Efficient Foreground Detection for Real-Time Surveillance Applications

S. Gruenwedel, N. I. Petrović, L. Jovanov, J.O. Niño-Castañeda, A. Pižurica and W. Philips

This paper addresses the problem of foreground detection in real-time video surveillance applications. We propose a framework, which is computationally cheap and has low memory requirements. It combines two simple processing blocks, both of which are essentially background subtraction algorithms. The main novelty of our approach is a combination of autoregressive moving average filter with two background models having different adaptation speeds. The first model, having a lower adaptation speed, models long-term background and detects foreground objects by finding areas in current frame which significantly differ from the proposed background model. The second model, with a higher adaptation speed, models the short-term background and is responsible for finding regions in the scene with a high foreground object activity. Our final foreground detection is built by combining the outputs from these building blocks. The foreground obtained by the long-term modeling block is verified by the output of the short-term modeling block, i.e. only the objects exhibiting significant motion are detected as a real foreground objects. The proposed method results in a very good foreground detection performance at a low computational cost.

Introduction: Visual surveillance systems are successfully used in a number of different applications, such as traffic monitoring, people tracking, etc. An important step in these application is the detection and tracking of moving objects in *real time*. The scene analysis should consume only a few percents of the available processing power and results should be available with small or no delay. However, to separate moving objects from the background scene remains challenging. In this work, we consider the detection of moving objects as a generic preprocessing step, which can be further used for object tracking and behavior analysis. We propose a computationally efficient foreground detection method, which is focused solely on accurate localization of moving objects, both in space and time. The basic assumption of background subtraction methods is a fixed camera position. Therefore, the background scene model is learned over time and moving objects are identified as areas where the current pixel values differ significantly from the current background model [2].

Background subtraction techniques usually impose two conflicting requirements: *accuracy* of the background model and fast *responsiveness* of the model to sudden changes in the background (such as illumination changes). An accurate background model should precisely represent the real background (BG) scene, not affected by foreground (FG) objects and noise. Therefore, such models are usually implemented as *slow-adaptive* algorithms which integrate information about the background over longer periods. A fast-responding model, on the other hand, has to react quickly on the appearance of moving objects and exclude them immediately from the background otherwise false objects are introduced [2]. Such models are typically *fast-adaptive*. However, models which are both accurate and responsive need a trade-off between slow and fast-adaptive background modeling. The optimal rate for a model adaptation is difficult to determine without having previous knowledge on the monitored scenes.

In this paper, we propose a novel algorithm which solves the conflicting requirements by combining two models, with significantly different adaptation speeds. The final segmentation of foreground regions is achieved by a combination of those two models. At first, the slow-adapting model performs accurate detection of moving objects whereas the fast-adapting model validates the detected objects afterwards. We tested the proposed foreground detection method in many surveillance scenarios. The results demonstrate a better detection of foreground objects compared to related methods ([3], [4], [6] and [5]) especially in transient cases, when new events occur in the scene. For example, objects, that are initially part of the background but start moving, as well as moving objects, that stop and become part of the background, are detected more accurately and more quickly by the proposed method. Moreover, our proposed method is more robust to sudden illumination changes.

The main property of the proposed background modeling method is its low computational complexity, which makes it a few times faster than state-of-the-art methods based on Mixture of Gaussian (MoG) models or non-parametric kernel density estimation (KDE) [2].

The Proposed Method: In contrast to existing solutions, we propose combining background subtraction with moving object validation at the pixel level, rather than at the object level. In this way the algorithm is faster, insensitive to the object's size and shape and capable of suppressing ghost objects even if they overlap with the real objects.

Our approach consists of a long-term and a short-term background model, B_k^l and B_k^s, which are based on the computationally efficient running average technique. At first, the i-th pixel of the k-th input frame I_k is compared with the long-term model resulting in a binary mask F_k^l as follows

$$F_k^l\left(i\right) = \mathrm{hyst}\left(\left|\mathrm{I_k}\left(\mathrm{i}\right) - \mathrm{B_k^l}\left(\mathrm{i}\right)\right|\right). \tag{1}$$

Function $\mathrm{hyst}\left(\right)$ denotes a hysteresis thresholding operation on an image. All pixels in I_k having a gray value larger than or equal to T_H are immediately accepted as foreground. These pixels are denoted as secure pixels. Conversely, all pixels with gray values less than $T_L = 0.5T_H$ are immediately rejected. In this way, detecting the noise as foreground is avoided, but some of the foreground pixels are not detected either. In order to include the missed ones, the pixels having gray values in between are accepted as foreground if they are in the neighborhood of secure pixels.

At the same time, the image I_k is compared with the short-term model resulting in a binary mask F_k^s

$$F_k^s\left(i\right) = \begin{cases} 1, & \text{if } \left|I_k\left(i\right) - B_k^s\left(i\right)\right| > T_L \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The thresholding is done using the same threshold T_L as for the mask F_k^l. Only the lower threshold is used because the possible false detection of motion due to noise is not critical. The final foreground mask F_k is found as the intersection of the foreground areas indicated by the masks F_k^l and F_k^s and therefore given by

$$F_k\left(i\right) = F_k^l\left(i\right) F_k^s\left(i\right). \tag{3}$$

The proposed foreground validation can also be seen as an integration of a standard background subtraction approach and a motion detection approach. The first one is realized by a long-term background model and the second one by a short-term background model.

To achieve low memory requirements, we estimate the mean value for B_k^l from previous frames in a recursive manner, using the following relation:

$$B_k^l\left(i\right) = \alpha_k\left(i\right) I_k\left(i\right) + \left(1 - \alpha_k\left(i\right)\right) B_{k-1}^l\left(i\right), \tag{4}$$

where $\alpha_k\left(i\right)$ is a *spatially adaptive* learning rate. We update the long-term model B_k^l selectively, only at the spatial locations where the foreground objects are not detected in I_k. This prevents incorporation of moving objects into the long-term background model. Therefore, we set $\alpha_k\left(i\right) = \left(1 - F_k\left(i\right)\right)\alpha_l$. This implies that no additional frame buffers are needed to store $\alpha_k\left(i\right)$ because α_l is a constant. Selectively updating of B_k^l could cause propagation of false detections in time because the model is not updated in foreground regions. However, this situation is prevented by the short-term background module which works in conjunction with the long-term background module.

The task of the short-term modeling block is to detect areas in the scene where motion appears. However, because of the fact that the long- and the short-term block are used jointly, we do not need to locate moving objects accurately. It is rather important to detect areas containing significant temporal activity. We assume that areas of significant activity contain all moving objects and occasionally some neighboring areas. Our intention is to detect slightly more motion in the scene than it is actually present, rather than leaving out parts of the moving objects from detection. Therefore, the short-term background model B_k^s is updated in a similar way as the long-background model B_k^l given as follows:

$$B_k^s\left(i\right) = \alpha_s I_k\left(i\right) + \left(1 - \alpha_s\right) B_{k-1}^s\left(i\right), \tag{5}$$

where α_s is the learning constant. In contrast to (4), α_s is constant and has larger values than α_l, which is the major difference between the two models. In typical surveillance applications the algorithm is robust to changes of α_s and produces satisfactory results as long as α_s is kept sufficiently large, i.e. in the range $[0.1, 0.5]$. We found experimentally that the learning rate $\alpha_s \approx 10\alpha_l$ produces good results in many scenarios.

Note that the proposed method has only two parameters, since the threshold parameters are connected as well as the learning speed parameters. Therefore, only the parameters T_L and α_s need to be chosen. As a result, the proposed method has the same number of parameters as the MoG background modeling scheme.

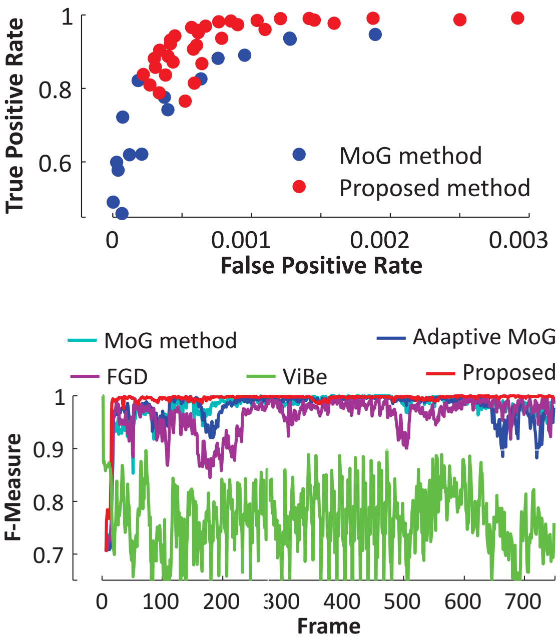

Fig. 1 *Representative results of video6 of the VSSN2006 Background Competition: (a) the overall performance in the ROC space for the MoG method and the proposed method calculated, and (b) the F-measure for MoG, adaptive MoG, FGD, ViBe and the proposed method.*

Experimental Results: We thoroughly compared the performance of the proposed foreground detection method with a few state-of-the-art background modeling methods: the MoG background subtraction technique [3], the adaptive MoG method [4], the foreground detection model (FGD) [6], which is not based upon MoG modeling scheme, and ViBe [5]. Our main target is to develop a low-complexity algorithm, suitable for the real-time implementation, while retaining sufficient quality level. Therefore, the complexity of the proposed method is compared mainly to the MoG method, since the other methods, used in comparison, have significantly higher complexity. The MoG method is well known and theoretically established. Nevertheless, regarding hardware implementation it is also considered as a low-complexity algorithm.

As stated in [1], we measure *sensitivity*, *specificity* and *precision* of the foreground detection methods. The sensitivity represents the proportion of TPs among the correct FG pixels and the missed FG pixels, as sensitivity = TP/ (TP + FN). Whereas the specificity specifies the proportion of TNs among all pixels detected as BG, as specificity = TN/ (TN + FP). Finally, the precision is defined as the proportion of TPs among all pixels detected as FG precision = TP/ (TP + FP). Usually there is a trade-off between sensitivity and precision; obtaining a high sensitivity usually means sacrificing precision and vice versa. Therefore, we used the F-measure which quantifies how similar the obtained FG mask is to the ground-truth.

In Fig. 1(a), we report the overall performance in the Receiver Operating Characteristic (ROC) space, which depicts the dependency between TP rate (*sensitivity*) and FP rate (1-*specificity*) calculated as an average over all frames. Each dot represents a different parameter set for the algorithms. In the MoG method, the number of Gaussians K is set to 5, while the learning speed parameter is varied in range $[0.0005, 0.05]$ and the threshold parameter in range $[0.1, 0.99]$ (see [3]). For our proposed algorithm, α_l is varied in range of $[0.01, 0.05]$, and T_H in range of $[4, 20]$. We can see that both algorithms perform well on average. However, on the sequences the proposed algorithm is able to achieve slightly better TP rate than the MoG method, at the cost of negligible increase in the FP rate. In Fig. 1(b), the evolution of the F-measure over time is shown. The proposed method adapts quickly (in approximately 10 frames). The MoG method, adaptive MoG and FGD method adapt at a similar speed as the proposed method, but achieve slightly lower accuracy. Only ViBe has problems in adapting, and achieves an average performance.

The memory requirements of the proposed algorithm are rather small. We need to store the long-term and short-term background models, B_k^l and B_k^s, and the binary mask F_k. Therefore, the requirements of the algorithm are basically three frames. This is much less than for the MoG method, which needs at least $3 \cdot K$ frames (K has to be at least three [3]). Moreover, the number of calculations needed for a model update is also small (4 multiplications and 2 additions), compared to the MoG method ($2K + 7$ multiplications and $2K + 4$ additions). In addition, the proposed algorithm is approximately four times faster than the MoG method (156 vs. 36 FPS on the VSSN2006 sequences).

The test sequences and corresponding results are publicly available: `http://telin.ugent.be/~ljj/fgs_results/`.

Conclusion: In this paper, we present a novel algorithm for foreground detection. Here, we divide the problem into two simple parts: modeling of the long-term background scene and finding regions containing motion. Both models are combined using autoregressive moving average filter. Therefore the advantage is twofold: at first, it is easy to analyze and predict the behavior of the system, and secondly the algorithm is suitable for hardware implementation due to modest memory and computation requirements. However, this does not result in performance loss. On the contrary, comparing to the state-of-the-art algorithms, the proposed method performs often better compared to the state-of-the-art algorithms, especially in transitional cases, when the background changes.

Acknowledgment: This work has been supported by the Flemish Fund for Scientific Research, through the project “Multi-camera human behavior monitoring and unusual event detection” (G.0.398.11.N.10)

S. Gruenwedel et al. (*Ghent University TELIN-IPI-iMinds, Gent, Belgium*)

E-mail: sebastian.gruenwedel@telin.ugent.be

References

1 Y. Yang: ‘An evaluation of statistical approaches to text categorization’, *Journal of Information retrieval*, 1999, **1**, p. 69-90

2 M. Cristani, M. Farenzena, D. Bloisi and V. Murino: ‘Background Subtraction for Automated Multisensor Surveillance: a Comprehensive Review’, *Journal on Advances in Signal Processing*, 2010, **2010**, p. 24

3 P.W. Power and J.A. Schoonees: ‘Understanding Background Mixture Models for Foreground Segmentation’, *Proc. of Image and Vision Computing*, 2002

4 Z. Zivkovic and F. van der Heijden: ‘Efficient adaptive density estimation per image pixel for the task of background subtraction’, *Pattern Recognition Letters*, 2006, **27**, p. 773-780

5 O. Barnich and M. Van Droogenbroeck: ‘ViBe: a powerful random technique to estimate the background in video sequences’, *Proc. of the IEEE Inter. Conf. on Acoustics, Speech and Signal Processing*, 2009, p. 945-948

6 L. Li, W. Huang, I.Y.H. Gu and Q. Tian: ‘Foreground object detection from videos containing complex background’, *Proc. of the Eleventh ACM Inter. Conf. on Multimedia*, 2003, p. 2–10