

병렬처리 그래픽 프로세서와 범용 프로세서에서의 보행자 검출 처리 속도 비교

박장식*

Comparison Speed of Pedestrian Detection with Parallel Processing Graphic Processor and General Purpose Processor

Jang-Sik Park*

요 약

영상기반 객체 검출은 지능형 CCTV 시스템을 구현하는데 있어 기본적인 기술이다. 객체 검출을 위하여 다양한 특징점과 알고리즘이 개발되었으나, 성능에 비례하여 계산량이 많다. 본 논문에서는 GPU와 CPU를 활용하여 객체 검출 알고리즘의 성능을 비교하였다. 일반적으로 보행자 검출에 널리 사용되고 있는 Adaboost 알고리즘과 SVM 알고리즘을 각각 CPU와 GPU에 맞도록 구현하고 동일 영상에 대하여 검출 처리 속도를 비교하였다. Adaboost 알고리즘과 SVM 알고리즘에 대하여 처리 속도를 비교한 결과 GPU가 CPU에 비하여 약 4 배 정도 빠른 처리를 할 수 있음을 확인하였다.

ABSTRACT

Video based object detection is basic technology of implementing smart CCTV system. Various features and algorithms are developed to detect object, however computations of them increase with the performance. In this paper, performances of object detection algorithms with GPU and CPU are compared. Adaboost and SVM algorithm which are widely used to detect pedestrian detection are implemented with CPU and GPU, and speeds of detection processing are compared for the same video. As results of frame rate comparison of Adaboost and SVM algorithm, it is shown that the frame rate with GPU is faster than CPU.

키워드

객체 검출, Adaboost, SVM, 병렬처리, GPU
Object Detection, Adaboost, SVM, Parallel processing, GPU

1. 서 론

CCTV(Closed Circuit TV)는 방법, 방재, 시설물 관리 그리고 교통체계 관리 등에 널리 적용되고 있다. 방법 및 방재를 위한 CCTV관제센터에는 수 백대의 카메라로부터 영상이 전송되어 관제하는데 한계가 있

다. 따라서 최근에는 영상처리, 영상인식 기술을 활용하여 범죄 또는 재난 상황을 인식하는 영상기반 지능형 CCTV 기술들이 개발되고 있다[1-4].

영상기반 지능형 CCTV 관제시스템을 구현하기 위한 요소기술로 객체 검출, 추적 및 상황인식 기술이 필요하다. 객체 검출은 영상기반 상황인식을 위한 기

* 경성대학교 전자공학과(jsipark@ks.ac.kr)
접수일자 : 2014. 11. 12

심사(수정)일자 : 2015. 01. 16

게재 확정일자 : 2015. 02. 09

본적인 기술이다. 객체 검출은 배경추정 기법을 이용하는 방법과 특징점을 활용한 학습 알고리즘을 이용하는 방법이 있다. 배경추정 알고리즘은 학습 알고리즘을 이용하는 방법[5], [6]으로 계산량은 적지만 사람이 움직이지 않는 경우에는 객체 검출 성능이 저하된다. 객체의 특징점을 활용한 학습 알고리즘으로 객체를 검출하기 위해서는 객체의 특징을 잘 표현하는 특징점(feature)을 추출하고, 특징점을 이용하여 검출하고자 하는 대상에 적합한 학습 알고리즘(learning algorithm)이 필요하다. 객체 검출을 위하여 다양한 특징점과 학습 알고리즘이 개발되었다. CCTV 관제에 있어 검출해야 하는 대상은 보행자와 차량이다. 보행자 검출에는 유사 Haar, HOG(Histogram of Oriented Gradient) 그리고 LBP(Local Bit Pattern) 특징점[7]이 주로 활용되고, 학습 알고리즘으로는 Adaboost[8]와 SVM(Support Vector Machine)[9] 알고리즘이 적용된다. 학습 알고리즘은 잡음이 있거나 객체가 겹치는 상황에서도 우수한 성능을 나타내지만 계산량이 다소 많다.

본 논문에서는 GPU를 활용하여 지능형 CCTV 관제시스템에서의 보행자 검출 속도를 개선하는 것에 관한 것이다. 보행자 검출에 널리 적용되고 있는 Adaboost 와 SVM 알고리즘을 CPU와 GPU 상에서 구현하여 검출 처리 속도를 비교한다. 엔비디아사(NVIDIA)의 CUDA를 활용하여 각 알고리즘 GPU 상에서 구현하여 두 알고리즘을 처리 속도를 CPU와 비교하였다. Adaboost 알고리즘이 SVM 알고리즘에 비하여 다소 처리 속도가 빠르고, GPU를 활용한 검출이 CPU에 비하여 4 배 정도 빠른 것을 확인하였다.

II. 영상기반 객체 검출

영상기반 객체 검출을 위하여 다양한 특징점(feature)과 학습 알고리즘(learning algorithm)이 제안되고 있다.

최근까지 주로 연구되고 있는 특징점들은 유사 Haar 특징점, HOG, LBP 등이 있다. 언급한 특징점과 조합하여 최근까지 적용되고 있는 학습 알고리즘은 Adaboost 알고리즘, SVM 알고리즘이 주로 사용되고 있다.

2.1. 영상기반 객체 검출 특징점

영상기반 객체 검출을 위한 특징점의 선택은 하는 것은 객체 검출 성능에 많은 영향을 준다. 그림 1은 기본형 유사 Haar 특징점을 나타낸 것이다. (a), (b), (c)는 각각 경계선 특징점, 선 특징점 그리고 중심 특징점이다. 그림 2는 유사 Haar 특징점을 활용하여 객체를 검출하는 원리를 나타낸 것으로 보행자의 각 움직임에 대하여 유사도가 높은 특징점을 나타낸 것이다.

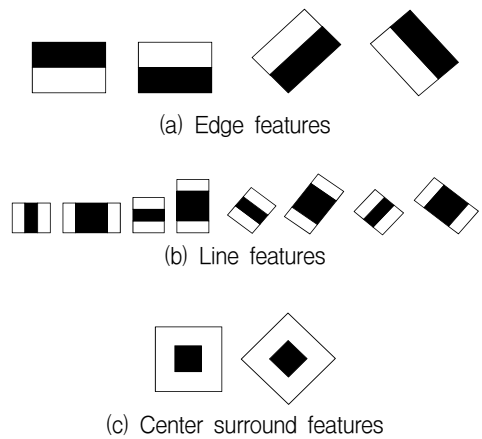


그림 1. 확장된 유사 Haar 특징점
Fig. 1 Sets of extended Haar-like features.

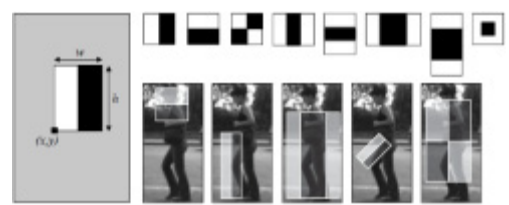


그림 2. 보행자를 포함하는 영역에서 높은 응답을 제공하는 유사 Haar 특징점
Fig. 2 Examples of Haar-like features that gives high response in regions containing pedestrians.

그림 3은 HOG 특징점을 얻는 과정을 나타낸 것이다. (a)는 야간에 촬영된 보행자 영상이며, (b)는 기울기를 검출한 결과이고 (c)는 기울기 방향을 표시한 것이다. (d)는 보행자 영상을 일정한 간격으로 영역을 설정하고 (e)는 각 영역에 대한 기울기 방향의 히스토그램을 나타낸 것이다. 보행자에 대한 기울기 방향 히스토그램 즉, HOG 특징을 비교하여 보행자를 검출하

는데 활용한다. 일반적으로 보행자 검출에 효과적인 특징점으로 알려져 있다.

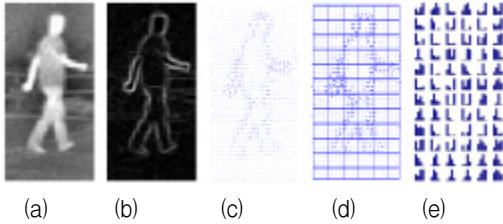


그림 3. HOG를 이용한 영상 특징 묘사. (a) 원영상, (b) 정규화 기울기, (c) 기울기 방향, (d) 셀 분포, (e) 히스토그램 계산

Fig. 3 Image characterization using HOG. (a) original image, (b) gradient norm, (c) gradient orientation, (d) cell splitting, (e) histogram computation.

그림 4는 LBP 특징점을 구하는 과정을 나타낸 것이다. 특정 영역에서 중심 화소를 기준으로 밝기값이 큰 화소는 1로 작은 화소는 0으로 설정하여 반시계 방향으로 생성된 비트 패턴을 특징점으로 활용한다.

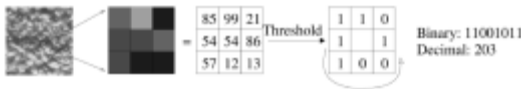


그림 4. 기본적인 LBP 연산
Fig. 4 The basic LBP operator.

2.2. 학습 알고리즘

객체의 형상 변화와 겹침 또는 잡음이 있는 상황에서 객체를 검출하기 위하여 학습 알고리즘들이 제안되고 있다.

표 1은 보행자와 얼굴 검출에 활용되고 있는 Adaboost 알고리즘이다. 3 단계에서 구한 약분류기들을 조합하여 4 단계에서 강분류기를 형성하여 특정 영역에 대한 유사 Haar 특징점 또는 HOG 특징점을 비교하여 보행자를 검출한다.

표 1. Adaboost 훈련
Table 1. Adaboost training

Given example images $(x_1, y_1), \dots, (x_n, y_n)$ where $y_i \in \{0, 1\}$ for negative and positive examples respectively.

Initialize weights $w_{1,i} = 1/2m, 1/2l$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.

For $t = 1, \dots, T$:

Step 1. Normalize the weights,

$$w_{t,i} = w_{t,i} / \sum_{j=1}^n w_{t,j}$$

so that w_t is probability distribution.

Step 2. For each feature, j , train a classifier h_j , which is restricted to using a single feature. The error is evaluated with respect to w_t ,

$$\epsilon_j = \sum_{i=1}^n w_{t,i} |h_j(x_i) - y_i|$$

Step 3. Choose the classifier h_t , with the lowest error ϵ_t .

Step 4. Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1 - e_i}$$

where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \epsilon_t / (1 - \epsilon_t)$.

Step 5. The final strong classifier is:

$$h(x) = \begin{cases} 1, & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0, & \text{otherwise} \end{cases}$$

where $\alpha_t = \log \frac{1}{\beta_t}$

표 2는 SVM 알고리즘을 나타낸 것이다.

표 2. SVM 훈련
Table 2. SVM training

For a given training set $(x_1, y_1), \dots, (x_n, y_n)$ where $y_i \in \{-1, 1\}$

$W \leftarrow 0$

repeat

$$(w, \xi) \leftarrow \operatorname{argmin}_{w, \xi \geq 0} \frac{1}{2} w^T w + C \xi$$

$$\text{s.t. } \forall c \in W: \frac{1}{n} w^T \sum_{i=1}^n c_i y_i x_i \geq \frac{1}{n} \sum_{i=1}^n c_i - \xi$$

for $i = 1, \dots, n$ do

```

     $c_i \leftarrow \begin{cases} 1 & y_i(\mathbf{w}^T \mathbf{x}_i) < 1 \\ 0 & \text{otherwise} \end{cases}$ 
  end for
   $\mathcal{W} \leftarrow \mathcal{W} \cup \{c\}$ 
until  $\frac{1}{n} \sum_{i=1}^n c_i - \frac{1}{n} \sum_{i=1}^n c_i y_i(\mathbf{w}^T \mathbf{x}_i) \leq \xi + \epsilon$ 
return  $(\mathbf{w}, \xi)$ 

```

III. GPU 임베디드시스템

3.1. GPU 테그라 K1

그래픽 또는 영상 처리를 위한 CPU의 부하를 줄이기 위하여 그래픽 프로세서인 GPU 사용되고 있다. 최근 멀티코어 CPU의 보급과 GPU의 성능 향상에 따라 개인용 컴퓨터에서 병렬처리가 가능하게 되었다. GPU는 엔비디아사와 AMD 등이 활발하게 연구 개발을 진행하고 있다.

엔비디아사의 테그라 K1(Tegra K1) SoC를 사용한 모바일 임베디드시스템을 활용하여 영상기반 객체 검출 시스템을 구현한다. 테그라 K1은 192개의 GPU 코어를 가지고 있어 그래픽 처리 기능이 우수하고 전력 효율이 높다. 테그라 K1은 엔비디아의 Kepler 아키텍처(architecture)를 로 구현되었으며, DirectX11과 OpenGL 4.4를 지원하여 PC 수준의 게임에 그래픽 처리를 지원한다. 레노보 ThinkVision, 샤오미 MI 패드, 에이스 크롬북, HP 크롬북에 적용되고 있다[10].

테그라 K1은 CPU, GPU, ISP(Image Signal Processor)가 함께 내장되어 있다. CPU는 엔비디아의 2.32GHz ARM 4코어 Cortex-A15이며, GPU는 Kepler GK20a로써 192개의 SM3.2 CUDA 코어를 포함한다.

테그라 K1에 적용된 Kepler GPU의 아키텍처는 고급 시스템에 사용되는 Kepler GPU 아키텍처와 거의 동일하지만, 모바일의 특성상 전력 효율을 유지하고, 모바일 시스템에 맞는 최적화가 적용되었다. 일반적으로 데스크탑, 워크스테이션 및 슈퍼 컴퓨터에 적용되는 Kepler GPU는 최대 2,880 개의 단정도 부동 소수점 CUDA 코어를 포함하고 수백 와트의 전력을 소비한다. 테그라 K1의 Kepler GPU는 192 개의 CUDA 코어로 구성되어 2 W 미만을 소비한다.

Kepler 아키텍처는 GPC(Graphics Processing Clu-

ster), SMX(Streaming Multiprocessor) 및 메모리 컨트롤러로 구성된다. Kepler 기반 테스트용 GeForce GTX 680 GPU는 GPC 4 개, SMX 8 개, 메모리 컨트롤러 4 개로 구성된다. 테그라 K1의 Kepler GPU는 GPC 1 개, SMX 1 개 그리고 메모리 인터페이스로 구성된다. 테그라 K1 GPU에는 4 개의 ROP가 포함되어 있으며, ROP와 메모리 인터페이스 사이에 128 KB L2 캐시가 있다.

테그라 K1의 Kepler GPU는 OpenGL 4.4 사양과 NVIDIA CUDA 6, 하드웨어 특징 레벨 11으로 DirectX 11.2 API를 지원함으로써 모바일 그래픽 처리 기능을 제공한다.

카메라 입력 영상에 대한 처리에 있어 테그라 K1에는 최대 100 메가픽셀 센서를 지원하는 듀얼 ISP(dual Image Signal Processor) 코어를 포함하고 있다. 코어에는 4096 개의 동시 초점을 처리하고 초당 1.2 G의 화소를 처리할 수 있다. 각 ISP 코어는 공간 변수 노이즈 감소, 다중 불량 화소 수정, 다운 스케일링, 색공간 변환기, 영역 프로세서 등의 기능을 통하여 화질을 보장하고 잡음이 적은 영상을 제공함으로써 영상을 기반으로 하는 영상처리 및 컴퓨터 비전 알고리즘의 성능을 향상시킬 수 있다. ISP는 최대 4,096 개의 초점을 사용하여 자동 포커스, 포커스 타임, 저조명 포커스 등의 포커스 기능을 제공한다.

3.2. CUDA 아키텍처

GPU를 단순히 화면 처리뿐만 아니라 병렬처리 계산에 활용하는 방법으로 GPGPU(General Purpose Computing on GPU)가 있다. GPGPU는 GPU를 개발하는 NVIDIA사의 CUDA(Computer Unified Device Architecture)가 있으며, AMD사의 ATI Stream 등이 있다. 또한 CPU, GPU 그리고 DSP 이종의 프로세서로 구성된 하드웨어 플랫폼을 효율적으로 이용하고 플랫폼에 의존하지 않는 응용 프로그램을 개발하기 위한 플랫폼으로 애플사(Apple)에 의하여 제공되는 개방 표준 규격인 OpenCL(Open Computer Language)가 있다.

본 논문에서 영상기반 객체 검출을 위하여 NVIDIA의 테그라 K1 SoC를 사용하기 때문에 NVIDIA의 전용 플랫폼인 CUDA를 활용한다. CUDA는 GPU에서 수행하는 알고리즘을 C 언어 등의 산업

표준 언어를 사용하여 작성 및 개발할 수 있도록 하는 GPGPU 기술이다. CUDA는 CUDA GPU내부의 명령어 세트와 대용량 병렬처리 메모리 접근(Memory Access)을 지원한다.

IV. 실험 결과 및 검토

그래픽병렬처리 SoC 테크라 K1을 탑재한 임베디드 시스템을 활용하여 Adaboost 알고리즘과 SVM 알고리즘에 대하여 보행자 검출 처리 속도를 비교 분석하였다. GPGPU로 CUDA 6.0을 사용하여 객체 검출 알고리즘을 컴파일하였다. 실험에 사용한 동영상의 해상도는 720*480 이다.



(a)



(b)

그림 5. CPU와 GPU에서 Adaboost 알고리즘을 이용한 검출 결과 (a) CPU 사용 결과 (b) GPU 사용 결과
Fig. 5 Detection results of Adaboost algorithm with CPU and GPU. (a) A result with CPU (b) A result with GPU

그림 5와 6은 각각 유사 Haar 특징점을 이용한 Adaboost 알고리즘과 HoG 특징점을 이용한 SVM 알고리즘에 대하여 CPU와 GPU를 이용하여 검출한 결과이다. 각 알고리즘을 CPU 또는 GPU를 이용하여 구현한 보행자 검출 결과, 보행자를 대체로 잘 검출하는 것을 확인할 수 있다.



(a)



(b)

그림 6. CPU와 GPU에서 SVM 알고리즘을 이용한 검출 결과 (a) CPU 사용 결과 (b) GPU 사용 결과
Fig. 6. Detection results of SVM algorithm with CPU and GPU (a) A result with CPU (b) A result with GPU

그림 7은 유사 Haar 특징점을 활용한 Adaboost 알고리즘과 HOG 특징을 활용한 SVM 알고리즘에 대하여 CPU와 GPU를 이용하여 검출 처리하는 속도를 비교한 것이다. 긴 점선과 실선은 유사 Haar 특징점에 대하여 CPU와 GPU를 활용하여 검출 처리하는 속도를 나타내고, 일점 쇄선과 짧은 점선은 HOG 특징점에 대하여 CPU와 GPU로 처리하는 속도를 나타낸다. 두 특징점을 활용한 검출 처리 속도를 비교하면 CPU에 비하여 GPU가 약 4배 정도 빠르게 처리하는 것을 확인할 수 있다. 그리고, 유사 Haar 특징점과 HOG

특징점을 이용한 검출 속도를 비교하면 Haar 특징점을 이용한 Adaboost 알고리즘이 HOG 특징점을 이용한 SVM에 비하여 1 프레임 정도 빠르게 처리할 수 있음을 확인할 수 있다.

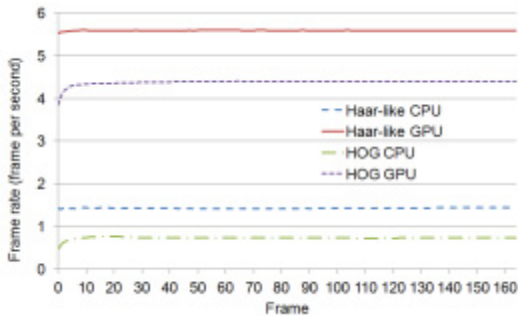


그림 7. CPU와 GPU를 사용한 검출 속도 비교
Fig. 7 Comparison of detection speed with CPU and GPU

V. 결론

본 논문에서는 CPU, GPU 그리고 ISP를 포함하고 있는 테그라 K1 프로세서 임베디드시스템으로 영상 기반 객체 검출 알고리즘의 처리 속도를 비교하였다. 비교 결과 GPU를 활용한 객체 검출 성능이 약 4 배 정도 향상된 것을 확인할 수 있었다. 향후 Deep 학습 알고리즘을 GPU로 구현하고 처리 속도를 비교 검토하고자 한다.

감사의 글

이 논문은 2014년도 경성대학교 학술연구비지원에 의하여 연구되었음.

References

- [1] H-M. Moon and S-B. Pan, "The human identification method in video surveillance system," *J. of The Korean Institute of Information Technology*, vol. 8, no. 5, May 2010, pp. 199-206.
- [2] H-M. Moon and S-B. Pan, "The analysis of de-identification for privacy protection in intelligent video surveillance system," *J. of The Korean Institute of Information Technology*, vol. 9, no. 7, July 2011, pp. 189-200.
- [3] H-T. Kim, G-H. Lee, J-S. Park, and Y-S. Yu, "Vehicle detection in tunnel using Gaussian mixture model and mathematical morphological processing," *J. of The Korean Institute of Electronic Communication Sciences*, vol. 7, no. 5, Oct. 2012, pp. 967-974.
- [4] M-W. Kim, C-M. Oh, D. Aurrahman, Y-G. Ahn, and C-W. Lee, "The virture screen using skin tone and GMM foreground segmentation," *In Proc. Conf. of The Korea Information Processing Society*, vol. 15, no. 1, May 2008, pp. 179-181.
- [5] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," *In Proc. IEEE Computer Vision and Pattern Recognition(CVPR) 1999*, vol. 2, June 1999, pp. 246-252.
- [6] A. Elgammal, D. Harwood, and L. S. Davis, "Non-parametric model for background subtraction," *In Proc. European Conf. on Computer Vision(ECCV 2000)*, vol. 1843, June. 2000, pp. 751-767.
- [7] T. Ahonen, A. Hadid and M. Pietikaninen, "Face description with local binary patterns : application to face recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, Dec. 2006. pp. 2037-2041.
- [8] D. Geronimo, A. D. Sappa, A. Lopez and D. Ponsa, "Pedestrian detection using Adaboost learning of features and vehicle pitch estimation," *In Proc. Int. Conf. Visualization, Imaging and Image Processing(IASTED)*, Palma de Mallorca, Spain, Aug. 2006. pp. 400-405.
- [9] Thorsten Joachims, "Training linear SVMs in linear time," *In Proc. of Int. conf. on Knowledge Discovery and Data Mining(KDD)*, Philadelphia Pennsylvania, Aug. 2006. pp. 217-226.
- [10] "NVIDIA Tegra K1, A New Era in Mobile Computing," *White-paper of NVIDIA*, 2013.

저자 소개



박장식(Jang-Sik Park)

1992년 부산대학교 전자공학과 졸업
(공학사)

1994년 부산대학교 전자공학과 졸업
(공학석사)

1999년 부산대학교 대학원 전자공학과 졸업(공학박사)

1997년 3월~2011년 2월 동의과학대학 전자과 교수

2011년 3월~현재 경성대학교 전자공학과 교수

※ 관심분야 : 신호처리, 음성 및 음향신호처리, 영상처리 및 이해, 임베디드시스템

