 %의 검출을 보여주었다. 반면, 제안한 알고리즘의 경우는 1586 개의 프레임 중 1453 개의 프레임에서 정상적으로 검출되었고 133 개의 프레임에서 오류를 발생하여 91.62 %의 검출율을 보여준다. 오류가 발생한 경우는 차량의 위치에서 10 % 이상 벗어나서 검출되거나 검출되지 않은 경우로 지정하였다.

실제 차량에 적용하여 사용할 수 있는지에 대한 판단을 위해 계산 시간에 대한 측정을 진행하였다. 속도는 2.67 GHz

i5 CPU 기반의 PC 환경에서 윈도우를 기반으로 수행한 시간을 측정하였다. 표 2 는 제안된 알고리즘에서의 각 단계별 수행 시간 결과와 검출 및 추적에 대한 분야별 수행시간 및 전체 알고리즘의 프레임당 수행 시간을 보여준다.

표 2. 제안된 알고리즘의 속도 성능 측정 결과
Table 2. Result of execution time of proposed method

부분	알고리즘	수행시간	
차량 검출	Haarlike 에지 검출	0.00570 sec / frame	0.00742 sec / frame
	특징 기반 차량 검출	0.00172 sec / frame	
차량 추적	SURF 기반 특징 추출	0.00403 sec / frame	0.07558 sec / frame
	SURF 기반 매칭	0.07155 sec / frame	
총합		0.083 sec / frame => 12.048 fps	

표 2 를 통해 알 수 있듯이 차량 검출 부분에 있어서는 프레임당 0.00742 초의 시간을 보임으로서 초당 약 134 프레임의 처리 성능을 보여주었으며, 추적 부분에 있어서는 프레임당 0.07558 초의 시간을 보임으로서 초당 약 13 프레임의 속도를 보여주었다. 따라서 각 부분별 결과에 대한 전체 알고리즘의 속도 결과는 프레임당 0.083 초, 즉 초당 약 12 프레임을 처리할 수 있는 결과를 보여주었다.

본 실험에서는 속도에 대한 성능을 개선하기 위해 검색 이미지의 관심영역을 제한하는 방법과 이미지 크기를 축소하는 전처리 단계를 적용하였다. 속도 측정 결과를 통해 본 제안된 알고리즘이 저속으로 진행되는 차량에서의 실시간 처리 가능성을 알 수 있다.

V. 결론

자동차 사고를 예방하기 위한 가장 기본적인 방법으로 단일 카메라를 이용하여 전방의 차량을 효과적으로 검출하고 추적 할 수 있는 알고리즘을 제안하였다. 검출을 위해서 수직 에지 특징이 뚜렷하게 나타나는 정보를 이용하였고 이에 대한 차량의 너비와 영상에서의 위치 정보에 대한 관계 테이블을 적용함으로써 빠르고 정확한 차량의 위치를 검출할 수 있었다. 하지만 단일 영상의 경우 배경에서 나타나는 유사 특징으로 인해 오 검출 및 미 검출 되는 경우가 발생할 수 있기 때문에 이에 대한 보완을 위해 이전 프레임에서 검출된 차량 정보를 통해 SURF 알고리즘을 이용하여 이전 프레임과 현재 프

레이스에서 나타나는 특징 비교를 통해 동일한 차량을 검출하여 주변의 배경 변화에도 강인함을 보였다. 앞으로의 연구에서는, 안개 낀 날씨, 비 오는 날씨와 같은 악조건의 환경에서도 강인한 차량 특징 추출 기법에 대한 연구가 필요하다.

감사의 글

“본 논문은 2011년 정부 교육과학기술부(한국학술진흥재단)의 2단계 BK21사업의 재원으로 지원을 받아 수행된 연구이며, 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음” (NIPA-2011-(C1090-1121-0010))

참고문헌

- [1] H. Cheng, N. Zheng, X. Zhang, J. Qin and H. Wetering, "Interactive Road Situation Analysis for Driver Assistance and Safety Warning Systems : Framework and Algorithms," IEEE Transactions On Intelligent Transportation Systems, Vol. 8, No. 1, pp. 157-167, March 2007.
- [2] K. A. Redmill, S. Upadhy, A. Krishnamurthy and U. Ozguner, "A Lane Tracking Systemfor Intelligent Vehicle Applications," 2001 IEEE Intelligent Transportation Systems Conference Proceedings, pp. 273-279, August ,2001.
- [3] Z. Sun, G. Bebis and R. Miller, "Monocular Pre-crash Vehicle Detection : Features and Classifiers," IEEE Transactions On Image Processing, Vol. 15, No. 7, pp. 2019-2034, July 2006.
- [4] C. C. R. Wang and J. J. J. Lien, "Automatic Vehicle Detection Using Local FeaturesA Statistical Approach," IEEE Transactions On Intelligent Transportation Systems, Vol. 9, No. 1, pp. 83-96 March 2008.
- [5] G. J. Burghouts, J. M. Geusebroek, "Performance evaluation of local colour invariants," Computer Vision And Image Understanding, Vol. 113, No. 1,

pp. 48-62, July 2009.

[6] K. van de Sande, T. Gevers, C. Snoek, "A comparison of color features for visual concept classification," Conference On Image And Video Retrieval, pp. 141-150, July 2008.

[7] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 27, No. 10, pp. 1615-1630, February 2005.

[8] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," Int'l J. Computer Vision, Vol. 60, No. 2, pp. 91-110, January 2004.

[9] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," European Conference On Computer Vision, Vol. 3951, pp. 404-417, May 2006.

[10] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," Proc. Alvey Vision Conf., pp. 147-151, 1988.

[11] T. Lindeberg, "Feature detection with automatic scale selection," International Journal Of Computer Vision, Vol. 30, No. 3, pp. 79-116, 1998.

[12] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," International Conference Computer Vision, Vol. 1 pp. 525-531, July 2001.

[13] P. Viola, M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," Computer Vision And Pattern Recognition, Vol. 1, pp. 511-518, December 2001.

[14] C. Liu, "A Bayesian discriminating features method for face detection," IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 25, No. 6, pp. 725-740, June 2003.

[15] G. Y. Song, K. Y. Lee, J. W. Lee, "Vehicle Detection Using Edge Analysis and AdaBoost Algorithm," Transactions Of The Korean Society Of Automotive Engineers, Vol. 17, No. 1, pp.1-11, 2009.

저 자 소개



유 재 형
 2007 : 송실대학교 정보통신전자공학부 공학사.
 2009 : 송실대학교 전자공학과 공학석사.
 현 재 : 송실대학교 전자공학과 박사과정
 관심분야: 영상처리, 임베디드 시스템
 Email : caution0@ssu.ac.kr



한 영 준
 1996 : 송실대학교 전자공학과 공학사.
 1998 : 송실대학교 전자공학과 공학석사.
 2003 : 송실대학교 전자공학과 공학박사
 현 재 : 송실대학교 정보통신전자공학부 부교수
 관심분야 : 로봇 비전, 컴퓨터 비전, 비주얼 서보잉
 Email : young@ssu.ac.kr



한 현 수
 1981 : 송실대학교 전자공학과 공학사.
 1983 : 연세대학교 전자공학과 공학석사.
 1991 : University of Southern California 전기공학과 공학박사
 현 재 : 송실대학교 정보통신전자공학부 교수
 관심분야 : 자동화시스템, 센서융합, 물체인식
 Email : hahn@ssu.ac.kr