

논문 2013-50-10-18

PCA 복원과 HOG 특징 기술자 기반의 효율적인 보행자 인식 방법

(An Efficient Pedestrian Recognition Method based on PCA Reconstruction and HOG Feature Descriptor)

김철문*, 백열민**, 김회율***

(Cheol-Mun Kim, Yeul-Min Baek, and Whoi-Yul Kim[Ⓢ])

요약

최근 보행자의 교통안전 개선을 위한 목적으로 차량에 장착되는 보행자 보호 시스템(PPS, Pedestrian Protection System)에 대한 관심과 요구가 증가하고 있다. 본 연구에서는 보행자 검출 후보 윈도우 추출과 셀(cell) 단위 히스토그램 기반의 HOG 특징 계산 방법을 제안하였다. 보행자 검출 후보 윈도우 추출은 주변밝기 비율체크, 수직방향 에지투영, 에지팩터(edge factor), 그리고 PCA(Principal Component Analysis) 복원 영상을 이용하였다. Dalal의 HOG는 겹침 블록 상의 모든 픽셀에 대해 가우시안 가중치와 삼선형보간에 의한 히스토그램 계산이 필요한데 반하여 제안하는 방법은 단위 셀마다 가우시안 가중 및 히스토그램을 계산하고 그것들을 인접 셀과 결합하므로 연산 속도가 빠르다. 제안하는 PCA 복원 에러 기반의 보행자 검출 후보 윈도우 추출은 보행자의 머리와 어깨 영역과의 차이를 기준으로 배경을 효율적으로 분류한다. 제안하는 방법은 카메라 캘리브레이션이나 스테레오 카메라를 이용한 거리 정보 없이도 영상만으로 전통적인 HOG에 비하여 연산속도가 크게 개선된다.

Abstract

In recent years, the interests and needs of the Pedestrian Protection System (PPS), which is mounted on the vehicle for the purpose of traffic safety improvement is increasing. In this paper, we propose a pedestrian candidate window extraction and unit cell histogram based HOG descriptor calculation methods. At pedestrian detection candidate windows extraction stage, the bright ratio of pedestrian and its circumference region, vertical edge projection, edge factor, and PCA reconstruction image are used. Dalal's HOG requires pixel based histogram calculation by Gaussian weights and trilinear interpolation on overlapping blocks, But our method performs Gaussian down-weight and computes histogram on a per-cell basis, and then the histogram is combined with the adjacent cell, so our method can be calculated faster than Dalal's method. Our PCA reconstruction error based pedestrian detection candidate window extraction method efficiently classifies background based on the difference between pedestrian's head and shoulder area. The proposed method improves detection speed compared to the conventional HOG just using image without any prior information from camera calibration or depth map obtained from stereo cameras.

Keywords : Histogram of oriented gradients, 보행자 인식, PCA 복원.

* 정회원, 피엘케이 테크놀로지 (PLK Technologies)

** 정회원, 현대모비스 (Hyundai MOBIS)

*** 정회원, 한양대학교 전자컴퓨터통신공학과 (Department of Electronics Computer Engineering, Hanyang University)

ⓈCorresponding Author(E-mail: wykim@hanyang.ac.kr)
접수일자: 2013년6월18일, 수정완료일: 2013년10월3일

I. 서론

보행자 인식 시스템의 연구는 컴퓨터비전의 분야의 주요 관심분야 중 하나로 급격하게 발전해왔다. 그중에서도 교통사고로부터 보행자를 보호하기 위한 목적으로 ADAS(Advanced Driver Assistance Systems)의 한 중

류인 보행자 보호 시스템(Pedestrian Protection System, PPS)에 대한 높은 관심과 함께 시스템 도입에 대한 필요성이 증대되고 있다. 교통사고에 의한 인명손실은 전 세계적으로 심각한 문제이다. 차량에 내장된 PPS는 카메라를 통해 입력된 영상을 처리하여 보행자를 포함한 전방의 장애물을 미리 감지하여 보행자 회피 및 브레이크를 제어하므로 운전자의 부주의에 의한 교통사고를 크게 줄일 수 있다.

대표적인 차량용의 보행자 보호 시스템 연구 중 하나로 PROTECTOR 시스템이 있었다. PROTECTOR 시스템에서는 스테레오 카메라를 통해 계산된 깊이 맵을 기준으로 관심 영역을 정한 후 계층적인 템플릿 기반 분류기를 이용하여 효율적으로 보행자를 검출하였다^[1].

ADAS의 PPS에 적용될 수 있는 보행자 인식 기술은 검출 성능도 중요하지만, 순간적인 미인식이 발생하더라도 주행 중인 차량에서 보행자를 빠르게 검출하여 즉각적인 경보 혹은 제어가 이루어지는 것이 매우 중요하다.

빠른 보행자 검출을 위한 기존의 접근 방법은 크게 sliding window 기반의 방법들과 knowledge 기반의 접근방법들로 나뉜다.

Sliding window 기반의 접근 방법은 보행자 검출 시 내부의 특징 값을 고속으로 추출하는 방법이다. 그 예로 누적영상(integral image)를 이용하여 스케일 변화에 불변한 haar-like 특징을 사용한 방법^[2-3]이 있다. 보행자 인식에 있어서 haar-like 특징은 검출 속도는 빠르지만 다양한 특징 값들을 결합한 ChnFtrs^[4]나 지역적인 블록내부에서 구한 경사도(gradient)의 방향성분 히스토그램 기반의 HOG 특징^[5]과 비교하여 검출 성능이 낮다. 반면에 ChnFtrs와 HOG 같이 추출이 복잡한 특징들은 인식률은 높으나 속도가 느린 단점이 있다.

ChnFtrs 방법을 제안한 Dollar는 특징 값 추출의 속도를 높이기 위해 k 번째 스케일에서만 ChnFtrs 특징을 추출하고 나머지는 인접한 스케일에서 특징 값을 근사하여 추출하는 FPDW 방법^[6]과 이미지 pyramid 생성 과정 없이도 N/k 번의 sliding windows의 스케일 변화에 대응되는 분류기를 따로 생성하여 입력 영상에서 영상의 스케일 변화없이 검출이 가능한 VeryFast 방법^[7]을 제안하였다.

고속처리를 위해 GPU 코드로 구현된 ChnFtrs와 VeryFast 알고리즘은 SVM 분류기와 결합되었을 때

각각 1.38Hz 과 2.68Hz로 작동하였다.

Sliding window 검출 기반의 다른 방법들로는 특징 값 추출 및 분류에 있어 느린 처리 속도 문제점을 개선하기 위해 coarse-to-fine 분류기 구조를 이용한 검출 방법들^[8-9]이 있다. 이러한 검출 방법들은 Coarse 단계에서 특징이 단순하면서 판단속도가 고속인 분류기를 통해 보행자가 될 가능성이 낮은 배경을 제외하고, fine 단계에서 나머지 sliding window에 대해 복잡하고 속도가 느리지만 분류 성능이 높은 fine 검출을 수행한다.

Knowledge 기반의 접근방법은 영상 내 보행자에 대한 사전 지식을 이용하여 보행자가 될 가능성이 있는 영역에 대해서만 특징 값 추출 및 보행자 판단 과정을 수행함으로써 검출 속도를 개선할 수 있다. 이러한 접근 방법으로 영상에서 추출한 보행자의 수직 에지성분을 이용한 연구^[10-12]가 있다. 또한 수직축을 기준으로 강력한 대칭성의 가정을 이용하여 보행자를 검출하는 접근방법들^[10, 13-17]이 제안되었다.

야간의 빠른 보행자 추출을 위한 방법으로 근적외선 영상에서 수직 성분 마스크의 필터 값이 가장 큰 경우만 에지 값을 구함으로써 초기 보행자의 검출 영역을 분할하였다^[11]. 스테레오 카메라에서 대칭성을 이용한 연구로 고도맵(elevation map)으로부터 추출된 보행자, 자동차, 도로 폴대와 같은 오브젝트의 윈도우를 구하고, 밝기 값과 이진 에지를 기준으로 두 개의 수직 대칭 맵에서 구한 히스토그램들의 가중 합을 기준으로 보행자 검출 후보를 한정하는 방법이 제안^[15]되었다.

기타 연구로 단안카메라의 영상만으로 건물, 사람, 자동차, 나무 등과 같이 지면에 수직이라 판단된 영역들의 후보 윈도우를 생성하는 방법^[18]이 제안되었다. 원적외선 기반의 보행자 인식 시스템에서는 픽셀 밝기 값 투영을 통해 보행자의 후보영역을 추출^[19]하였다.

본 논문에서는 knowledge 기반의 보행자검출 후보 추출과 고속의 HOG 특징 및 분류를 통한 효율적인 보행자 검출 방법을 제안한다. 보행자검출 후보 추출 단계에서는 보행자 영역과 주변의 밝기 차이와 에지팩터(edge factor), 수직방향 에지성분, 머리와 어깨 부분 영상의 PCA(Principal Component Analysis) 복원 영상과 원영상의 차이를 이용하여 배경영역의 검출 윈도우를 제거하였다. Dalal이 제안한 기존의 HOG^[5] 특징은 모든 블록에 대하여 픽셀 단위로 에지성분의 크기 및 방향 성분 값을 참조하여 계산하였으나, 제안하는 방법은

겹쳐지는 블록 영역 내부 셀의 히스토그램을 먼저 계산해 놓고 블록 내의 HOG 계산 시 이를 가중 결합하는 방법을 통해 고속으로 연산이 가능하다.

II. 본 론

1. 제안하는 방법의 전체적인 구성

그림 1은 제안하는 보행자 검출 방법의 전체적인 블록 다이어그램 이다. 제안하는 방법은 전처리, 보행자 검출 후보추출, 보행자 검출 단계로 구성된다.

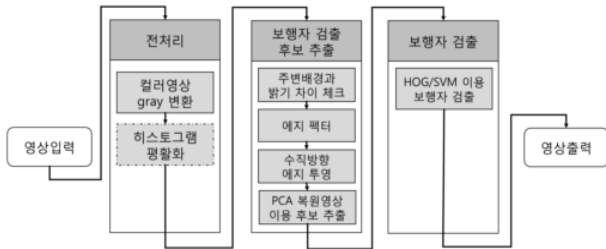


그림 1. 제안하는 방법의 블록도
Fig. 1. Block diagram of the proposed method.

2. 전처리

전처리 단계에서는 입력 컬러 영상에 대한 gray 변환을 수행한다. Gray 영상은 보행자 검출 후보 추출에 사용된다. 전체 영상의 밝기 분포를 균일화하기 위해 gray 영상에 히스토그램 평활화를 수행한다.

3. 보행자 검출 후보 추출

보행자 인식 단계에서 가장 많은 연산 시간을 차지하는 단계는 보행자 검출 윈도우 내의 특징 값 계산 단계이다. 그러므로 보행자가 존재하기 어려운 영역을 미리 제거하여 보행자 검출 단계의 연산 시간을 줄이는 과정이 필요하다.

가. 주변 배경과 밝기 차이 체크

보행자 검출 후보를 위한 첫 번째 단계는 검출 윈도우 내에서 보행자와 그 주변부의 밝기 변화를 측정하는 단계이다. 보행자 학습과 검출을 위해 사용한 검출 윈도우는 64X128 크기로, 그림 2와 같이 보행자 영역을 중심으로 둘레에 16 픽셀의 가장자리를 포함하였다.

영상 내에서 크기가 다른 보행자의 검출을 위해 입력 영상의 스케일을 일정하게 변화하며 64X128 크기의 검

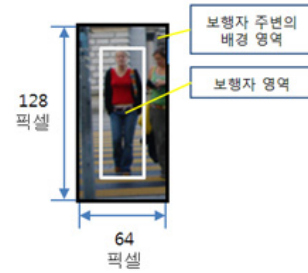


그림 2. 보행자 검출 윈도우
Fig. 2. Pedestrian detection window.

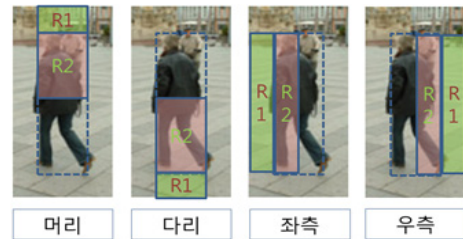


그림 3. 보행자 주변 밝기 차이 계산 영역
Fig. 3. Bright difference calculation area around a pedestrian.

출 윈도우 내의 보행자 여부를 판단하였다.

밝기 차이 계산을 위해 그림 3과 같이 설정한 윈도우에서 위쪽, 아래쪽, 왼쪽과 오른쪽 영역에서 각각 밝기 차를 비교한다. 위쪽의 경우 보행자의 머리와 어깨를 포함한 영역에 해당되며 아래쪽은 다리, 좌우는 보행자의 좌우측면에 해당된다.

보행자 영역과 배경 영역간의 밝기 변화 차이 $d_{R1, R2}$ 는 식(1)을 이용하여 계산한다.

$$f_{R1} = \frac{\sum_{x,y \in R1} f(x,y)}{N_{R1}}, f_{R2} = \frac{\sum_{x,y \in R2} f(x,y)}{N_{R2}} \quad (1)$$

$$d_{R1, R2} = |f_{R1} - f_{R2}|$$

식 (1)에서 $R1$ 과 $R2$ 는 그림 3에서 나타내는 것과 같이 보행자 영역 일부와 인접한 배경 영역이며, f_{R1} 과 f_{R2} 는 영역 내부의 밝기 합을 영역 내 픽셀 수인 N_{R1} 과 N_{R2} 로 각각 나눈 평균 밝기 값이다. 영역 내의 밝기 합은 입력 그레이 영상의 누적 영상(integral image)을 계산하고 이를 이용하여 고속으로 구한다. 만약 머리와 다리영역에서 구한 주변밝기 차이의 합 혹은 좌측과 우측 몸통 영역에서 구한 주변 밝기 차이의 합이 설정된 임계 값 이내이면 해당 서브 윈도우는 곧바로 제

거된다.

나. 에지팩터

에지팩터는 밝기 변화가 일정한 배경과 보행자를 구분하기 위해 사용되었다^[8]. 보행자가 존재하지 않으면서 밝기 값 변화가 적은 하늘, 빌딩, 건물, 도로 바닥 면에서의 경사도는 보행자 내부의 경사도 크기보다 작다. 동시에 큰 경사도 값을 지닌 픽셀의 수도 작다. 에지팩터는 영상에서 gradient 의 크기가 임계 값 보다 큰 픽셀 수를 윈도우의 크기만큼 나눈 값이다. Dalal 이 HOG 특징 추출 시 사용했던 기존방법^[5]에서 에지팩터 계산에 필요한 x, y 픽셀 좌표에서의 gradient 의 크기 $grad(x, y)$ 는 식(2)와 같이 구하였다.

$$grad(x, y) = |G_x| + |G_y| \quad (2)$$

식(2)에서 G_x 와 G_y 은 x와 y 방향의 1 차원 gradient 이다. Gradient 의 크기는 연산을 단순화하기 위해 G_x 와 G_y 의 절대 값의 합으로 근사하여 구하였다.

하지만 식(2)를 이용하여 구한 근사 값은 G_x 와 G_y 가 증가함에 따라 실제 값과 오차가 커지게 된다. 제안하는 방법에서는 식(2)의 gradient 크기의 근사치를 사용했을 때의 오차에 따른 영향을 없애기 위해 실제 gradient 크기 그대로를 적용하였다. 대신에 x, y 방향의 1차원 픽셀 변화율 값에 해당되는 $grad(x, y)$ 값을 미리 계산하여 저장한 룩업테이블을 참조하여 빠른 연산이 가능하도록 하였다. 또한 윈도우 영역 내에서 임계 치 보다 큰 픽셀 수는 누적 영상을 이용하여 빠르게 계산하였다.

다. 수직방향 에지투영

지면에 서있는 보행자는 몸통 및 다리 부분에서 수평 방향의 밝기 변화가 잘 나타난다. 또한 영상에서 수평 에지성분은 보행자의 좌, 우측 경계에서 크기가 크면서 상하로 연속하여 나타난다. 제안하는 방법은 이러한 특성을 검사하기 보행자의 검출 윈도우를 x 좌표 기준으로 8 픽셀 씩 이동하며 너비 16, 높이 128 크기의 서브 윈도우 내의 수평 에지 합을 구하였다.

이후 좌, 우측에서 서브 윈도우 중 수평 에지의 최대 값을 구하여, 그 값이 임계 값 이하인 경우 해당 윈도우를 보행자 검출 후보에서 제거하였다. 서브 윈도우 내

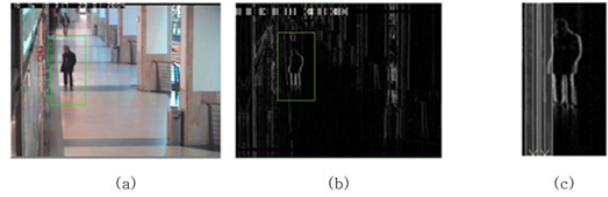


그림 4. 보행자 수직방향 에지투영 (a) 컬러입력영상, (b) 수평방향 에지계산, (c) 겹침 윈도우 영역 안에서 수직방향 에지투영

Fig. 4. Vertical edge projection (a) color input image (b) horizontal edge calculation (c) vertical edge projection in overlapped window area.

수평 에지의 합은 수평 에지영상의 누적영상을 이용하여 고속으로 계산하였다.

라. PCA 복원영상 이용 후보추출

Gilherme은 PCA 복원영상과 원 영상간의 복원에러를 기반으로 한 보행자 인식 모델을 제안^[20]하였다.

그러나 Guilherme 이 제안한 방법의 PCA 학습에서 사용된 영상 밝기와 에지정보 만으로는 보행자가 가진 다양한 자세와 걸모습을 효과적으로 표현하기 어렵다. 또한 사용된 주성분 벡터에 따라 최적 결과를 위한 복원에러의 가중치를 조절해주어야 한다.

본 연구에서는 PCA 복원 영상의 유사도 판단에 의한 보행자 접근 검출 아이디어를 보행자 후보 추출에 사용하였다. 제안하는 방법은 32X32 크기의 보행자의 머리 및 어깨 영역의 gray 영상을 이용하여 PCA 학습을 한 후에, 원영상과 PCA 복원 영상 유사도를 기준으



그림 5. 보행자의 머리 어깨 영역에서 원영상과 PCA 복원 영상간의 차영상 생성 (a) 검출 윈도우 생성, (b) 추출된 머리 어깨 영역, (c) PCA 복원영상, (d) (b)와(c)의 절대값 차영상

Fig. 5. Difference image generation between the original image and the PCA reconstructed image in the head and shoulders area. (a) image in detection window (b) extracted head and shoulders area (c) PCA reconstructed image (d) absolute difference image between (b) and (c).

로 패턴이 상이한 배경 영역을 보행자 후보에서 제외하였다. 영상에서 머리와 어깨 부분은 보행자의 다양한 자세 변화에 따른 패턴의 변화가 적은 곳으로 PCA 학습을 통하여 보행자의 머리 및 어깨 윤곽에 대한 주요한 특징을 찾아내고 적은 차원의 데이터로 보행자의 패턴을 효율적으로 나타낼 수 있다.

원 영상과 복원 영상간의 유사도 d 는 식(3)과 같이 두 영상의 절대 값 차이의 합(SAD, Sum of Absolute Distance)으로 구하였다.

$$d_i = |u_i - u'_i|, d = \sum_i d_i \quad (3)$$

식 (3)에서 u_i 와 u'_i 는 각각 원영상과 복원영상의 동일 위치에서의 픽셀 밝기 값이다.

만약 d 가 설정된 값 미만일 경우 보행자와 유사한 영역이라 판단하고 보행자 후보로 추출한다.

제안하는 방법에서는 16개의 주성분 벡터만을 사용하여 영상을 복원하였다. 또한 PCA 복원된 영상과 원영상간의 SAD의 빠른 계산을 위해 CPU 에 탑재된 SSE 명령어를 사용하여 한번에 16 픽셀의 절대 값 및 누적 합을 계산하도록 구현하였다.

4. 보행자 검출

보행자 검출 단계에서는 추출된 보행자검출 후보에 대해서만 HOG기술자를 계산하고 SVM 분류기의 예측 결과를 기준으로 보행자를 구분한다.

가. 셀 단위 히스토그램 기반의 HOG 블록 기술자

보행자 검출을 위한 최종 HOG 기술자는 검출 윈도우 내부에 존재하는 블록에서 구한 HOG 기술자를 모두 결합함으로써 구한다. 검출 윈도우 내부의 HOG 기술자 블록 구조는 셀(cell)단위로 겹쳐져서 배치된다. 셀은 HOG 기술자 생성을 위한 경사도 크기 히스토그램을 계산하는 단위로 Dalal 이 제안한 HOG 에서는 엘리어싱(aliasing)에 의한 영향성을 감소하기 위해 경사도 크기 성분은 주변 히스토그램 bin 및 블록 내부의 인접 셀 간에 보간법(interpolation)을 통해 경사도 크기 성분을 누적한다.

그림 6은 2X2 개의 셀로 구성된 블록에서 삼선형 보간을 통하여 인접 셀로 히스토그램 bin 이 누적되는 패

셀		블록	
1	1, 3	3, 1	3
1, 2	1, 2, 3, 4	3, 4, 1, 2	4, 3
2, 1	2, 1, 4, 3	4, 3, 2, 1	4, 3
2	2, 4	4, 2	4

그림 6. 삼선형 보간을 위한 주변 셀로의 누적 패턴
Fig. 6. Pattern of the surrounding cells which is accumulated by tri-linear interpolation.

턴을 보여준다. 예를 들어 2번 셀과 인접한 1번 셀 내부의 서브 셀(sub-cell)은 1, 2로 표시되는데, 이는 해당 서브 셀에 위치한 픽셀에서 계산된 경사도 크기 성분은 1번과 2번의 히스토그램으로 보간 누적됨을 의미한다.

Dalal이 제안한 인접 셀 간의 삼선형 보간 과정은 픽셀 단위로 HOG 기술자 블록의 위치에 따라 다른 서브 셀(sub-cell)의 가중치로 경사도 값을 히스토그램에 누적하므로 연산량이 매우 크다.

이러한 계산 과정을 단순화 하여 속도를 개선하기 위해 본 논문에서 제안하는 방법은 먼저 셀 단위의 히스토그램을 먼저 계산하고 이를 가중 결합하여 HOG 블록 기술자를 구하는 것이다. 셀 단위의 히스토그램의 결합에 의한 HOG 블록 기술자는 그림 6에 제시된 2X2 개의 셀에 대하여 서브 셀들로부터 1, 2, 3, 4번 셀로 누적되는 가중치의 총합을 구하여 그림 7과 같이 각 셀마다의 테이블로 저장한다.

그림 7은 블록 기술자 계산을 위한 가중치 테이블 참조를 통한 인접 셀로의 히스토그램 누적 패턴을 보여준다. 블록 기술자는 블록 내부의 각 셀에서 구한 히스토그램에 1, 2, 3, 4번으로 누적되는 누적 가중치를 곱하여 인접 셀의 히스토그램에 가중하여 더한다. 제안하는 방법은 Dalal 의 방법과 달리 동일한 셀에 대하여 블록의 위치에 따른 픽셀 단위의 반복 계산을 피하므로 연산속도를 크게 향상 시킬 수 있다.

셀		블록	
1, 2, 3, 4		3, 4, 1, 2	
2, 1, 4, 3		4, 3, 2, 1	

그림 7. 셀 단위 히스토그램의 주변 셀로의 누적패턴
Fig. 7. Cumulative pattern into the surrounding cells in unit cell histogram based method.

실험을 통해 셀 단위의 히스토그램에 의한 HOG 기술자 계산은 삼선형 보간법을 사용한 기존 방법^[5]과 비교하여 인식성능은 거의 동일하면서도 검출 속도는 매우 개선됨을 확인하였다.

나. HOG/SVM 이용 보행자 검출

HOG 기술자의 계산은 에지의 크기와 각도 성분을 계산하는 데 있어서 제곱근과 역탄젠트 연산이 필요하고, 히스토그램 누적 시에는 엘리어싱에 의한 영상을 줄이기 위해 인접 각도 bin과 보간을 수행한다. 이러한 과정은 복잡한 연산이 필요한 부분이다. 제안하는 방법은 특징 값 계산의 효율성을 높이기 위해 수평, 수직 방향의 가능한 에지값 조합에 대해 결과를 미리 계산하여 룩업 테이블(look up table)에 저장하여 HOG 특징 계산 시 참조하였다.

제안하는 방법은 전체 검출 윈도우 중에서 보행자검출 후보로 추출된 윈도우에 내부에 대해서만 HOG 특징을 계산하면 된다. 이를 위해 lazy descriptor 라는 새로운 블록 단위 HOG 특징 값 계산구조를 제안하였다. Lazy descriptor 방법은 전체 영상 내에서 계산되는 전체 블록의 수만큼 선언된 배열을 이용하여 특정 블록 내부의 특징 값이 이미 계산되어 있는지 여부를 체크한다. 만약 검출 윈도우 내부 블록의 HOG 특징을 이미 계산한 적이 있다면 해당 값을 복사하여 그대로 사용한다. 하지만 계산한 적이 없는 블록이라면 내부의 HOG 기술자를 계산하고 해당 블록 위치에 계산 여부를 체크한다.

보행자 분류기는 보행자 영상과 보행자를 포함하지 않는 배경의 샘플들에서 HOG 특징 값을 추출한 후, SVM 학습을 통해 보행자와 배경간의 마진을 최대화하는 초평면(hyper plane)을 구함으로써 생성된다.

III. 실험

제안하는 방법의 성능 평가를 위해 전통적인 HOG 방법과 동일한 파라미터 (셀 크기, 블록 크기, 블록의 겹침 정도, 검출 윈도우의 step, 영상 스케일 변화)를 설정하였다. 성능 평가에 사용된 시스템은 Intel 3.4 Ghz CPU 와 8GB DDR3 RAM 으로 구성된다. 트레이닝과 인식 성능의 실험을 위해 INRIA 의 보행자/배경 데이터 셋^[21]을 활용하였다. 트레이닝에 사용된 보행자

영상은 1208 장의 영상과 각 영상에 대한 좌우 반전 영상으로 총 2416 장이 사용되었다. 배경 트레이닝 샘플 추출을 위해 보행자가 포함되지 않은 1218 장의 영상에서 임의의 스케일과 위치에서 총 36540 장의 영상을 추출하였다. 추출된 HOG 특징 값의 SVM 학습을 위해 SVMLight 라이브러리^[22]를 이용하였다. 이후 학습된 분류기를 이용하여 배경 트레이닝에 사용된 영상을 입력하여 모든 스케일 및 위치에서 배경 내에서 보행자로 오인식이 발생한 검출 윈도우 내의 영상을 추출하고, 이들을 더하여 다시 SVM 학습을 수행하여 최종 HOG/SVM 분류기를 생성하였다. PCA 를 통한 보행자의 머리, 어깨 영역의 학습은 1208 장의 영상 중 550 장의 영상을 무작위 선별 후 이용하였다.

1. 동영상에서의 인식성능 측정

보행자 성능 측정을 위해 CAVIAR 의 테스트 영상^[23]을 사용하였다. 그림 8은 동일 장면에서 Dalal의 HOG 기술자^[5]와 제안하는 방법의 검출 결과를 비교하였다. 각 스케일 상에서 검출되는 결과의 차이로 최종 검출 윈도우의 위치에서 약간의 차이가 발생하나, 두 방법을 적용하였을 때 거의 유사한 위치에서 보행자가 검출됨을 알 수 있다.

정량적인 인식성능 비교를 위해 테스트 영상에서 인식율과 오검출 수를 계산하였다. 영상에서 가장 자리에 위치한 보행자 검출이 가능하도록 영상 둘레에 16 픽셀의 padding 영역을 설정하였다. 보행자 영역의 일부가 영상에서 벗어나는 경우에는 인식률 측정에서 제외

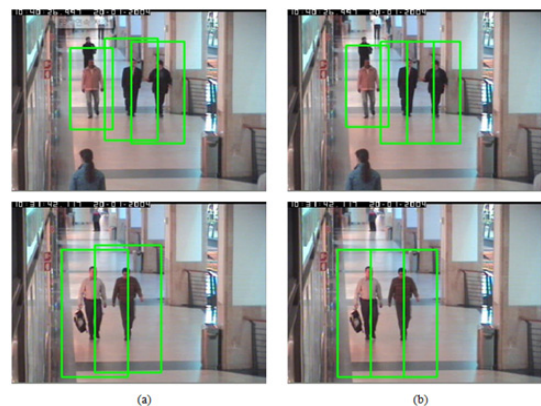


그림 8. 인식 결과 비교 (a) 전통적인 HOG 방법, (b) 제안하는 방법

Fig. 8. Recognition result comparison. (a) Dalal's HOG (b) proposed method.

표 1. 제안하는 방법과 Dalal의 HOG만 적용 시 성능 비교

Table 1. Performance comparison between proposed method and Dalal's HOG.

	제안하는 방법	Dalal의 HOG/SVM
영상 내 검출대상 보행자의 수	10416	10416
인식한 보행자의 수	9720	9911
보행자 인식률	93.31%	95.15%
오인식 개수	0	0

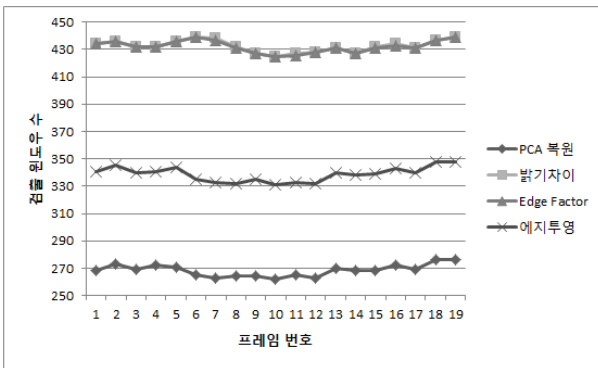


그림 9. 보행자 후보 추출 및 보행자 선 검출 단계 후 검출 윈도우 수 변화

Fig. 9. Number of detection windows after pedestrian candidate extraction step.

하였다.

표 1은 CAVIAR 데이터 셋에서 전체 보행자 수 대비 인식한 보행자의 수를 측정한 결과이다. 제안하는 방법과 Dalal의 방법 모두에서 배경을 보행자로 오검출이 발생하지 않았다.

제안하는 방법은 Dalal의 방법과 비교하여 거의 동등한 보행자 인식 성능을 보인다. 제안하는 방법에서 인식한 보행자 수의 차이는 보행자 후보 추출 단계에서 실제 보행자 윈도우를 배경으로 오 판단하여 발생한다.

제안하는 보행자 후보 영역 추출 방법의 효율성을 측정하기 위해, 보행자가 존재하지 않는 배경 영상에 대하여 후보 영역 추출 단계 별 검출 윈도우 수를 구하여 그림 9에 제시하였다. 시험에 사용한 영상의 크기는 384X288 이고 전체 영상 스케일에서 생성되는 검출 윈도우의 총합은 2191개이다.

그림 9의 측정 결과를 보면 주변 배경과 밝기차이 체크에 의해 많은 수의 보행자 검출 윈도우가 제거 되었으나 에지팩터를 거친 후에는 윈도우의 감소는 거의 없다. 이는 밝기차이 체크에 의해 바닥면 등과 같이 밝기

변화가 크지 않은 영역이 대부분 제거되었기 때문이다. 한편 보행자 후보 검출 마지막 단계인 PCA 복원영상을 이용한 후보추출을 거치면서 보행자의 머리, 어깨 패턴과 유사도가 낮은 배경인 후보 윈도우가 매우 효과적으로 제거됨을 확인할 수 있다.

동작 속도 측정을 위해 제안하는 보행자 후보 추출 방법 및 셀 단위 히스토그램 기반의 HOG 블록 기술자 특징 추출 방법을 적용했을 때 최적화된 코드로 구현된 Dalal의 HOG 방법을 적용 시 384X288 크기의 동영상에서 초당 처리 프레임을 계산하였다. 동일한 파라미터 상에서 Dalal의 HOG는 최대 8 FPS 속도로 동작하나, 제안하는 방법을 모두 적용하면 약 21 FPS 정도로 작동하였다.

2. 셀 단위 히스토그램 기반의 HOG 블록 기술자 성능 측정

보행자 검출 후보추출 이후 보행자 검출을 위해 논문에서 제안된 셀 단위 히스토그램 기반의 HOG 기술자와 Dalal이 제안한 HOG 기술자^[5]의 정량적인 검출 성능을 비교하였다.

보행자의 검출 성능 지표로써 INRIA 에서 제공하는

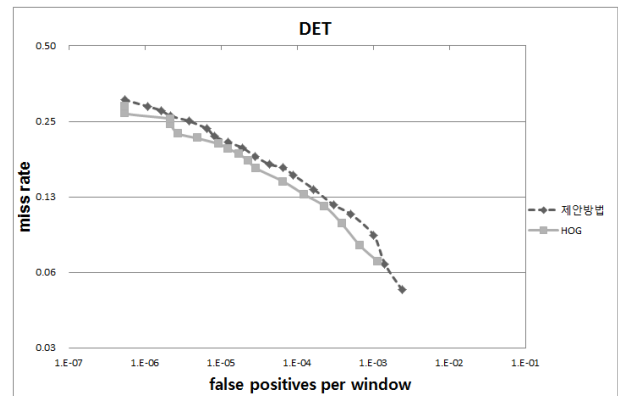


그림 10. 제안하는 방법과 HOG 간의 검출성능 지표

Fig. 10. Performance indicator between proposed method and Dalal's HOG.

표 2. 제안하는 방법과 Dalal이 제안한 HOG 기술자의 평균 검출시간

Table 2. Average detection time of proposed method and Dalal's HOG.

	제안하는 방법	Dalal의 HOG/SVM
평균검출시간	0.748ms	1.787ms

보행자/배경 데이터 셋으로부터 동일 파라미터를 가지고 계산된 FPPW(False Positives Per Window) 와 미검출율(missing rate)을 측정하였다.

또한 검출 속도 비교를 위해 64X128 사이즈의 검출 윈도우 내부의 특징 값 추출 및 판별에 소요되는 평균 시간을 측정하였다.

그림 10과 표 2부터 알 수 있듯이 제안하는 셀 기반의 블록 기술자 계산방법을 사용하면 인식 성능의 차이는 매우 적으면서도 검출 속도는 2 배 이상 개선됨을 확인하였다. 또한 FPPW가 극히 적은 수치임을 감안하면 제안하는 방법의 보행자 성능이 약간 더 우수하게 측정되었다.

IV. 결 론

본 논문에서는 보행자 후보 검출과 PCA 복원 영상을 이용한 보행자 선 검출 과정을 통하여 연산 복잡도가 높은 HOG 특징 값 계산이 필요한 보행자 검출 윈도우의 수를 줄임으로써 보행자 검출 속도를 높일 수 있는 방법을 제안하였다. 보행자 샘플 영상의 머리와 어깨 영역에서 추출한 크기의 영상과 PCA 보행자 학습 과정에서 추출한 16개의 주성분만으로 입력 영상에서 보행자 머리와 어깨의 영상을 유사하게 복원함으로써 보행자와 밝기 변화 패턴을 갖는 보행자 검출 윈도우를 효과적으로 판단할 수 있었다.

본 논문에서 제안한 lazy descriptor 구조는 검출 윈도우에서 HOG 특징 값을 추출해야 할 필요성이 있는 경우에만 블록 및 셀 내부의 특징 값을 추출하여 캐시에 저장하기 때문에 보행자 후보 영역 추출방법과 결합하여 불필요한 HOG 특징 값 계산을 효과적으로 줄일 수 있다. 또한 룩업 테이블을 이용하여 계산 량이 많은 제공된 계산을 피하고, 인접 각도 bin으로의 보간을 테이블의 값 참조만으로 가능하도록 개선하였다.

본 논문에서 제안하는 방법은 카메라 컬리브레이션 정보나 스테레오 카메라에서 구한 깊이 정보 등을 이용하지 않고도, 영상만으로 보행자와 유사한 검출 후보 윈도우를 효과적으로 선택할 수 있다. 또한 본 논문에서 제안하는 보행자 검출을 위한 HOG 기술자는 기존 HOG 와 비교하여 인식 성능이 동등하면서도 검출 시간은 크게 줄어드는 것을 확인하였다.

REFERENCES

- [1] D. Gavrila, J. Giebel, and S. Munder, "Vision-Based Pedestrian Detection: The PROTECTOR System", IEEE Intelligent Vehicles Symposium, pp. 13-18, 2004.
- [2] P. Viola, M. Jones, and D. Snow, "Detecting Pedestrians Using Patterns of Motion and Appearance", IEEE International Conference on Computer Vision, pp. 153-161, 2005.
- [3] G. Monteiro, P. Peixoto, and U. Nunes, "Vision-based pedestrian detection using Haar-like features", Robotica, pp. 16-20, 2006.
- [4] P. Dollar and Z. Tu, P. Perona, S. Belongie, "Integral channel features", British Machine Vision Conference, pp. 1-11, 2009.
- [5] Dalal, N and Triggs, B, "Histograms of oriented gradients for human detection". Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 886-893, 2005.
- [6] P. Dollar and S. Belongie, P. Perona, "The Fastest Pedestrian Detector in the West", British Machine Vision Conference, 2010.
- [7] R. Benenson and M. Mathias, R. Timofte, L. Van Gool, "Pedestrian detection at 100 frames per second". Computer Vision and Pattern Recognition (CVPR), pp. 2903-2910, 2012.
- [8] G. Xu, X. Wu, L. Liu, and Z. Wu, "Real-time Pedestrian Detection Based on Edge Factor and Histogram of Oriented Gradient", International Conference on Information and Automation (ICIA), pp. 384-389, 2011.
- [9] W. Xing, Y. Zhao, R. Cheng, J. Xu, S. Lv, X. Wang, "Fast Pedestrian Detection Based on Haar Pre-Detection", International Journal of Computer and Communication Engineering, Vol. 1, pp. 207-209, 2012.
- [10] A. Broggi, M. Bertozzi, A. Fascioli, and M. Sechi, "Shape-based Pedestrian Detection", IEEE Intelligent Vehicles Symposium 2000, pp. 215-220, 2000.
- [11] T. Kancharla, P. Kharade, S. Gindi, K. Kutty, and V. Vaidya, "Edge based Segmentation for Pedestrian Detection using NIR Camera", 2011 International Conference on Image Information Processing, pp. 1-6, 2011.
- [12] A. Mazoul, K. Zebbara, and M. Ansari, " Street crossing pedestrian detection based on edge curves motion", International Journal of Computer Applications, vol. 41, pp. 570-575,

2007.

- [13] H. Kataoka, Y. Aoki, "Symmetrical Judgment and Improvement of CoHOG Feature Descriptor for Pedestrian Detection", MVA2011, pp. 13-15, 2011.
- [14] Sang-Hun Kim, Dong-Gon Yoo, and Young-Hwan Kim, "High Performance Pedestrian Detection System Using A Cascade Algorithm Structure", IEEK SoC Conference, pp. 91-94, 2011.
- [15] A. Cosma, R. Brehar, and S. Nedeveschi, "Part-based pedestrian detection using HoG features and vertical symmetry", Intelligent Computer Communication and Processing (ICCP), pp. 229-236, 2012.
- [16] G. Lie, W. Rongben, J. Lisheng, and Z. Mingheng, "STUDY ON PEDESTRIAN DETECTION AHEAD OF VEHICLE BASED ON MACHINE VISION", International Conference on Transportation Engineering ASCE, pp. 570-575, 2007.
- [17] G. Lie, W. Rongben, J. Lisheng, L. Linhui, and Y. Lu, "Algorithm Study for Pedestrian Detection Based on Monocular Vision", Vehicular Electronics and Safety, pp. 83-87, 2006.
- [18] D. Cheda, D. Ponsa, and M. Lopez, "Pedestrian candidates generation using monocular cues", Intelligent Vehicles Symposium(IV), pp. 7-12, 2012.
- [19] Y. Fang, K. Yamada, Y. Ninomiya, K. Horn, and I. Masaki, "A Shape-independent method for pedestrian detection with far-infrared images", Vehicular Technology, pp. 1679-1697, 2004.
- [20] Guilherme V. Carvalho, Lailson B. Moraes, "A weighted image reconstruction based on PCA for pedestrian Detection", Neural Networks (IJCNN), The 2011 International Joint Conference on, pp. 2005-2011, 2011.
- [21] INRIA Person Dataset, <http://pascal.inrialpes.fr/data/human/>
- [22] SVMlight, <http://svmlight.joachims.org/>
- [23] CAVIAR Test Case Scenarios, <http://homepages.inf.ed.ac.uk/rbf/CAVIARDATAI/>

— 저 자 소 개 —



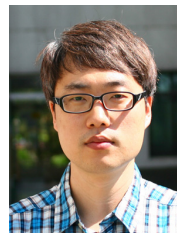
김 철 문(정회원)

2006년 숭실대학교 정보통신전자공학부 학사 졸업

2013년 한양대학교 전자컴퓨터통신공학과 석사졸업

2010년~현재 피엘케이 테크놀로지 전임연구원

<주관심분야 : 지능형 하이빔, 보행자 인식, 패턴인식, 지능형 자동차 등>



백 열 민(정회원)

2005년 한양대학교 전자전기컴퓨터공학부 학사 졸업

2007년 한양대학교 전자통신컴퓨터공학과 석사졸업

2013년 한양대학교 전자컴퓨터통신공학과 박사 졸업

2013년~현재 현대모비스 영상센서설계팀 선임연구원

<주관심분야: 영상처리, 패턴인식, 지능형 자동차 등>



김 회 울(정회원)

1980년 한양대학교 전자공학과 학사 졸업

1983년 Pennsylvania State University 전기공학과 석사 졸업

1989년 Purdue University 전기공학과 박사 졸업

1989년~1994년 University of Texas 조교수

1994년~현재 한양대학교 융합전자공학부 정교수

<주관심분야 : 영상처리, 컴퓨터비전, 패턴인식, 머신비전, MPEG-7, 지능형 자동차 등>