2012 International Conference on Image, Vision and Computing (ICIVC 2012)
IPCSIT vol. 50 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V50.5

Efficient Block-Based Foreground Object Detection for Outdoor Scenes on Embedded SoC Platforms

Wen-Kai Tsai [1 +], Ming-Hwa Sheu [1], Shyue-Wen Yang [1], Cing-Hua Gao [1] and Tzu-Hsuing Chen [2]

[1] National Yunlin University of Science & Technology

[2] Taipei Chengshih University of Science and Technology

Abstract. Foreground object detection is one of crucial techniques for image surveillance systems. This paper presents a block-based foreground detection algorithm that possesses fast processing and low memory requirement to be performed on the embedded system for real-time operations. The gray distribution of each block is analyzed, and then the major gray pixels are selected to construct the block background model. Because of using gray image, our proposed approach can save up to 94 % memory consumption under the requirement of reasonable object detection, when comparing to other existing methods. Besides, this algorithm can improve objection detection speed. At last, the proposed approach for the moving object detection has been implemented on the TI DaVinci platform. It can achieve 25 frames per second for the benchmark video with image size 640×480.

Keywords: background subtraction, object detection, embedded platform.

1. Introduction

Real-time foreground object detection is important to many computer vision applications such as visual surveillance, intelligent behavior recognition and vehicle motion tracking. These applications are highly rely on the accuracy of foreground object detection. Most methods of current foreground object detection trade computation complexity and memory space for segmentation accuracy. Thus, they require the higher computation power and memory size such that they are only executed in high performance PC. Although embedded system has advantages of low cost, and small dimensions for real applications, it has the performance limitation in the systemic computation and memory.

While developing an embedded system, the processor speed, memory space, and accuracy have to be taken into account since the resource of an embedded system is confined. Background subtraction, which is a common and efficient approach for image object detection, is not suitable for embedded system. In general, backgrounds are non-static, thus, multi-layer background model or multimodal background, such as MoG model [1,2], Codebook model [3], SACON [4], and texture model[5,6], are used in recent years. If these methods work in the circumstances of non-static backgrounds, their memory consumption and computation effort will depend on the background complexity. Thus, these methods are hard to be implemented in the embedded system for real case applications.

In contrast to multi-layer background model, single layer background model [7, 8, 9] has the advantages of low memory consumption and fast processing. It can be applied in an embedding system if the variation of scene changes is small. In order to solve the problem, [10] proposed the probability-based background extraction algorithm. The approach combining with good post-processing was able to improve real-time processing and memory consumption but it could not overcome a sudden scene change.

+ Corresponding author. Tel.: + 886-5534-2601 Ext. 4320.
E-mail address: g9610804@yuntech.edu.tw.

In recent years, many object detection algorithms are designed and implemented in an embedded system. In [11], a real-time moving object detection system was implemented on SoC embedded platform by hardware/software co-design. It applied the technique of image difference; thus, all non-static objects are regarded as foreground objects. [12] presented an ALW algorithm which had low memory consumption and memory access, but it only operated for minor scene change. To reducing the mass calculation and memory accesses required in multi-media and communication systems, [13] proposes a high efficient but low-power consumption embedding platform, PAC Duo Soc. Its architecture includes an ARM926EJ-S kernel and two PAC digital signal processing kernels. It is able to process foreground object detection in VGA (640*480) resolution for 30 frames per second. However, its algorithm requires some post-processing such as labeling, noise removing, morphological operation, etc. to solve the problem of non-static scenes.

In this paper a block-based background model is presented. Memory consumption is dramatically reduced since only the major gray of each block of background model is saved. In the detection phase, the input block and the major gray of its corresponding background block are compared, and then foreground block can be detected quickly. A mechanism is also applied to smoothen edge blocks; thus block effect can be improved and background object can be detected precisely. In addition, the major gray is applied to obtain a background model in case a non-static object exists in a scene; thus, the proposed method can be used in a complex background. Applying block-based major gray to construct a background model and detect foreground objects has the advantages of high speed and low memory consumption. On the embedding platform of Davinci, our proposed approach reaches at least 16 fps (frames per second) with image size of 768×576, and 25 fps with image size of 640×480. After detection, the object similarity value is up to 95% in non-static scenes.

2. Proposed Method

2.1. Block-based Background Construction

The basic concept of the proposed block-based algorithm is to model background as a composition of major gray. The background model can be constructed by Gaussian Mixture Model. Each of these background frames is divided into non-overlapping square blocks. After block-based Gaussian distribution analysis[1], a block's major gray, $BM=\{ bm(r) | r=1,2,\ldots,s \}$, can be obtained, where s is the number of major grays in the block. The analysis of major gray is introduced as follows. Let $BL = \{x_1, x_2,\ldots, x_t\}$ be an input block which size is $n \times n$, the probability of this block is

$$P(BL) = \sum_{i=1}^{k} \omega_i \, \eta(BL, \mu_i, \sigma_i) \tag{1}$$

where k is the number of Gaussians. ω_i, μ_i, and σ_i are weight, mean, and the standard deviation of the ith Gaussian in the mixture, respectively. In order to update initial model, every pixel in the block (BL) is checked against the k Gaussian distributions to determine whether this value is within 2.5 standard deviation of one of them. If the input pixel is determined to match any distributions, the weights of the k distribution are updated as follows:

$$\omega_i = (1-\alpha) \times \omega_i + \alpha \times M_i \tag{2}$$

where α is a learning rate, and M_i is 1 for the distribution which matched to the input pixel and 0 for the other distributions. Similarity, μ_i and σ_i are updated as follows:

$$\mu_i = (1-\rho) \times \mu_i + \rho \times x_t \tag{3}$$

$$\sigma_i = (1-\rho) \times \sigma_i + \rho (x_t - \mu_i) \tag{4}$$

where ρ is a learning rate for adapting distributions.

If none of the k distributions match the pixel, the least probable distribution is replaced with a new distribution. And then, we set the parameters of distribution by x_t. After learning N frames, k Gaussian distributions can be built. Finally, the distribution having a low weight value will be eliminated. The distributions are sorted in descending order according to the corresponding weights, and the k Gaussian distributions are selected as for the real background model by using Eq. (5).

$$s = \arg\min_i [(\sum_{i=1}^{k} \omega_i) > \tau] \tag{5}$$

where τ is a threshold value ($0 < \tau < 1$) and s $\leq$ k.

2.2. Foreground Object Detection and Background Update

After the construction of background model, foreground objects can be obtained by background subtraction. In order to achieve the real time performance, the input pixel is compared with major grays of the corresponding background block in the detection phase. However, for efficient computation, the algorithm in the detection phase does not use the Gaussian distribution analysis to detect the foreground objects. It consists of the following steps:

The first step of block-based foreground objects detection compares the input pixel with background model (BM). Let $I_{x,y}$ be an input pixel with gray level of the corresponding block. And then, a foreground object can be detected by block-based background subtraction, as in Eq. (6):

$$FO_{i,j} = \begin{cases} 0, & if \ \left| I_{x,y} - bm(r) \right| \leq th \\ 1, & else \end{cases} \tag{6}$$

where x and y are the coordinates of the block. And then, bm(r) is the corresponding background model. While FO is equal to 1, it stands for foreground object pixel; otherwise, it represents background.

In the second step, we must update the background model over time to prevent detection errors resulting from outdated background information. As shown in Eq. (7), if $FO_{i,j}$=0, update the matched major gray color of background model by a running average.

$$bm(r) = \alpha \times I_{x,y} + (1-\alpha) \times bm(r) \tag{7}$$

where α is a constant whose value is less than 1.

3. Experimental Results and Comparisons

To evaluate the performance of foreground object detection, four test video sequences including waving trees, Hall, and PETS 2000 are used in the experiments. The performance of the proposed method was compared with those of SACON[4], Wang[9],and ALW[12]. A pixel-based total error[14] based on ground truth is a fair and often adapted assessment method, and is used to evaluate the accuracy of object detection. The total error pixels is given in Eq. (8),

$$\text{Total error pixels} = fp + fn \tag{8}$$

where *fn* and *fp* represent the sum of false negative and the sum of false positive, respectively. Thus, the larger number of total error pixels, the more detected object are unlike ground truth.

In Fig. 1, we demonstrate that the proposed apporach exhibits better the results of foreground detection than other methods. Fourthermore, Fig. 2 demonstrate that the proposed method presents a lower total error pixels [14] in the ground truth comparison. The resources of an embedded platform are limited, so the implementation has to consider memory consumption. Table 1 lists the real memory utilization for all four methods when applied to the different video sequences. The resources of an embedded platform are limited, so the implementation has to consider memory consumption. Table 1 lists the real memory utilizations for various methods on difference video sequences. Because of using gray color image, our proposed approach consumes 94.2% less memory compared to other methods; thus, the proposed method is suitable for implementing in an embedded platform. Our proposed approach has been implemented on an embedded system to meet the real-time requirement as illustrated in Fig. 3. This embedded platform, TI Davinci, is a dual-core system including ARM926, C64x DSP, peripheral devices, interfaces and CCD camera. Since the DSP processor only supports fixed-point operations, many methods with floating-point operations [4, 9, 12] are not suitable for this platform. The proposed method on the platform can achieve real-time execution for an image size of 640×480 in 25 frames per second.

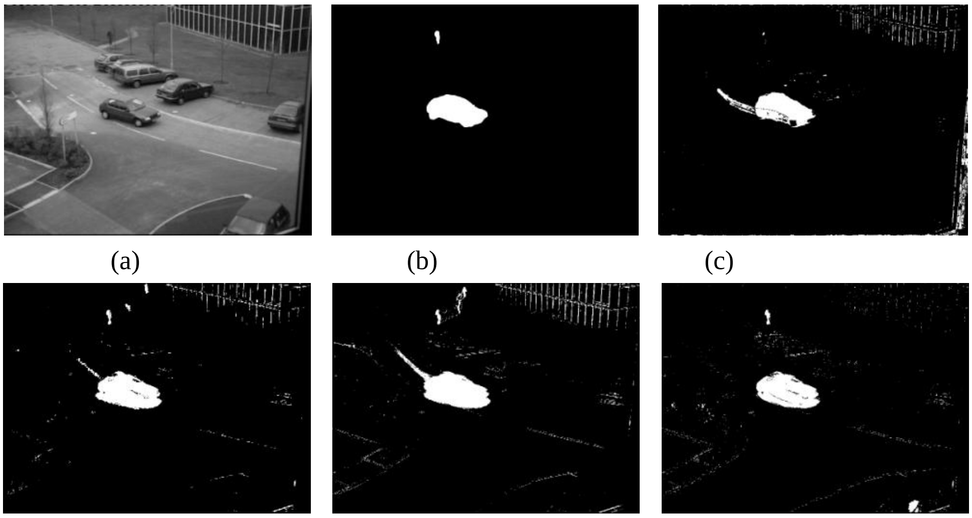

Fig. 1: Foreground object detection results with Dataset 1 of PETS 2000. (a) Original image. (b) Ground truth. (c) SACON[4]. (d) Wang [9]. (e) ALW[12]. (f) Tsai (4×4).

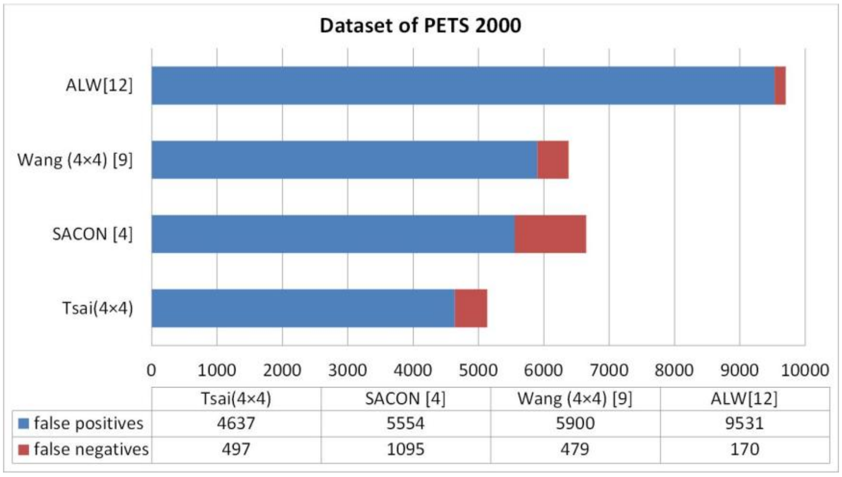

Fig. 2: Performance comparison in PETS 2000

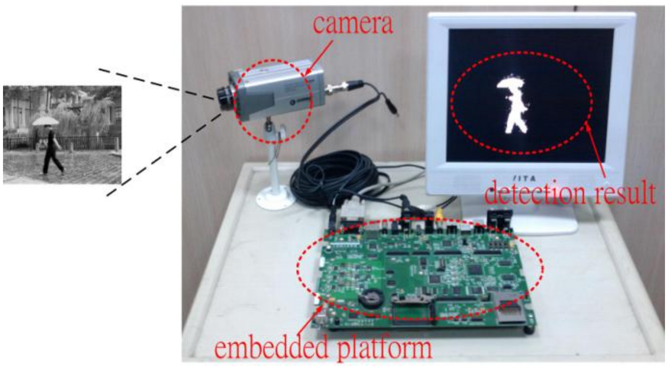

Fig. 3: Implementation of the embedded system

Table. 1: Memory consumption for background modeling

	PETS 2000
Image Size	768×576
SACON [4]	34.2 MBytes
Wang [9]	4.02 MBytes
ALW [12]	2.78 MBytes
Tsai (4×4)	31.2 KBytes

4. Conclusions

In this paper, we propose an accurate and fast foreground object detection algorithm that can be performed on embedded platform. It introduces a novel block-based major gray approach to construct moving background model. Therefore, memory consumption can be greatly reduced. In average, the memory consumption of the proposed method is 94.2% lower than the single layer background model. Thus, the proposed algorithm is suitable for implementation on an embedded system. Under those resource constraints, the experimental results show that the proposed method can achieve real-time at 25 FPS on the Davinci platform. Finally, the detection results indicate that the images created by our proposed method look better and are the closest to ground-truth image. Besides, the proposed method reduces 64.2%~92.5% total error pixels than others in dynamic scene.

5. References

[1] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 8, pp. 747-757, Aug. 2000.

[2] H. L. Eng, J. Wang, Alvin Harvey Kam Siew Wah and W. Y. Yau, " Robust human detection within a highly dynamic aquatic environment in real time," IEEE Trans. Image Processing, vol. 15, no. 6, pp. 1583-1600, June. 2006.

[3] J. M. Guo, Y. F. Lin, C.H Hsia, M. H. Shih, and C. S. Hsu, " Hierarchical Method for Foreground Detection Using Codebook Model," IEEE Trans. Circuits Syst. Video Technol., vol. 21, no. 6, pp. 804-815, Jun. 2011

[4] Hanzi Wang, and David Suter, "A consensus-based method for tracking modeling background scenario and foreground appearance," Pattern Recognition, pp.1091-1105, 2006.

[5] M. Heikkila, and M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 4, pp. 657-662, Apr. 2006.

[6] Y. T. Chen, C. S. Chen, C. R. Huang and Y. P. Hung, "Efficient hierarchical method for background subtraction," Pattern Recognition, vol. 40, pp. 2706-2715, Nov. 2007.

[7] S. Y. Chien, Y. W. Huang, B. Y. Hsieh, S. Y. Ma, and L. G. Chen, "Fast video segmentation algorithm with shadow cancellation, global motion compensation, and adaptive threshold techniques," IEEE Trans. Multimedia, vol. 6, no. 5, pp. 732-748, Oct. 2004.

[8] C. F. Juang, C. M. Chang, J. R. Wu, and D. M. Lee, "Computer vision-based human body segmentation and posture estimation," IEEE Trans. Syst., Man, and Cyber., Part A: Systems and Humans, vol.39, no.1, pp. 119-133, January 2009.

[9] W. Wang, J. Yang, and W. Gao, " Modeling background and segmenting moving objects form compressed video," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 12, pp. 1477-1490, Dec. 2006.

[10]C. C. Chiu, M. Y. Ku, et.al., "A robust object segmentation system using a probability-based background extraction algorithm," IEEE Trans. Circuits Syst. Video Technol., vol. 20, no. 4, pp. 518-528, Apr. 2010.

[11]C. H. Moon, D. Y. Jang, and J. N. Choi, "An SoC system for real-time moving object detection," Lecture notes in computer science, vol. 4681, pp. 879-885, 2007.

[12]Y. Wang, S. Velipasalar, and M. Casares, "Cooperative object tracking and composite event detection with wireless embedded smart cameras," IEEE Trans. Image Processing, vol. 19, no. 10, pp. 2614-2633, Oct. 2010.

[13] C. Y. Cho, J. K. Yang, S. Y. Tseng, and M. H. Sheu, "Real-time object detection system on PAC Duo platform," International Conference on Signal Processing Systems, pp. 585-590, 2010.

[14]K. Toyame, J. Krumm, B. Brumitt and B. Meyers, "Wallflower: Principles and practice of Background Maintenance," Int. Conf. on Computer Vision, vol. 1, pp. 255-261, 1999.