

실시간 보행자 검출을 위한 HOG 연산 알고리즘 고속화 방법

이윤구*, 이재흥*
*한밭대학교 컴퓨터공학과
e-mail:lygoo89@naver.com

A Speed-up Method of HOG Computation Algorithm for Realtime Pedestrian Detection

Yun-Gu Lee*, Jae-Heung Lee*

*Dept of Computer Engineering, Hanbat National University

요 약

보행자 검출과정은 특징추출, 추출된 특징을 기반으로 한 학습과정, 그리고 학습된 데이터를 기반으로 한 분류과정으로 나눌 수 있다. 이들 중 연산시간이 가장 오래 걸리는 특징추출과정이다. 기존의 HOG 특징 추출은 하나의 학습 샘플 이미지에 대하여 많은 픽셀 연산이 필요하기 때문에 많은 시간이 소요되었다. 본 논문에서는 실시간 스트리밍 환경에서 이전 프레임의 HOG 특징 검출정보를 분석하여 다음 프레임에서 보행자가 존재 할 가능성이 높은 부분에 대해서만 특징을 추출한다. 이 방법으로 기존의 연구와 비교하여 인식성능에는 거의 영향을 주지 않고 인식 속도를 향상할 수 있다.

1. 서론

오늘날의 물체 검출 및 인식 추적 관련 기술은 계속해서 연구되고 있다. 보행자 검출 관련 연구 또한 활발하게 진행되고 있다. 보행자를 검출하는 것은 상당히 어려운 문제이다. 사람마다 키나 몸무게 등 외형적인 모습이 다르고 입고 있는 옷들도 다르고 주변 환경(예:조명, 빛)이 다르기 때문이다. 기존의 많은 연구에서는 얼굴 및 보행자검출을 위하여 Viola-Jones의 물체 검출 방법[1]을 많이 사용하였다. 하지만 검출 성능이 더욱 우수한 HOG(Histogram of Oriented Gradient) 특징을 이용한 보행자 검출 방법[2]이 제안되었다. HOG 특징은 연산량이 많다는 단점이 있다. 이러한 단점을 해결하기 위하여 기존의 많은 연구들은 HOG기반 학습기와 분류기의 속도와 검출율을 개선하기 위한 방법을 연구해왔다.

본 논문에서는 HOG기반 학습기와 분류기의 속도를 향상시키기 위하여 하나의 샘플이미지 당 HOG 특징 추출량을 줄이는 방법을 제시한다. 비디오 환경에서 이전 프레임의 HOG 특징 검출정보를 분석하여 다음 프레임에서 보행자가 분포할 가능성이 높은 영역에 대해서만 HOG 특징을 추출한다. 이 방법으로 기존의 연구와 비교하여 인식성능에는 거의 영향을 주지 않고 인식 속도를 향상할 수 있다.

2. 관련 연구 및 관련 기술

2.1 HOG

HOG는 현재 보행자 검출에서 가장 많이 사용되고 있는 특징 중에 하나이다. HOG 특징을 이용하여 영상의 지역적인 기울기 크기와 방향 정보를 얻을 수 있다. 이러한 정보를 이용하여 보행자 뿐만 아니라 다른 여러 물체 검출에도 이용할 수 있다.

HOG는 기울기의 크기와 방향을 가지는 히스토그램이다. HOG 특징은 기본적으로 64x128 크기의 샘플 이미지에 대해서 계산한다. 기울기의 크기와 방향을 계산하는 방법은 아래와 같다. 가로방향 픽셀의 변화량을 d_x 라고 하고, 세로방향 픽셀의 d_y 라고 하면 아래의 (1), (2)와 같이 나타낼 수 있다.

$$d_x(x, y) = I(x+1, y) - I(x-1, y) \quad (1)$$

$$d_y(x, y) = I(x, y+1) - I(x, y-1) \quad (2)$$

여기서 $I(x, y)$ 값은 (x, y) 에서의 픽셀 값의 크기를 나타내고, 기울기의 크기 $G(x, y)$ 와 기울기의 방향 $\theta(x, y)$ 는 아래의 (3), (4)와 같이 구할 수 있다.

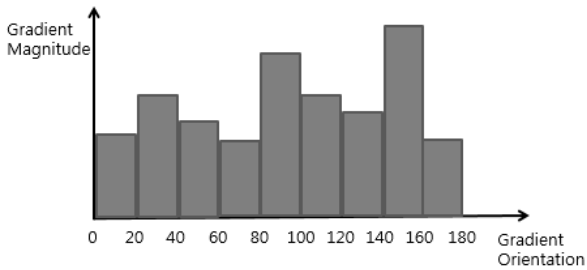
$$G(x, y) = \sqrt{d_x(x, y)^2 + d_y(x, y)^2} \quad (3)$$

$$\theta(x, y) = \tan^{-1} \frac{d_y(x, y)}{d_x(x, y)} \quad (4)$$

위의 기울기의 크기와 기울기의 방향을 모든 64x128 샘플 이미지에 대해서 계산을 한 후 히스토그램 생성 과정을 거치게 된다. 히스토그램의 생성은 cell단위로 이루어진다.

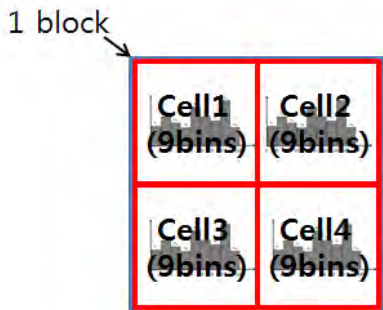
*본 연구는 교육부와 한국연구재단의 지역혁신인력양성 사업으로 수행된 연구결과임 (No. 2012H1B8A2026119)."

하나의 cell은 8x8픽셀을 나타낸다. 먼저, 기울기의 방향을 0° ~ 20° 에서 160° ~ 180° 까지 분류하여 각각의 기울기 방향에 해당하는 기울기의 크기 값을 누적한다. 이 과정을 전체 샘플 이미지에 대해서 수행하게 되면 아래의 그림 1과 같은 히스토그램이 총 128(8x16)개 생성된다.



(그림 1) 하나의 cell에 대한 기울기 히스토그램

히스토그램의 정규화 과정은 block 단위로 진행된다. 하나의 block은 2x2셀, 즉 16x16픽셀로 이루어진다. 하나의 block에는 총 36개(4cells x 9bins)의 특징이 생성되게 된다. 그리고 block들은 하나의 cell 단위로 겹치면서 이동함으로써 정규화를 수행하게 된다. 그렇기 때문에 64x128의 샘플이미지에 대해서 105(7x15)개의 block에 대해 정규화가 진행된다. 최종적으로 3780개의 특징이 생성된다.



(그림 2) 하나의 block을 구성하는 4개의 cell

3.2 Integral Histogram

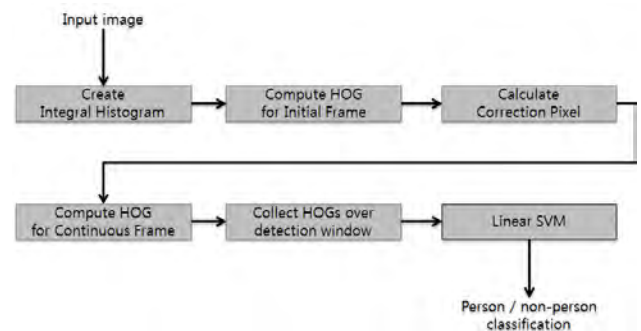
보행자 검출은 크게 단일 Single Window 방법과 Part-based 방법으로 연구되고 있다. HOG의 경우에는 Single Window 방식의 대표적인 특징이다. HOG는 검출률을 향상시키는 연구와 검출의 속도를 향상시키기 위한 연구가 함께 진행되고 있다. 이에 따라 최초의 HOG특징을 제안한 Dalal-Triggs의 방법을 개선하기 위한 연구들이 많이 진행되어왔다. 대표적으로 Qiang Zhu et al.[3]은 속도향상을 위해서 HOG특징을 계산하는 데에 Integral Histogram을 사용하였다. Integral Histogram은 Integral Image와 유사하다. 64x128의 샘플이미지의 모든 픽셀에 대해서 9개의 방향성을 가지는 기울기 크기 값으로 분류한다. 이후 그림 2와 같이 9개의 Integral Image를 생성한다. 생성된 9개의 Integral Image를 Integral Histogram이라고 한다. I

ntegral Histogram을 이용하여 결과적으로 기울기 값을 얻기 위해 총 4회의 데이터 참조로 가능하다. 초기에 Integral Histogram을 생성하는 데에 다소 많은 연산 시간이 필요하지만, 이후 반복된 연산을 줄임으로써 전체적인 연산 속도는 향상된다.

3. 본 논문의 실시간 보행자 검출을 위한 방법

본 논문에서 실시간 보행자 검출을 위해서 두 가지의 방법을 적용하였다. 먼저, F. Porikli et al.[4]에서 제안한 Integral Histogram을 적용하였다. 2장에서 설명한 것과 같이 Integral Histogram을 이용하여 적은 데이터 참조로 HOG특징을 계산한다. 최초의 전체 이미지에 대한 Integral Histogram만 계산이 되면 빠른 속도로 HOG특징을 계산할 수가 있다.

두 번째로 실시간 스트리밍 영상에 대해서 이전 프레임에서의 HOG특징 검출정보를 이용하여 다음 프레임의 HOG특징을 계산한다. 먼저, 실시간 스트리밍 영상에서 첫 프레임에 대한 HOG특징을 계산한다. 계산된 HOG특징이 분포하는 영역의 좌표 정보를 저장한다. 이후 다음 프레임에서 첫 프레임에 대한 HOG특징이 분포하는 영역에서 상하좌우로 보정 픽셀 값(본 논문에서는 32픽셀로 적용)이 떨어진 영역에 대해서만 HOG 특징을 계산한다. 이어지는 프레임에서도 동일한 방법으로 바로 이전 프레임의 HOG 특징이 분포하는 영역을 참조하여 특정 영역에 대해서만 HOG특징을 계산한다. 전체적인 보행자검출 흐름도를 그림 3에 나타낸다.



(그림 3) 실시간 보행자 검출 흐름도

4. 실험 및 고찰

본 논문에서 제안하는 방법은 기존의 Dalal-Triggs의 HOG 보행자 검출 방법에 적용하여 성능 평가를 진행하였다. 3.3GHz Quad Core CPU가 내장된 PC에서 성능을 평가하였다. 또한 320x240 크기의 실시간 스트리밍 영상이 들어오는 환경에서 테스트를 진행하였다. 기존의 Dalal-Triggs 방법의 경우 전체 픽셀에 대해서 HOG 특징을 계산하고, 본 논문에서 제안하는 방법은 Integral Histogram을 이용하여 픽셀에 접근하는 횟수를 감소시켜 성능을 높였

다. 또한 이전 프레임의 HOG특징 검출정보를 활용하여 특징을 계산하는 영역을 감소시켜 기존의 Dalal-Triggs의 방법보다 향상된 결과를 얻었다. 그림4와 같이 Dalal-Triggs의 방법이 한 장의 이미지에서 보행자를 검출할 시 100 ms가 걸린 반면 본 논문에서 제안하는 방법을 이용하면 보정 픽셀 값에 따라 속도가 최대 15배까지 빨라지는 것을 확인 할 수 있다.

Correction Pixel	24	28	32	36	40	48
Dalal-Triggs Method	100	100	100	100	100	100
Proposed Method	33	27	23	18	14	6

(그림 4) 보정 픽셀 값에 따른 속도 비교

5. 결론

본 논문에서는 실시간 스트리밍 영상이 들어오는 환경에서 보행자 검출을 위한 방법을 제안하였다. 기존의 방법과 비교하여 320x240 크기의 이미지에서 64x128 서브윈도우를 가지고 보정 픽셀 값을 32픽셀로 하였을 때 약 4배 정도의 속도 향상을 보였다. 그림 4와 같이 보정 픽셀 값에 따라 속도향상은 약 3배에서 15배까지 나타났다.

본 논문에서 제안하는 방법은 다중 보행자 검출을 위한 방법이 더욱 더 자세히 연구될 필요가 있다. 또한 HOG특징을 이용한 보행자 검출 뿐만 아니라 얼굴 검출, 차량검출 등 다른 물체 검출에도 다양하게 이루어지도록 최적화 과정이 필요하다.

참고문헌

[1] Paul Viola and Michael J. Jones, "Robust Real-Time Face Detection," International Journal of Computer Vision, vol.57, no.2, pp.137-154, 2004

[2] Navneet Dalal and Bill Triggs, "Histogram of Oriented Gradients for Human Detection," Proceedings of the IEEE, Computer Society Conference on Computer Vision and Pattern Recognition, vol.1, pp.886-893, 2005.

[3] Qiang Zhu, Shi Avidan, Mei-Chen Yeh, and Kwang-Ting Cheng, "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients," Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol.2, pp.1491-1498, 2006.

[4] F. Porikli, "Integral histogram: A fast way to extract histograms in cartesian spaces," Conference on Computer Vision and Pattern Recognition (CVPR), 2005