Journal of Korea Multimedia Society Vol. 18, No. 11, November 2015(pp. 1289-1301)
http://dx.doi.org/10.9717/kmms.2015.18.11.1289

A Real-time Pedestrian Detection based on AGMM and HOG for Embedded Surveillance

Thanh Binh Nguyen[†], Van Tuan Nguyen[††], Sun-Tae Chung[†††]

ABSTRACT

Pedestrian detection (PD) is an essential task in various applications and sliding window-based methods utilizing HOG (Histogram of Oriented Gradients) or HOG-like descriptors have been shown to be very effective for accurate PD. However, due to exhaustive search across images, PD methods based on sliding window usually require heavy computational time. In this paper, we propose a real-time PD method for embedded visual surveillance with fixed backgrounds. The proposed PD method employs HOG descriptors as many PD methods does, but utilizes selective search so that it can save processing time significantly. The proposed selective search is guided by restricting searching to candidate regions extracted from Adaptive Gaussian Mixture Model (AGMM)-based background subtraction technique. Moreover, approximate computation of HOG descriptor and implementation in fixed-point arithmetic mode contributes to reduction of processing time further. Possible accuracy degradation due to approximate computation is compensated by applying an appropriate one among three offline trained SVM classifiers according to sizes of candidate regions. The experimental results show that the proposed PD method significantly improves processing speed without noticeable accuracy degradation compared to the original HOG-based PD and HOG with cascade SVM so that it is a suitable real-time PD implementation for embedded surveillance systems.

Key words: Pedestrian Detection, Embedded Surveillance, HOG, AGMM

1. INTRODUCTION

Pedestrian detection (PD) has attracted much attention in the last years due to its direct applications to many domains such as automotive safety, visual surveillance, robotics, traffic controls, and so on [1,2,3]. During the last decade, various PD methods have been proposed to improve accuracy (detection ratio) and processing speed [1-23]. Along with those researches, many significant progresses have been achieved. However, current state-of-the-art PD technology still needs more improvements for reliable and embedded applications such as embedded visual surveillance. Moreover, the improvement of detection quality by devising multiple features and complex classification algorithms causes more computational burden, which makes real-time pedestrian detection more difficult for embedded systems. A trade off between the preciseness and computation cost of processing need to be considered carefully.

To improve the processing speed performance of PD, many ideas have been exploited [16,21]; better features [5,8,10,11], better classifier [6,13,15, 19,20], prior contextual knowledge [18], cascades [6,19,27], coarse-to-fine search [15], branch and

※ Corresponding Author : Sun-Tae Chung, Address: (156-743) Dept. of Smart Systems Software, Soongsil Univ., 369, Sangdo-Ro, Dongjak-Gu, Seoul, Korea, TEL : +82-2-820-0638, FAX : +82-2-821-7653, E-mail : cst@ssu.ac.kr
Receipt date : Aug. 4, 2015, Revision date : Nov. 10, 2015
Approval date : Nov. 12, 2015

[†] School of Electronic Engineering, Soongsil University (E-mail : binh@ssu.ac.kr)
[††] School of Electronic Engineering, Soongsil University (E-mail : tuannguyen}@ssu.ac.kr)
[†††] Dept. of Smart Systems Software, Soongsil University
※ This work was supported by the Soongsil University Research Fund.

bound search [7] and so on. All these approaches basically utilize the well-known, 'slide window' paradigm for detecting pedestrians. Sliding window paradigm [7,25] requires exhaustive search with detection windows over position and scale in the whole scene images, where a classifier tests for object presence for each candidate image window.

A single-scale detection requires classifying around more than several tens or hundreds of thousands windows per image, thus the number of search windows grows by an order of magnitude for multi-scale detection, which is the major causes for high computational burden of sliding window-based object detection.

Recently, in order to avoid exhaustive sliding window search across images, selective search methods [24,25] for object detection have been proposed and shown noticeable improvements with respect to detection speed. However, these selective search methods adopt some complicated segmentations to isolate the plausible regions, which still requires some sizable computational processing time.

In this paper, we concentrate on pedestrian detection for some specific application domains such as visual surveillance where the backgrounds are fixed [2]. One good thing about pedestrian detection for a specific application domain is that one can utilize the application domain prior knowledge. In visual surveillance for designated areas, the background is fixed as opposed to image-based driver assistance systems where background is changing. Thus, the pedestrians appear as moving foreground objects. In intelligent visual surveillance systems, the moving foreground objects are extracted by the minimum bounding boxes of foreground blobs. Foreground blobs under fixed backgrounds are usually extracted by background subtraction methods, among which AGMM (Adaptive Gaussian Mixture Model)-based background subtraction is well known to work successfully even under complex background scenes. Since the bounding boxes of foreground blobs are highly plausible candidate regions for pedestrians, one had better search pedestrians only in the extracted candidate regions. This kind of selective search can save search processing time significantly.

In this paper, we propose a reliable real-time pedestrian detection method for embedded visual surveillance, where search of pedestrians is restricted to the candidate regions obtained as minimum bounding boxes of foreground blobs extracted by AGMM-based background subtraction. The feature vector adopted in this paper for pedestrian detection on the candidate regions is HOG descriptor. HOG-like features based on some form of gradient histograms [5,10] are popularly utilized alone or to be combined with other features for more accurate performance of pedestrian detection.

The proposed PD method first extracts foreground blobs using AGMM-based background subtraction, takes the minimum bounding boxes of each foreground blobs as candidate regions for pedestrian existence and calculates HOG descriptors on candidate regions only, not on the whole scene. Since pedestrian sizes for the purpose of visual surveillance can be varying depending upon distance between humans on the field of view and surveillance camera, the proposed PD method chooses to adopt detection windows of three different sizes (16×32, 32×64, 64×128), and trains SVM classifiers in offline using HOG descriptor for the detection window of each size. Then, it applies the calculated HOG descriptors on each candidate region to the suitable SVM classifier according to the size of each candidate region. The calculation of HOG descriptor for each candidate image window requires not low computational cost, so that we use approximate calculation of the HOG descriptors. In addition, the proposed PD method develops an implementation in fixed-point arithmetic mode, which saves processing time further. Possible accuracy degradation due to approximate

computation is compensated by applying an appropriate one among three offline trained SVM classifiers according to sizes of candidate regions.

Through experiments, it is shown that the proposed PD method can achieve much faster pedestrian detection without noticeable accuracy performance degradation compared to OpenCV implementation [26] of the original HOG-based PD [5] and HOG with cascade SVM classifier [27]. Thus, the proposed PD method can be considered to be implemented in real-time on an embedded system.

The rest of the paper is organized as follows. Section 2 introduces backgrounds and related works necessary for understanding the contributions of the paper. Section 3 describes our proposed pedestrian detection method. Experimental results are discussed in Section 4, and finally the conclusion is presented in Section 5.

2. BACKGROUNDS AND RELATED WORKS

2.1 AGMM-based Background Subtraction

The rationale in 'background subtraction' for detecting foreground objects in videos from static cameras (the background scene of camera is fixed) is to detect the foreground objects from the difference between the current frame and a reference frame, often called 'background model'. AGMM (Adaptive Gaussian Mixture Model) [28] is a statistical background modeling which is well-known to be effective for extracting foreground objects in complex background scenes like crosswalk scenes. The pedestrians under static surveillance cameras can be extracted as foreground objects by AGMM-based background subtraction method.

Background subtraction methods are sensitive to lighting variations and scene clutters, and have difficulty in handling the grouping and fragmentation problems, and moreover foreground objects extracted from background subtraction methods are not guaranteed to include pedestrians.

2.2 HOG Descriptor

The essential idea behind the Histogram of Oriented Gradient (HOG) descriptors [5] is that local object appearance and shape within an image can be characterized by the distribution of intensity gradients or edge directions.

Fig. 1 shows processing stages of HOG-based pedestrian detection proposed in [5]

The HOG descriptor is roughly calculated as follows. First, an image window is divided into small spatial regions (cells), and then for each cell, a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell is accumulated into orientation bins of evenly distributed over $0^\circ \sim 180^\circ$ ("unsigned" gradient). For improved accuracy, the local histograms is contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination and shadowing. The final HOG descriptor is then the vector of all components of the normalized cell responses from all of the blocks in the detection window.

The OpenCV HOG implementation of [5] use 9 orientation bins, a cell of 8×8 pixel size and a block of 16×16 pixel size (2×2 cells), a detection window of 64×128 pixel size, and for contrast normalization, adopts L2-Hys, L2-norm followed by clipping (limiting the maximum values of the normalized descriptor vector $\hat{V}$ to 0.2) by default.

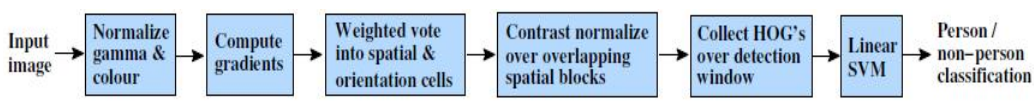


Fig. 1. Processing stages of HOG-based pedestrian detection proposed in [5].

The HOG descriptor has a few key advantages over other descriptors. Since the HOG descriptor operates on localized cells, the method upholds invariance to geometric and photometric transformations, except for object orientation. HOG has been proved to be the most stable and effective feature vector for pedestrian detection. On the other hand, HOG descriptor is not computationally light, and moreover, HOG descriptor itself is not scale invariant as opposed to Haar-like or SIFT feature vectors, which may incur a huge computation time for processing the multi-scale object detection.

2.3 Multi-scale Object detection based on sliding windows

In sliding window approach [7,16,25], a detection window, which is expected to contain the wanted object tightly, is scanned over the whole image frame. Then, for each image window, object features like HOG are extracted and applied to a classifier which determines whether the image window contains the considered objects or not. The classifier is usually trained for each detection window in offline mode. Since sliding window approach reduces the detection problem to a binary classification problem (object existence or non-existence in a detection window), training the classifier to work well is very important for good performance. The sliding window approach has been tested intensively and found to be very effective for object detection, even under image frames of low to medium resolution [16].

Since objects appears in different sizes in images and most of useful object features are not scale invariant, multi-scale object detection has been developed. Sliding window-based multi-scale object detection largely have three approaches [13];

1) (dense image pyramid) one classifier is applied to several scaled images generated from the original image frame.

2) (classifier pyramid) several level classifiers for each object size model are trained and each classifier is applied to the image frame consequently.

3) (hybrid approach) several classifiers are applied to several scaled images which generated from the original image frame.

The original HOG application to pedestrian detection [5] adopted the dense image pyramid approach. [19] uses classifier pyramid approach. [16] groups pedestrians by their image size (height in pixels) into three scales: near (80 or more pixels), medium (between 30-80 pixels) and far (30 pixels or less). This division into three scales is motivated by the distribution of sizes in the dataset, human performance and automotive system requirements. And, [16] notes that 69% of the pedestrians from most pedestrian datasets lie in the medium scale. In the context of similar observation, [5] chooses a detection window of 64x128 size and detects pedestrians of bigger sizes by applying the 64×128 detection window into down-scaled image pyramids of the original image frame.

[13] and our proposed PD method utilize hybrid approach to detect pedestrians of several different sizes effectively.

2.4 Related Works

Pedestrian (human) detection has high demand from society and industry because of its direct applications in car safety, surveillance, robotics, and so on so that it has attracted much research and development efforts and many noticeable methods have been proposed and tested during the last decade [1-23]. Most of successful methods are based on the sliding window approach [7,16,25], and thus sliding window-based PD works are reviewed with a focus on detection speed improvement.

As explained in Section 2.3, selection of features characterizing objects effectively and construction of a well-working classifier are two main pillars in the paradigms for object detection. Objects usu-

ally have different postures and shapes. Deformable objects like humans or occluded objects may not be characterized enough by features, but more effectively with a suitable object model like discriminately trained part-based model (DPM). DPM [14] and its variants[3, 15] are systematically outmatched by methods using a single component and no parts [21], casting doubt on the need for parts. Recent work has explored ways to capture deformations entirely without parts [22].

Features proposed and tested during the past decade include Haar-like [4], HOG [5], HOG-like [1], LBP (Local Binary Pattern) [6,10], Covariance matrix [9], integral channel features[11], bag-of-words over dense SIFT [7], combination of these features [8,10,11,13], and so on. Even though multiple features show better accuracy performance, they usually cost more processing time. Also, recent research [8,9] shows that HOG alone is not best but still a competitive feature.

For a given set of features, the choice of classifier has a substantial impact on the resulting speed and quality, often requiring a trade-off between these two. Boosted classifier, SVM, and neural networks are popularly adopted as a classifier for sliding-window based object detection. In the work on MultiFtrs [8], it was argued that, given enough features, Adaboost and linear SVM perform roughly the same for pedestrian detection. Definitely, nonlinear SVMs such as latent SVM and HIK SVM performs better accuracy but it costs more computational time.

As a result, linear classifiers such as Adaboost, and linear SVMs are more commonly used. Recent work on the linear approximation of nonlinear kernels seems a promising direction. A frequently used method for speeding up classifiers is a cascade idea, which splits classifiers up into a sequence of simpler classifiers. By having the first stages prune most of the false positives, the average computation time is significantly reduced [6].

In general, image processing greatly benefits from prior knowledge. For pedestrian detection, the presence of a single dominant ground plane has often been used as prior knowledge to improve both speed and quality [15,18].

Common detection methods require resizing the input image and computing features multiple times (refer to Fig. 2(a)). [4] showed the benefits of "scaling the features not the images", however such approach cannot be applied directly to HOG-like feature because of blurring effects. [19] presented the first detector based on orientation gradients that requires no image resizing. The core idea of [19] is to move the resizing of the image from test time to training time (Fig. 2(b)). Since the PD method in [19] can approximate the feature responses across scales, it can decide how to adjust a given stump classifier to classify correctly, as if the feature response had been computed at a different scale. This can be seen as a "better classifier". This improvement provides a ~ 3.5 times algorithmic speed-up on the features computation stage.

Recently, in order to avoid exhaustive sliding window search across images, selective search

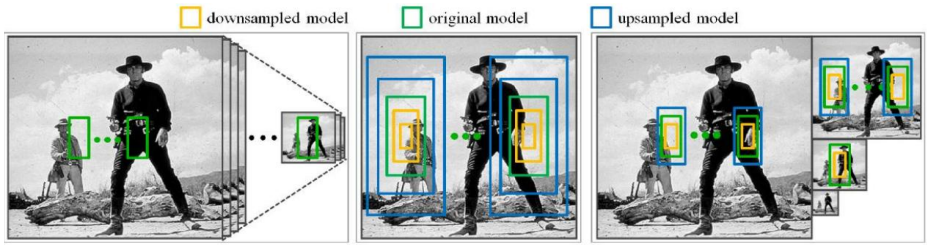


Fig. 2 Approaches to sliding window-based multiscale object detection [13].

methods [24,25] for object detection have been proposed and shown noticeable improvements with respect to detection speed. However, these selective search methods adopt some complicated segmentations to isolate the plausible regions, which still requires some sizable computational processing time. The proposed method in this paper is similar to the selective search methods, but is different from them in the sense that our proposed PD method applies searches to foreground object candidate regions extracted by background subtraction while those selective searches of [24, 25] are applied to regions segmented by grouping or other segmentation algorithms, which is more heavier than background subtraction method like AGMM-based one.

Some works on a sliding window-based pedestrian detection for visual surveillance with fixed cameras have been reported [2,23]. [2] discussed how to automatically transfer a generic pedestrian detector to a scene-specific detector in static video surveillance without manually labeling samples from the target scene, and did not deal with how to improve processing speed. Even though [23] searches pedestrians on regions of interest extracted by background subtraction similar to our proposed PD method, it is different from ours at least in 3 ways; 1) [23] simply applies frame difference and the extracted ROI may be less plausible as a pedestrian container than that by AGMM adopted in this paper. 2) [23] applies a SVM classifier trained with HOG features in one size of detection window (64×128) and verifies more securely with covariance matrix whether the extracted ROI contains pedestrians or not. But, [23] does not mention how to deal with when the extracted ROIs have sizes less than 64×128, which happens when the camera needs to monitor far view scenes. On the other hand, our proposed PD method uses three SVM classifiers trained with HOG features in three sizes of detection windows (16×32, 32×64, 64×128) so that it can detection small sized pedestrians. Also, ours uses approximate HOG feature in order to speed up processing. 3) [23] does not provide detailed experimental results as opposed to ours.

3. THE PROPOSED REAL-TIME PEDESTRIAN DETECTION

Fig. 3 shows the working flow for the proposed pedestrian detection method as a module under the common embedded system.

3.1 Extraction of Candidate Regions using AGMM-based background subtraction

Under a fixed surveillance camera, one can model the scene without any foreground objects (humans, vehicles or some other objects) as a background scene. Then, pedestrians on the scene can be found on foreground blobs, which identifies regions of foreground blobs, a collection of foreground pixels. For pedestrian detection purpose, a minimum bounding box of a foreground mask is extracted as a candidate region which may contain pedestrians.

To extract foreground blobs from a frame image, we adopt AGMM-based background subtraction method[28]. Fig 4. shows the steps of our adopted

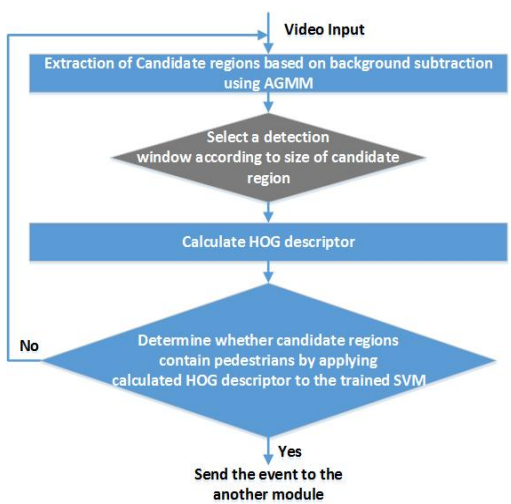


Fig. 3. Working flow of the proposed method (at testing).

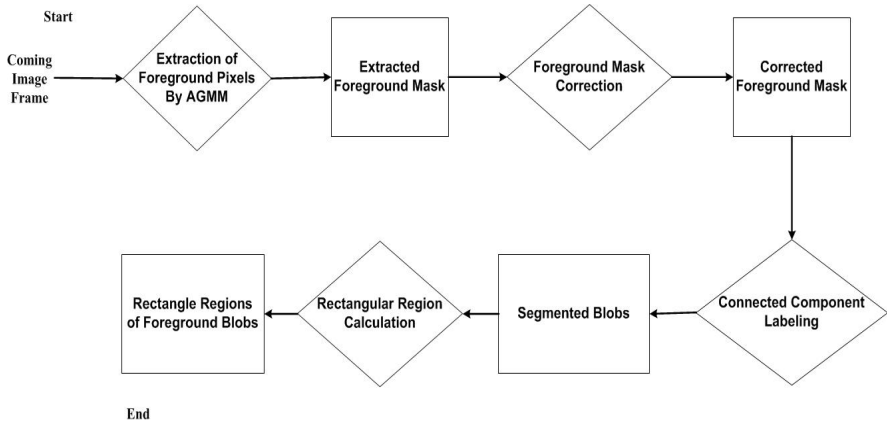
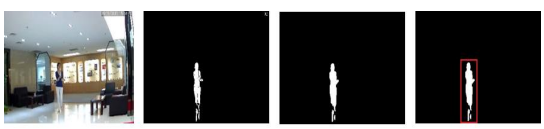


Fig. 4. Steps of the adopted AGMM-based candidate region extraction.

AGMM-based extraction of candidate regions for foreground pedestrians [29].

Fig. 5. shows some resulting images associated with steps in Fig. 4.



(a) (b) (c) (d)

Fig. 5. The result images of candidate region extraction steps of Fig. 4. ; (a) original image, (b) Foreground mask, (c) corrected foreground mask, (d) candidate region; rectangle region of the corrected foreground mask

3.2 Applying HOG-based PD to the candidate regions

3.2.1 Approximate calculation of HOG descriptors on candidate regions

In order to reduce computation time of HOG descriptors, the proposed PD method approximates calculation of HOG descriptors' gradient in a less computation time by reducing the number of orientation binning for the cell histogram to 4 from 9, originally chosen in [5] over $0^{\circ} \sim 180^{\circ}$.

Less number of the orientation bins is reported to show performance degradation. But, in the proposed PD method, possible degradation owing to less number of orientation bins is compensated by applying a suitable SVM classifier among three SVM classifiers trained in three different sizes of detection windows (16×32, 32×64, 64×128). Then, noticeable accuracy performance degradation is not observed through extensive experiments.

3.2.2 Block normalization

For block normalization scheme, we adopt L2-Hys as in OpenCV implementation of HOG-based PD [26]. L2-Hys is L2-norm followed by clipping (limiting the maximum values of the normalized descriptor vector $\hat{V}$ to 0.2), where normalization by L2-norm is to normalize the unnormalized descriptor vector V by $\hat{V} = V/(\sqrt{(\|V\|_2^2 + \epsilon^2)}$.

3.2.3 Hybrid Sliding Windows

As mention in Section 2.3, the original HOG-based PD [5] scans a detection window across all of the dense image pyramids. The proposed PD has only to scan a detection window over the candidate regions. Each candidate region is expected to contain a pedestrian tightly, if it contains one.

In the similar way to [5], one may apply one detection window for image pyramids of the candidate regions. However, in this paper, we apply a different detection window of the appropriate size according to the size of each candidate region. The selected detection windows are of three sizes; 16

× 32, 32 × 64, and 64 × 128. If the candidate region is less than 16 × 32, it is discarded. If it is greater than 64 × 128, the proposed PD method applies the detection window of 64 × 128 to the down-scaled image pyramids of the candidate region. It is noted that the minimum down-scaled image is not less than 64 × 128. Otherwise, the proposed PD method applies the detection window with the size close to that of each candidate region.

As mentioned in Section 2.3, this hybrid approach computationally has advantage over the original image pyramid approach. The three classifiers are trained in offline mode, and in the testing stage, the proposed PD can save a lot of computational time since it does not have to scan the down-scaled image pyramids or just have only to scan a few down-scaled image pyramids. Also, as mentioned earlier, it is observed that applying a suitable SVM classifier according to the size of the candidate regions compensate the possible accuracy degradation due to approximate computation.

3.3 Training Support Vector Machine

For classification of the HOG descriptors, we adopt relatively computationally light linear SVM, which evaluates.

$$Y(x) = w^T x + b \tag{1}$$

The weight vector w and the bias b are determined during the training phase of the SVM. If $Y(x) > 0$, it is determined that pedestrian is detected, otherwise not. We trained SVM using sample images from INRIA database [30]. Training sample data set consists of two parts; positive images (pedestrian images) and negative images (non-pedestrian images). The data set was scaled to three different sizes to train three SVM classifiers proportional to level 1 = 16 × 32, level 2 = 32 × 64, and level 3 = 64 × 128. Fig. 6 shows some example images in INRIA database.

Fig. 6. Some sample images in INRIA pedestrian database.

3.4 Fixed-point mode Computation

Many processing steps including computation of HOG descriptors, classifying by SVM, and so on require floating-point arithmetic. Even in PC environments, fixed-point operations are faster than floating-point operations. In the embedded environments, many of embedded processors, especially DSP, support only fixed-point integer computation. The processing with floating-point arithmetic is usually more precise but computationally heavier than the fixed-point arithmetic.

In order to reduce the numerical impreciseness due to fixed-point computation, we adopt Q-format for approximating floating-point computations more precisely in fixed-point integer computations. As is known, Q-format is a fixed point number format where the number of fractional bits (and optionally the number of integer bits) is specified. For example, a Q10 number has 10 fractional bits. For our proposed PD method, we apply Q10 fixed-point arithmetic for calculating HOG descriptors, SVM training, and SVM classification, and all other computational processing as a maximized optimum of shifting. The choice of the number of fractional bits depends on the maximum value of variables during processing. If one selects a small number for fractional bits, the margin of error in calculations becomes large and the preciseness of the computing result will be reduced. On the contrary, if one desires the number of fractional bits too big, then the result of formular which applies addition or multiplication to model with a list of

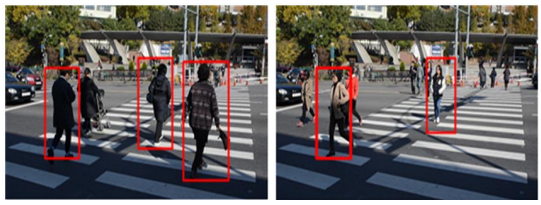
vector's elements may produce value which cannot be expressed by 32 bit integer number.

Through the experiments, it is found that the fixed-point processing using Q10 format even in the PC environments achieves significant speed improvements without severe performance degradation, and that the fixed-point implementation helps a lot to make the proposed PD method operate in real-time for DSP like TI DSP DM6437.

4. EXPERIMENTAL RESULTS

For performance evaluation of the proposed method, we implemented it in both conventional floating-point mode and the fixed-point mode, and compared the implementations with the PD implemented in Open-CV library [26] (hereafter, OpenCV-HOG-PD) and the HOG with cascade SVM [27] with respect to accuracy and processing time. The OpenCV-HOG-PD implements 'HOG + SVM' [5]. The HOG with cascade SVM we utilized in this paper for comparison is what we retrained the object detection with cascade SVM of [27] using HOG descriptors for pedestrian detection.

For testing, we prepared two videos. Video1 is taken about the crosswalk at near Soongsil University, Seoul, Korea. It consists of 901 frames of 1920 x 1080. Video2 is a street view movie obtained from Internet and it consists of 1856 frames of 1920 × 1080. Testing Windows PC has a processor Core i7 2600 of 3.4GHz and main memory of 8GB.



(a)

(b)

Fig. 7. The result of applying the proposed method; (a) Video 1, frame 126 and 488. (b) Video 2, frame 480 and 700.

Our employed measure for the correct detection is PASCAL measure [16], which determines correct object detection if the area of overlap exceed 50%:

$$a_0 := \frac{area(BB_{dt} \cap BB_{gt})}{area(BB_{dt} \cup BB_{gt})} > 0.5 \quad (2)$$

Fig. 7 shows some result images from applying the proposed PD method to Video 1 and Video 2.

We compared our proposed PD method with OpenCV-HOG-PD and HOG with cascade with respect to accuracy and processing time. The experimental results of Table 1 show the comparison with respect to accuracy.

Table 2 shows the comparison experimental results among HOG-PD, HOG with cascade SVM and the proposed methods in two arithmetic modes of floating-point and fixed-point with respect to

Table 1. Comparison with respect to accuracy

Method Name	Video 1		Video 2	
	Miss Rate (%)	False positive per frame	Miss Rate (%)	False positive per frame
OpenCV-HOG-PD	20	0.80	20.5	0.67
HOG with Cascade SVM [27]	19	0.77	19.3	0.62
Proposed method in floating-point mode	21.5	0.88	21.3	0.67
Proposed method in fixed-point mode	22	0.92	22.4	0.69

Table 2. Comparison with respect to processing time

Method Name	Video 1		Video 2	
	Time (s)	Speed (fps)	Time (s)	Speed (fps)
OpenCV-HOG-PD	47.56	13.51	137.72	13.15
HOG with Cascade SVM [27]	41.23	22.22	114.54	15.70
Proposed method in floating-point mode	8.45	135.33	18.89	71.42
Proposed method in fixed-point mode	8.42	138.02	17.77	75.33

processing time. It is noted that the original frame sizes are scaled down to 480 × 320 before processing.

The experimental results in Table 1 and Table 2 show that the proposed method performs faster than OpenCV-HOG-PD [26] and HOG with Cascade SVM [27] without noticeable accuracy performance degradation. The implementation in fixed-point showed faster processing time with marginal accuracy degradation compared floating-point implementation. HOG with cascade SVM [27] generally shows better precise and processing speed [16] compared to conventional HOG-SVM approach, but the cascading classifiers employed in HOG with cascade SVM [27] will not be very useful for our proposed method. Since the cascading classifiers achieves processing speedup by discarding unplausible negative samples sequently through cascading SVM classifiers, speedup gain will be greater when the number of negative samples are far more than that of positive samples. In our proposed PD method, the candidate regions by AGMM is likely to include pedestrians. Thus, applying cascading classifiers to candidate regions may not have any computational gain much.

The implementation in fixed-point mode was easily ported into TI DM 6437 DSP Processor with fixed-point arithmetic only supported, and it was found to show 10 fps processing speed for input video resolution of 720 × 480 with scaled-down 480 × 320 for internal processing.

5. CONCLUSIONS

In this paper, we proposed an improved real-time pedestrian detection (PD) method for embedded visual surveillance. The proposed PD method utilizes selective search, but not exhaustive search required for original HOG-based PD methods by restricting searching to candidate regions extracted by AGMM-based background subtraction. The proposed PD method first extracts foreground blobs as candidate regions for pedestrians, computes approximate HOG descriptors on the candidate regions, not on the whole scene frame and determines whether the candidate regions contain pedestrians by applying the calculated HOG descriptors to one of the three trained SVMs according to sizes of the candidate regions. In order to speed up processing more, all computations are done in fixed-point mode without noticeable accuracy degradation by utilizing Q-format. This fixed-point implementation was easily ported into fixed-point DSPs such as TI 6437 processor and the ported proposed PD method was found to run in real-time.

Through experiments, it is shown that our proposed method in both floating-point mode and fixed-point mode is much faster without noticeable accuracy performance degradation compared to OpenCV-HOG-PD[26] and HOG with Cascade SVM [27].

REFERENCE

[1] A. Shashua, Y. Gdalyahu, and G. Hayun,

"Pedestrian Detection for Driving Assistance Systems: Single-Frame Classification and System Level Performance," *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 13-18, 2004.

[2] X. Wang, M. Wang, and W. Li, "Scene-Specific Pedestrian Detection for Static Video Surveillance," *IEEE Transactions on Software Engineering*, Vol. 36, No. 2, pp. 361-374, 2014.

[3] J. Yan, X. Zhang, Z. Lei, S. Liao, and S.Z. Li, "Robust Multi-Resolution Pedestrian Detection in Traffic Scenes," *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3033-3040, 2013.

[4] P. Viola and M. Jones. "Robust Real-Time Face Detection," *International Journal of Computer Vision*, Vol. 57, Issue 2, pp. 137-154, 2004.

[5] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 886-893, 2005.

[6] Q. Zhu, M.C. Yeh, and K.T. Cheng, "Fast Human Detection using a Cascade of Histograms of Oriented Gradients," *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1491-1498, 2006.

[7] C.H. Lampert, M.B. Blaschko, and T. Hofmann. "Beyond Sliding Windows: Object Localization by Efficient Subwindow Search," *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, 2008.

[8] C. Wojek and B. Schiele, "A Performance Evaluation of Single and Multi-Feature People Detection," *Lecture Notes in Computer Science*, Vol. 5096, pp. 82-91, 2008.

[9] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian Detection via Classification on Riemannian Manifolds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 10, pp. 1713-1727, 2008.

[10] X. Wang, T.X. Han, and S. Yan, "An HOG-LBP Human Detector with Partial Occlusion Handling," *Proceeding of IEEE International Conference on Computer Vision*, pp. 32-39, 2009.

[11] P. Dollar, Z. Tu, P. Perona, and S. Belongie, "Integral Channel Features," *Proceeindg of The British Machine Vision Conference*, pp. 91.1-91.11, 2009.

[12] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: A Benchmark," *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 304-311, 2009.

[13] P. Dollar, S. Belongie, and P. Perona, "The Fastest Pedestrian Detector in the West," *Proceeding of The British Machine Vision Conference*, pp. 68.1- 68.11, 2010.

[14] P.F. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, Issue 9, pp. 1627-1645, 2010.

[15] D. Park, D. Ramanan, and C. Fowlkes, "Multiresolution Models for Object Detection," *Proceeding of European Conference on Computer Vision*, pp. 241-254, 2010.

[16] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: An Evaluation of the State of the Art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34, Issue 4, pp. 743-761, 2011.

[17] M. Pedersoli, A. Vedaldi, and J. Gonzalez. "A Coarse-to-Fine Approach for Fast Deformable Object Detection," *Proceeding of IEEE Conference Computer Vision and Pattern Recognition*, pp. 1353-1360, 2011.

[18] P. Sudowe and B. Leibe, "Efficient Use of Geometric Constraints for Sliding-Window

Object Detection in Video," *Proceeding of International Conference on Computer Vision Systems*, pp. 11-20, 2011.

[19] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, "Pedestrian Detection at 100 Frames per Second," *Proceeding of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2903-2910, 2012.

[20] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian Detection with Unsupervised Multi-Stage Feature Learning," *Proceeding of IEEE Conference Computer Vision and Pattern Recognition*, pp. 3626-3633, 2013.

[21] R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Ten Years of Pedestrian Detection, What Have We Learned?," *Proceeding of European Conference on Computer Vision*, pp. 613-627, 2014.

[22] B. Hariharan, C.L. Zitnick, and P. Dollár, "Detecting Objects using Deformation Dictionaries," *Proceeding of IEEE Conference Computer Vision and Pattern Recognition*, pp. 1995-2002, 2014.

[23] K.M Bhuvanarjun and T.C. Mahalingesh, "Pedestrian Detection in a Video Sequence using HOG and Covaraince Method," *International Journal of Electrical and Electronics Engineers*, Vol. 7, Issue 1, pp. 183-190, 2015.

[24] K. van de Sande, J. Uijlings, T. Gevers, and A. Smeulders, "Segmentation as Selective Search for Object Recognition," *Proceeding of IEEE International Conference on Computer Vision*, pp. 1879-1886, 2011.

[25] J. Hosang, R. Benenson, Piotr Dollár, and B. Schiele," What Makes for Effective Detection Proposals?," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Accepted, 2015.

[26] OpenCV Library, http://opencv.org/, 2015. 11. 30.

[27] X. Yang, C. Yi, L. Cao, and Y. Tian, "*Media-CCNY at TRECVID 2012: Surveillance Event Detection*," NIST Trecvid Workshop, 2012.

[28] C. Stauffer and C, W.E.L Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," *Proceeding of Conference on Computer Vision and Pattern Recognition*, pp. 246-252, 1999.

[29] T.B. Nguyen, S.T. Chung, and S.W. Cho, "An Effective Moving Cast Shadow Removal in Gray Level Video for Intelligent Visual Surveillance," *Journal of Korea Multimedia Society*, Vol. 17, No. 4, pp. 420-432, 2014.

[30] INRIA Person Dataset, http://pascal.inrialpes.fr/data/human/, 2015. 11. 30.

Thanh Binh Nguyen

He was born in Viet Nam in 1984. He received the B. Eng. degree in computer science from the University of Science, Ho Chi Minh, Viet Nam, in 2005, the M.Sc. degree in information and telecommunication engineering from the University of SoongSil, Seoul, South Korea, in 2010, and pursuing the Ph.D. program in engineering at the University of SoongSil, Seoul, South Korea, from 2012. He had worked as a software developer for 3 years from 2005 at Viet Nam. He is currently a principal software R&D researcher at Embedded Vision Inc., Seoul, South Korea, and has been a research assistant at Embedded Real-time Computing Lab., University of Soongsil, Seoul, South Korea. His research interests cover the design and analysis of various smart embedded software system, IoT and also intelligent image, video analytic algorithms which are applied to visual surveillance, recognition systems, and etc.

Van Tuan Nguyen

He was born in Vietnam in 1991. He received the B.E degree in electronics and computer engineering from Hanoi University of Science and Technology, Hanoi, Vietnam, in 2014. He is currently a research assistant at Embedded Real-time Computing Laboratory, Soongsil University, Seoul, South Korea. His main areas of research interest are embedded systems, image processing, visual surveillance, recognition systems and machine learning.

Sun-Tae Chung

He received B.E. degree from Seoul National University, and M.S. degree and Ph.D. degree in Electrical Eng. and Computer Science from the University of Michigan, Ann Arbor, USA, in 1986 and 1990, respectively. Since 1991 until, he had been with the School of Electronic Eng. at the Soongsil university, Seoul, Korea. Since 2015, he is now affiliated with Department of Smart Systems Software, Soongsil University, which has been established in 2015 for education and researches focusing in the area of industrial fields needing embedded systems and smart software technologies His research interests include: computer vision, realistic media, visual surveillance, biometrics, digital signage, and embedded systems.