

특집논문 (Special Paper)

방송공학회논문지 제20권 제3호, 2015년 5월 (JBE Vol. 20, No. 3, May 2015)

<http://dx.doi.org/10.5909/JBE.2015.20.3.421>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

GPU를 이용한 야간 보행자 검출과 추적 시스템 구현

최 범 준^{a)}, 윤 병 우^{a)}, 송 중 관^{a)}, 박 장 식^{a)†}

Implementation of Pedestrian Detection and Tracking with GPU at Night-time

Beom-Joon Choi^{a)}, Byung-Woo Yoon^{a)}, Jong-Kwan Song^{a)}, and Jangsik Park^{a)†}

요 약

이 논문은 적외선 영상을 이용하여 보행자를 검출하고 추적하는 방법에 관한 것이다. 영상기반 보행 검출 및 추적 처리 속도를 개선하기 위하여 병렬처리언어인 CUDA(Computer Unified Device Architecture)를 활용한다. 보행자 검출은 하프 유사 특징을 기반으로 Adaboost 알고리즘을 적용한다. Adaboost 분류는 적외선 영상으로 제작한 데이터셋을 이용하여 훈련한다. Adaboost 분류기로 보행자를 검출한 후, HSV 히스토그램을 특징점으로 파티클 필터를 이용하여 보행자를 추적하는 방법을 제안한다. 제안하는 검출 및 추적 방법을 Linux 환경에서 소프트웨어를 개발할 수 있는 NVIDIA의 Jetson TK1 개발보드 상에 구현하였다. 이 논문에서는 보행자 검출 및 추적을 CUDA 개발환경인 GPU를 이용하여 병렬처리한 결과를 나타내었다. GPU를 이용한 보행자 검출과 추적 처리 속도가 CPU 처리속도에 비하여 약 6 배 빠른 것을 확인할 수 있다.

Abstract

This paper is about an approach for pedestrian detection and tracking with infrared imagery. We used the CUDA(Computer Unified Device Architecture) that is a parallel processing language in order to improve the speed of video-based pedestrian detection and tracking. The detection phase is performed by Adaboost algorithm based on Haar-like features. Adaboost classifier is trained with datasets generated from infrared images. After detecting the pedestrian with the Adaboost classifier, we proposed a particle filter tracking strategies on HSV histogram feature that exploit adaptively at the same time. The proposed approach is implemented on an NVIDIA Jetson TK1 developer board that is full-featured device ideal for software development within the Linux environment. In this paper, we presented the results of parallel processing with the NVIDIA GPU on the CUDA development environment for detection and tracking of pedestrians. We compared the object detection and tracking processing time for night-time images on both GPU and CPU. The result showed that the detection and tracking speed of the pedestrian with GPU is approximately 6 times faster than that for CPU.

Keyword : Pedestrian Detection, Tracking, Adaboost Algorithm, Particle Filter, GPU

a) 경성대학교 전자공학과(Dept. of Electronic Eng., Kyungseong University)

† Corresponding Author : 박장식(Park Jangsik)

E-mail: jsipark@ks.ac.kr

Tel: +82-51-663-4768

ORCID: <http://orcid.org/0000-0003-1794-7631>

※ This work was supported by the Industrial Strategic technology development program of Ministry of Science, ICT and Future Planning, Korea, (10045260, Development of Context Awareness Monitoring and Search System Based on High Definition Multi-video) and Brain Busan 21 program of Busan City.

※ 이 논문의 연구결과 중 일부는 "IWAIT 2015"에서 발표한 바 있음.

· Manuscript received March 23, 2015; revised May 19, 2015; accepted May 19, 2015.

I. Introduction

Object detection and tracking has been applied in various areas such as the automatic security surveillance systems, human-computer interfaces, smart vehicle systems and so on^[1-3]. Automatic security surveillance system has become an important sector as the crime prevention and social security become important issues. CCTV cameras are used for daytime surveillance system, and infrared cameras are used for nighttime systems. When a person monitors many videos at the same time, the efficiency of surveillance decreases 45% after 12 minutes, and 95% after 22 minutes^[4] respectively. So, the development of intelligence video surveillance systems which can replace the conventional systems is important.

There are many studies on object detection and tracking systems, but the performance of them are affected very much by the changes of weather, lighting environment, rain, color of the objects. The contour and shadow of daytime image can be detected well. But, the feature detection of nighttime image is limited because of the low luminance and high brightness of background^[5,6]. In this paper, we proposed a method which can enhance the features of nighttime objects. After that, we developed the program with the NVIDIA GPU on CUDA development environment and compared the calculation speed of the proposed algorithm between GPU and CPU.

II. Proposition of a pedestrian detection and tracking algorithm

Studies about pedestrian detection and tracking with video images are accomplished by many people, because detecting and tracking of pedestrian is useful for many vision based applications including visual surveillance, human computer interfaces, traffic monitoring system, video compression and

so on. Detecting and Tracking of pedestrians in video sequence is one of the main issues of computer vision. It can be utilized to detect and track pedestrian of auto security monitoring system and smart vehicle system^[1-3].

In recent years, feature-based pedestrian detection algorithm that employs training and classification methods is demonstrated excellent results. Examples of feature-based pedestrian detection techniques include the Adaboost algorithm^[4] and SVM(Support Vector Machine)^[5]. Also many studies about the pedestrian tracking algorithms such as particle filters^[6] and Kalman-filters^[7], have been conducted. However, pedestrian detection and tracking systems suffers from false alarms due to occlusions of human body and dynamic changing of background and especially for night-time environments.

We implement a pedestrian detection and tracking method which uses Adaboost algorithm and particle filter with GPU and compare detection rate and processing speed with CPU platform. We used infrared cameras for detection and tracking of nighttime pedestrians.

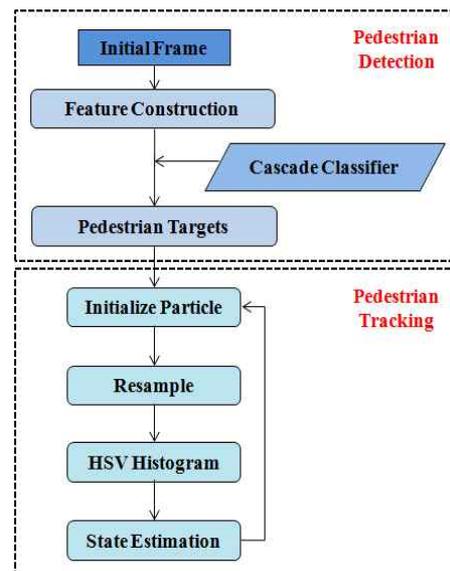


그림 1. 제안하는 알고리즘의 흐름도
Fig. 1. Flow chart of the proposed algorithm

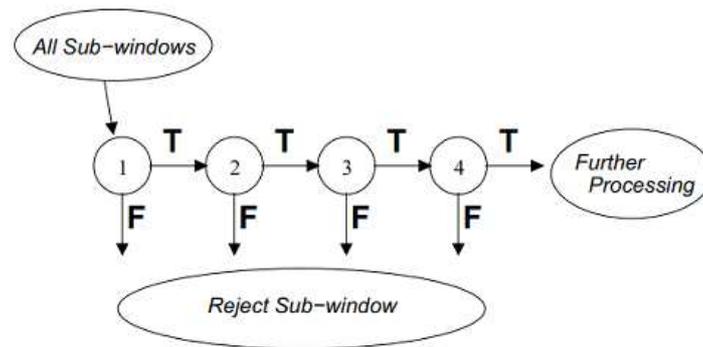


그림 2. Adaboost 알고리즘의 검출 단계
 Fig. 2. The stage of Adaboost algorithm

Fig. 1 shows the flow chart of the proposed algorithm. The detection phase is performed by a cascade classifier with Haar-like feature and Tracking phase is performed by a particle filter with HSV-histogram feature.

We used two kinds of methods for detection and tracking of pedestrians. For the first method, the features of pedestrians are extracted by Adaboost algorithm which uses Haar-like features. And then, the pedestrians and background are separated by the cascade classifier. For the second method, SVM(Support Vector Machine) training algorithm which uses HOG(Histogram of Oriented Gradient), is used for the detection of pedestrians. And then, the pedestrians and background are separated by HOG classifier. At the stage of pedestrians tracking after detection, pedestrians are detected by partial filter which uses the characteristics of HSV histogram.

In this paper, Adaboost algorithm is used to detect pedestrians for surveillance at night. Adaboost algorithm is introduced by Freund and Schapire. It solved many difficulties of boosting algorithms and applied for many applications.

Fig. 2 shows the stage of Adaboost algorithm. This algorithm selects a set of features and train the classifier. The classifier uses a cascade structure to reduce the number of features considered for each sub-window. This approach

reduces computations significantly, and can be applied for real-time video analysis systems. Since boosting is used to select features for classifier, the detection is applicable to additional object classes.

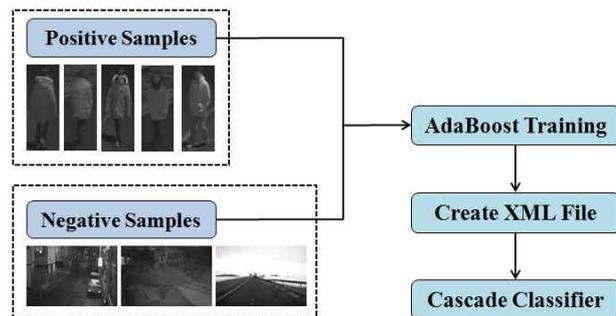


그림 3. Adaboost 알고리즘을 이용한 보행자 검출 과정
 Fig. 3. Pedestrian detection flow with Adaboost algorithm

Fig. 3 shows the flow diagram of pedestrian detection with Adaboost algorithm. Positive and negative samples are generated from infrared images, and they are used for training the Adaboost algorithm. The Haar-like features of pedestrians from the training is stored to a XML file, and the pedestrians and background are separated by cascade classifier.

Particle filter is a typical method for predicting the state of the non-linear system. It is widely used in many fields such as the signal processing, video processing and robot engineering^[8-10]. The objective of a particle filter is to esti-

mate the posterior density of the state variables given the observation variables. The particle filter is designed for a hidden Markov Model, where the system consists of hidden and observable variables

Particle filter is used to approximate the N samples $S = \{(X_t^{(n)}, \pi_t^{(n)}) | t = 1, \dots, N\}$ with weight from given observation probability distribution. Where $X_t^{(n)}$ denotes particles, and $\pi_t^{(n)}$ denotes the weights corresponding to each particle. At estimation step, each selected samples is changing as the propagation process, and $\pi_t^{(n)}$ of obtained samples are calculated. At observation step, observation probability which the similarity between the target and each sample, is measured and each sample is weighted as the result.

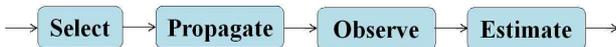


그림 4. 파티클 필터의 추적 단계
Fig. 4. The tracking stage of particle filter

Fig. 4 shows the tracking stage of particle filter, and every stage is performed for every frame of image.

III. Implementation with GPGPU

GPU(Graphics Processing Unit) is a special processor designed to rapidly manipulate and alter memory and accelerate the creation of images in a frame buffer intended for output to a display^[11]. GPU are used in embedded systems, mobile phones, personal computers, workstations, and game consoles. GPU has overwhelming computational speed than the CPU, and there are continuous effort to use it for general purpose. This technique is called as GPGPU(General Purpose GPU). Fig. 5 shows the hardware structures of CPU and GPU^[14].

In this paper, the parallel processing program for detecting and tracking pedestrian at nighttime was developed



그림 5. CPU와 GPU의 하드웨어 구조
Fig. 5. Hardware architecture of CPU and GPU

with CUDA cooperated with OpenCV. CUDA is one of the GPGPU technology which allows developers to program C language more easily and intuitively. CUDA gives developers direct access to the virtual instruction set and memory of the parallel computational elements in CUDA GPUs. GPUs have a parallel throughput architecture that emphasizes executing many concurrent threads slowly, rather than executing a single thread very quickly^[11]. The CUDA platform is accessible to software developers through CUDA-accelerated libraries.

The unit of program execution in CUDA is thread. It gives the function block and grid for the management of multiple threads. Multiple threads become a block, and multiple blocks make a grid. It is called grid-block model. CUDA program consists of CPU code and GPU code. CPU is host and GPU is device. The code which does not need parallel processing is performed on the host, and the code which needs parallel processing is performed on the device. The device code is written in function form, and it is called as kernel. If the kernel is called from the host, the host code stops execution and device code begin execution. At this time, a number of threads are created for parallel processing, and each thread perform the kernel^[14,15]. When compiling, NVCC(NVIDIA C Compiler) separates host and device codes, and general C compiles the host code and NVCC compiles the device code. A grid can be consisted in one dimension, two dimension, or three dimension

of blocks. A block can be consisted in one dimension, two dimension, or three dimension of threads.

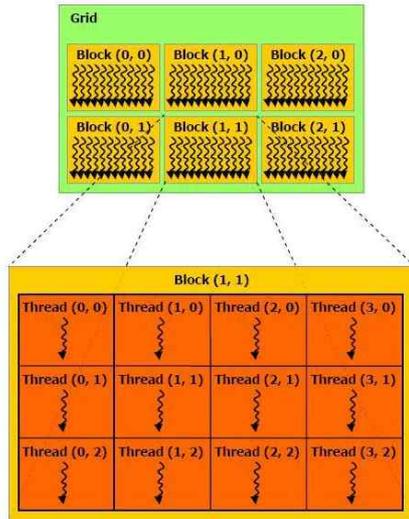


그림 6. CUDA 쓰레드 계층
 Fig. 6. The thread layer of CUDA

Fig. 6 shows the structure of block which is consisted in a number of threads, and grid which is consisted in blocks. Each host and device has separate memory space in CUDA^[14]. To execute the kernel in device, host has to allocate data to the device memory. And host has to move the processed data from the device to the host. We compared the speed of pedestrian detection from GPU parallel

processing and CPU program.

IV. Experiment of pedestrian detection and tracking

Some videos of KISA(Korea Internet and Security Agency) dataset are used to compare the performance of nighttime pedestrian detection and tracking algorithm. KISA dataset was developed for the performance evaluation of intelligent CCTV. It includes scenarios such as loitering and intrusion taken at alley, playground, local facility and cultural property. The resolutions of videos are 1280×720 taken with the HD CCTV cameras. In this paper, the detection and tracking was carried out with videos down sampled 480×320 resolution.

1. Training of Adaboost and SVM algorithm

For the training of Adaboost and SVM algorithm, we used 1,000 positive images and 3,000 negative images. We took the positive and negative training images from the infrared camera for nighttime surveillance.

Fig. 7 shows sample images of positive infrared pedestrian dataset. Figure (a) is for front face images and (b) is for side face images.



(a) Front face images

(b) Side face images

그림 7. 적외선 긍정 영상 예
 Fig. 7. Sample of the positive infrared image dataset



그림 8. 적외선 부정 영상 예
Fig. 8. Sample of infrared images for negative dataset.

Fig. 8 shows sample of infrared images for negative dataset. Negative training images are used to decrease detection error. The images which do not have any relation with the object, is better for the negative training images.

2. The result of pedestrian detection

We performed pedestrian detection experiment for alley, playground, local facilities, and cultural properties. Pedestrian detections were performed by two methods. At first, features are extracted by Adaboost algorithm which

uses Haar-like features, and the pedestrian detection was performed by the cascade classifier. At second, features are extracted by SVM which uses HOG features, and the pedestrian detection was performed by the HOG classifier.

Videos used in the detection simulation are divided into near, middle and far according to CCTV camera position at each place. Near, middle and far videos were captured by the CCTV cameras about 10, 20 and 30 m distance from the position of the event, respectively. Fig. 9 shows some samples of the detection results at alley and playground. In this figure, detected pedestrians are denoted as rec-



그림 9. 보행자 검출 영상 예
Fig. 9. Sample images of pedestrian detection

tangular box. Table 1 shows the result of pedestrian detection for loitering scenario at some places. From the result, we could see that the pedestrians who are in long distance can't be detected. The order of detection rate was near distance, far distance and middle distance. As results of simulations, the detection performances were degraded because Haar-like or HOG features are blurred with distance as shown in fig. 9 (b) and (d).

표 1. 다양한 시나리오에 대한 보행자 검출 결과
 Table 1. The result of pedestrian detection for various scenarios

Location	Number of frame	Detected frame	Detection rate(%)
Alley(near)	2,199	1,590	82.43
Alley(mid)	2,308	1,140	49.39
Alley(far)	2,510	687	27.37
Playground(near)	1,576	1,176	74.62
Playground(mid)	1,725	1,190	68.99
Playground(far)	1,803	842	46.70
Local facility(near)	2,079	1,261	60.65
Cultural property(near)	2,358	1,644	89.40
Cultural property(far)	2,547	427	16.76

3. The result of Pedestrian Tracking

After detecting the pedestrian with the Adaboost classi-

fier, HSV-histogram feature is used for pedestrian tracking under the particle filter framework. Fig. 10 shows some samples of the tracking results for alley scenarios. In this figures, pedestrian tracking are denoted as ellipse box. Point in the ellipse box is number of particle and particle distribution.

4. Comparison of processing speed between GPU and CPU

We used the CUDA that is parallel processing language in order to improve the video-based object detection and tracking processing time. A video of Loitering KISA dataset taken at the alley is used to compare processing speed with GPU and CPU. Fig. 11 shows parallel processing results, where the image (a) is from CPU and image (b) is from GPU. Table 2 is the result of calculation speed comparison between CPU and GPU with Haar-like feature and HOG respectively. From the result, we could see that the processing speed of GPU is 6.4 times faster than that of CPU for cascade classifier with Haar-like feature. For the HOG descriptor, GPU was 5.4 times faster than CPU.



그림 10. 골목길 시나리오에 대한 보행자 추적 결과
 Fig. 10. The result of pedestrian tracking for alley scenario



그림 11. GPU와 CPU의 처리 속도 비교

Fig. 11. The comparison of the processing speed between GPU and CPU

표 2. CPU와 GPU의 계산 속도 비교

Table 2. Comparison of the calculation speed between CPU and GPU

Algorithm	CPU(ms)	GPU(ms)
Haar-like feature	196.94	30.63
HOG Descriptor	389.38	71.89

IV. Conclusions

In this paper, pedestrian detection and tracking from infrared image is performed with Adaboost algorithm and particle filter. After detection and tracking, we compared the pedestrian detection time for night-time image on both GPU and CPU.

The speed of calculation is enhanced with parallel processing based on GPU process. The pedestrians are tracked successfully by optimization of the number, distance distribution, and the size of particles. We performed experiment for various outdoor scenarios by performing Adaboost algorithm and cascade classifier. The detection ratios were 75% for near images, 60% for middle distance image, 30% for far distance images respectively. The calculation speeds of GPU for cascade classifier was 6.4 times faster than that of CPU. For the HOG classifier, GPU was 5.4 times faster than CPU. From the result, we could see that GPU is very useful for realtime video surveillance, be-

cause this application needs lots of computation. In the future study, we will improve the degradation of detection rate according to the distance and the performance evaluation will be carried out with various dataset.

참 고 문 헌 (References)

- [1] D. Geronimo, A. M. Lopez, A. D. Sappa and T. Graf, "Survey of Pedestrian Detection for Advanced Driver Assistance Systems," IEEE transactions on pattern analysis and machine intelligence, vol. 32, no. 7, pp.1239-1258, July, 2010.
- [2] D. Xia, H. Sun and Z. Shen, "Real-time Infrared Pedestrian Detection Based on Multi-block LBP," Proc. on 2010 International Conference on Computer Application and System Modeling, vol. 12, pp. 140-142, 2010.
- [3] M. Bertozzi, A. Broggi, C. Caraffi, M. Del Rose, M. Felisa and G. Vezioni, "Pedestrian Detection by Means of Far-infrared Stereo Vision," Computer vision and image understanding 106, pp. 194-204, 2007.
- [4] P. Viola and M. Jones, "Robust Real Time Object Detection," Proc. on IEEE ICCV Workshop on Statistical and Computer Theories of Vision, 2001.
- [5] E. Osuna, R. Freund and F. Girosi, "Training Support Vector Machines: An Application to Face Detection," Proc. on IEEE Conf. Computer Vision and Pattern Recognition, pp. 130-136, 1997.
- [6] J. Giebel, D. Gavrilu, and C. Schnorr, "A Bayesian Framework for Multi-Cue 3D Object Tracking," Proc. on European Conf. Computer Vision, pp. 241-252, 2004.
- [7] U. Franke and A. Joos, "Real-Time Stereo Vision for Urban Traffic Scene Understanding," Proc. on IEEE Intelligent Vehicles Symp, pp. 273-278, 2000.
- [8] I. S. Kim and H. Shin, "A Study on Development of Intelligent CCTV

Security System Basrd on BIM," Journal of the Korea Institute of Electronic Communication Sciences, vol. 6, no. 5, pp. 789-795, 2011.

[9] M. Isard and A. Blake, "CONDENSATION - Conditional Density Propagation for Visual Tracking," International Journal on Computer Vision vol. 29, no. 1, pp. 5-28, 1998.

[10] K. Nummiaro, E. Koller-Meier, and L. V. Gool, "A Color-based Particle Filter," Proc. of 1st International workshop on generative-model-based vision, pp. 53-60, 2002.

[11] http://en.wikipedia.org/wiki/Graphics_processing_unit.

[12] <https://developer.nvidia.com/cuda-zone>

[13] NVIDIA CUDA "Cuda Reference Manual v2.0"

[14] NVIDIA CUDA "CUDA C Best Practices Guide v6.5"

[15] NVIDIA CUDA C Programming Guide, Version 4.0

[16] <http://docs.opencv.org/modules/gpu/doc/gpu.html>

저 자 소 개



최 범 준

- 2013년 : 경상대학교 전자공학과 졸업(공학사)
- 2015년 : 경상대학교 전자공학과 졸업(공학석사)
- ORCID : <http://orcid.org/0000-0002-5456-4501>
- 주관심분야 : 신호처리, 영상처리 및 이해



윤 병 우

- 1989년 3월 ~ 1992년 2월 : 부산대학교 전자공학과 공학박사
- 1993년 5월 ~ 1995년 2월 : 한국전자통신연구원 선임연구원
- 1995년 ~ 현재 : 경상대학교 전자공학과 교수
- ORCID : <http://orcid.org/0000-0002-8597-807X>
- 주관심분야 : 신호처리, 영상처리, VLSI설계, 소나시스템



송 종 관

- 1989년 : 부산대학교 전자공학과(공학사)
- 1991년 : KAIST 전기및전자공학과(공학석사)
- 1995년 : KAIST 전기및전자공학과(공학박사)
- 1995년 ~ 1997년 : SK 텔레콤 선임연구원
- 1997년 ~ 현재 : 경상대학교 전자공학과 교수
- ORCID : <http://orcid.org/0000-0001-9584-7017>
- 주관심분야 : 영상처리, 디지털신호처리, 디지털신호처리응용



박 장 식

- 1992년 : 부산대학교 전자공학과 졸업(공학사)
- 1994년 : 부산대학교 대학원 전자공학과 졸업(공학석사)
- 1999년 : 부산대학교 대학원 전자공학과 졸업(공학박사)
- 1997년 ~ 2011년 : 동의과학대학 디지털전자과 교수
- 2011년 ~ 현재 : 경상대학교 전자공학과 교수
- ORCID : <http://orcid.org/0000-0003-1794-7631>
- 주관심분야 : 적응신호처리, 영상 및 음향신호처리, 임베디드시스템