SIViP (2013) 7:45–52
DOI 10.1007/s11760-011-0230-z

ORIGINAL PAPER

Single-class SVM for dynamic scene modeling

Imran N. Junejo · Adeel A. Bhutta · Hassan Foroosh

Received: 16 March 2010 / Revised: 9 May 2011 / Accepted: 9 May 2011 / Published online: 28 May 2011

Abstract Scene modeling is the starting point and thus the most crucial stage for many vision-based systems involving tracking or recognition. Most of the existing approaches attempt at solving this problem by making some simplifying assumptions such as that of a stationary background. However, this might not always be the case, as swaying trees or ripples in the water often violate these assumptions. In this paper, we present a novel method for modeling background of a *dynamic* scene, i.e., scenes that contain "non-stationary" background motions, such as periodic motions (e.g., pendulums or escalators) or dynamic textures (e.g., water fountain in the background, swaying trees, or water ripples, etc.). The paper proposes single-class support vector machine (SVM), and we show why it is preferable to other scene modeling techniques currently in use for this particular problem. Using a rectangular region around a pixel, spatial and appearance-based features are extracted from *limited* amount of training data, used for learning the SVMs. These features are unique, easy to compute and immune to rotation, and changes in scale and illumination. We experiment on a diverse set of dynamic scenes and present both qualitative and quantitative results, indicating the practicality and the effectiveness of the proposed method.

Keywords Background subtraction · Scene modeling · Dynamic scene · Single-class classification · Support vector machine

I. N. Junejo (✉)
University of Sharjah, Sharjah, UAE
e-mail: ijunejo@sharjah.ac.ae

A. A. Bhutta · H. Foroosh
University of Central Florida, Orlando, FL, USA

1 Introduction

Intelligent video surveillance has attracted immense attention from both the researchers and the industry at large. Applications such as human tracking, surveillance, segmentation, and automatic analysis of foreground objects are some of the key applications. Most of these tasks, however, involve background modeling. The goal is to extract objects of interest, generally assumed to be *non-stationary* with respect to the created scene model. A robust (and efficient) solution to this initial phase holds the key to an improved vision-based system performing meaningful higher level tasks.

Surveillance systems typically use stationary cameras to monitor an area of interest. Stationarity of the sensor has been the key assumption that has allowed researchers to employ various statistical techniques for scene modeling. It is assumed that generally an *interesting* object will be moving or non-stationary compared with the static scene [22] and an accurate tracking relies on a reliable detection of such objects [7,25]. However, this assumption is often violated in real-world scenarios where many factors, such as windy conditions, often cause the sensor to move or sway slightly. Examples include water ripples, swaying trees, fan, or a water fountain in the background. In addition, a stationary sensor does not guarantee a stationary *background*. It is these periodic or recurrent motions that cause non-stationarity in the scene and prompt the existing methods to recognize these phenomena as interesting.

Previously, in order to model a dynamic scene, a single Gaussian distribution was proposed to model each pixel in the scene. Wren et al. [22] in their seminal work, fitted a single three dimensional Gaussian per pixel, where the model parameters, i.e., mean and the standard deviation, were estimated from the pixels in consecutive frames. However, one Gaussian model per pixel has proved to be ill-suited to capture different underlying processes generating pixel intensities in an outdoor scene. Stauffer et al. [19] thus proposed modeling the scene with a mixture of Gaussians (MoG). A non-parametric kernel density estimation (KDE) was adopted by Elgammal et al. [4] for per pixel background modeling. Both of these methods address nominal camera movements but to a large extent have been unable to handle larger phenomena generating from the background. Ko et al. [10] models the temporal variation of color distribution per pixel. See [1,5,6,9,10,12,15,17,20,26] for a review of other methods.

The work most relevant to that of ours is that of [3,13,18,24]. In their work, Zhang et al. [24] construct a covariance matrix from the region surrounding a pixel by using the spatial and the appearance attributes. The covariance matrix from a new pixel is matched to that of the constructed model to distinguish a foreground from the background using certain threshold values. Monnet et al. [13] addresses the non-stationary background and uses on-line auto-regressive models for scene modeling. Sheikh and Shah [18] models the scene using a single probability density in addition to modeling the foreground. They introduce a *temporal persistence* for accurate detection. The maintained background and the foreground models are used in a Maximum A Posteriori in the Markov Random Field (MAP-MRF) selection framework for performing the object detection in a stationary camera, where the maximum a posteriori solution is obtained by finding the minimum cutoff of the constructed graph. Cheng et al. [3] uses a weighted linear combination of selected past observations with time decay to model background. Their work focuses on the temporal dynamics of the pixel processes and ignored the *spatial* properties.

In this paper, we treat the scene or the background modeling problem as a single-class classification (SCC) problem and propose the single-class SVM for scene modeling [23]. SCC aims to distinguish one class of data from the universal set of multiple classes. Without requiring a large amount of data, single-class SVM classify one class of data from the rest of feature space given only positive data by drawing a optimum non-linear boundary of the positive data set in the feature space. In addition, we use a novel set of region-based features to capture the dynamics of the background. These proposed features not only capture the dynamics at each pixel but also capture the spatial context of the region surrounding a pixel. In essence, we combine the use of both spatial and temporal properties to model the scene. A shorter version of this work appeared in [8].

The rest of the paper is organized as follows: Section 1 reviews related work in the field and discusses the proposed approach in context of previous work. The scene modeling step, involving feature extraction and single-class classification using SVM, is presented in Sect. 2. We also provide an algorithm for the proposed technique. Qualitative and quantitative experimental results performed on *fountain* and *railway* sequences are presented in Sect. 3, followed by conclusion in Sect. 4.

2 Scene modeling

In a dynamic scene, every pixel in the image is undergoing a certain periodic or a repetitive change in intensities at each time instance. It is too simplistic to assume that a pixel intensity varies independently of its neighbors [18]. For example, in a typical scene with swaying trees or water ripples, such as Fig. 1, a larger region of the image, not just a single pixel, is involved in the same type of motion. At the same time, there is a temporal continuity in the motion, as in the case of swaying trees, where branches or the leaves move back and forth. Thus, it is essential that both the spatial and the temporal context be captured for an accurate scene modeling.

Let $\{\mathbf{I}(t)\}^{t=1\ldots k}$ be a set of images. In order to model the background, we first compute the optical flow by using Lucas and Kanade method [11] on the whole image using two consecutive frames and generate their representations, i.e., the v_x and v_y components of optical flow such that: $\mathcal{F} = \{v_x, v_y\}$. The idea is to extract a set of features that uniquely capture the dynamics of the scene by using these representations.

2.1 Feature set

Once we have computed the optical flow, for every pixel p_i^t in the image, i.e., the ith pixel in image t, a rectangular region of the size $M \times N$ is used to compute the following set of simple features:

Entropy: The standard way of defining entropy is,

$$h_i = -\sum_{m=1}^{k} \mathcal{P}_i \log(\mathcal{P}_i) \tag{1}$$

where k refers to the number of histogram bins and $\mathcal{P}_i$ refers to the histogram count of $\mathcal{F}_i$ for $M \times N$ region around the ith pixel p_i^t. Generally, this is set to be 5×5 in our experiments. The entropy h is a statistical measure of the randomness that can be used to characterize the flow vectors.

Energy: The energy of flow vectors in an $M \times N$ region surrounding the ith pixel is computed as:

$$e_i = \sum_{u=1,v=1}^{M,N} (\mathcal{F}_i)^2 \tag{2}$$

where $\mathcal{F}_i$ refers to flow vectors as defined above and u, v refer to the pixel location. e measures the energy presented in the flow vectors in an $M \times N$ region around a pixel.

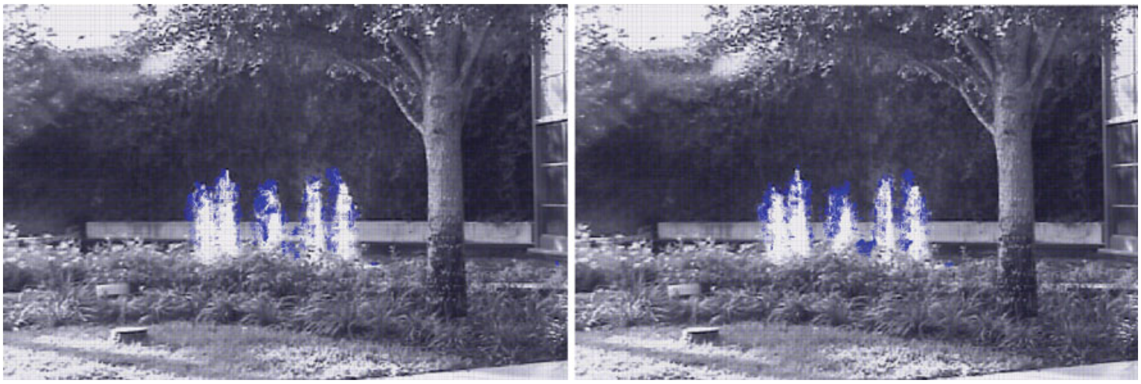

Fig. 1 Sample images from the *fountain* sequence showing the flow vectors super-imposed on the original image. Sequences such as these, contain dynamic and periodic motions, causing challenges for the traditional approaches

Inertia: Finally, we define the inertia as,

$$j_i = \sum_{u=1,v=1}^{M,N} (u-v)^2 \mathcal{F}_i \tag{3}$$

where $\mathcal{F}_i$ refers to flow vectors and u, v refer to the pixel location. j measures an object's resistance to changes in its rotation rate.

The features defined above are unique and yet simple to compute. Entropy, inertia, and energy are relatively immune to *rotation*, since the order is not important. These measures are *scale* invariant and are inherently invariant to linear change in *illumination* as well. The output at this stage is a 6-dimensional feature vector

$$H^{p_i^t} = \{\{h_i, e_i, j_i\}_{v_x}, \{h_i, e_i, j_i\}_{v_y}\} \tag{4}$$

for every pixel p_i^t in the frame t. This feature vector represents a set of features that uniquely capture the dynamics of the scene.

2.2 Single-class classification

The scene modeling problem involves observing a scene which is assumed to contain an acceptable behavior. During this phase, which is generally termed as the training phase, it is possible to only gather the *positive data* that describes what belongs to the scene. However, during this phase, it is not possible to include the negative data that are to be detected at a later time. This scenario is a good candidate for applying the single-class classification techniques.

Given a limited amount of training data, the *optimal* class boundary function is the one that gives the best generalization performance representing the performance on unseen examples. For supervised learning, SVM tries to maximize the generalization by maximizing the margin and supports nonlinear separation using advanced kernels; thus, avoiding underfitting and overfitting [23].

Algorithm 1 Scene modeling using single-class SVM

```
1 Train SVM:
    * Using training sequence, generate flow components, $\mathcal{F}_i = \{v_x, v_y\}$ for each pixel
    * For each pixel $p_i^t$, compute $H^{p_i^t}$ as defined in (4)
2 Testing:
    * For a test sequence, generate flow components, $\mathcal{F}_i = \{v_x, v_y\}$ for each pixel
    * In a 5 × 5 window around each pixel $p_i^t$, compute $H^{p_i^t}$ as defined above
    * Detect foreground and background pixels using SVMC framework
```

More specifically, we adopt the Support Vector Mapping Convergence (SVMC) as proposed by Yu [23], which employs the Mapping Convergence framework where the algorithm generates the boundary close to the optimum. As the sample size increases, SVMC prevents training time from increasing dramatically, and the running time is shown to be asymptotically equal to that of a SVM. The approach is to use minimally required data at each iteration so that the data does not degrade the accuracy of the boundary. In their work, Yu [23] prove that the training time is $O(n^2)$, where n is the size of the training data. Thus, for training on data set of size K images, we compute the feature vector $H^{p_i} = \{H^{p_i^1}, H^{p_i^2}, \ldots, H^{p_i^K}\}$ for each pixel location. This feature vector is used to train the SVMC at each pixel location. The complete algorithm is given in Algorithm 1.

SVMC has been shown to have a good accuracy for single-class classification by computing accurate classification boundary around the positive data (during the training phase) using the unlabeled data in a systematic way. Moreover, SVMC does not require a large amount of positive training data while still maintaining performance close to that of original SVM while providing good generalization, as the results in the next section show.

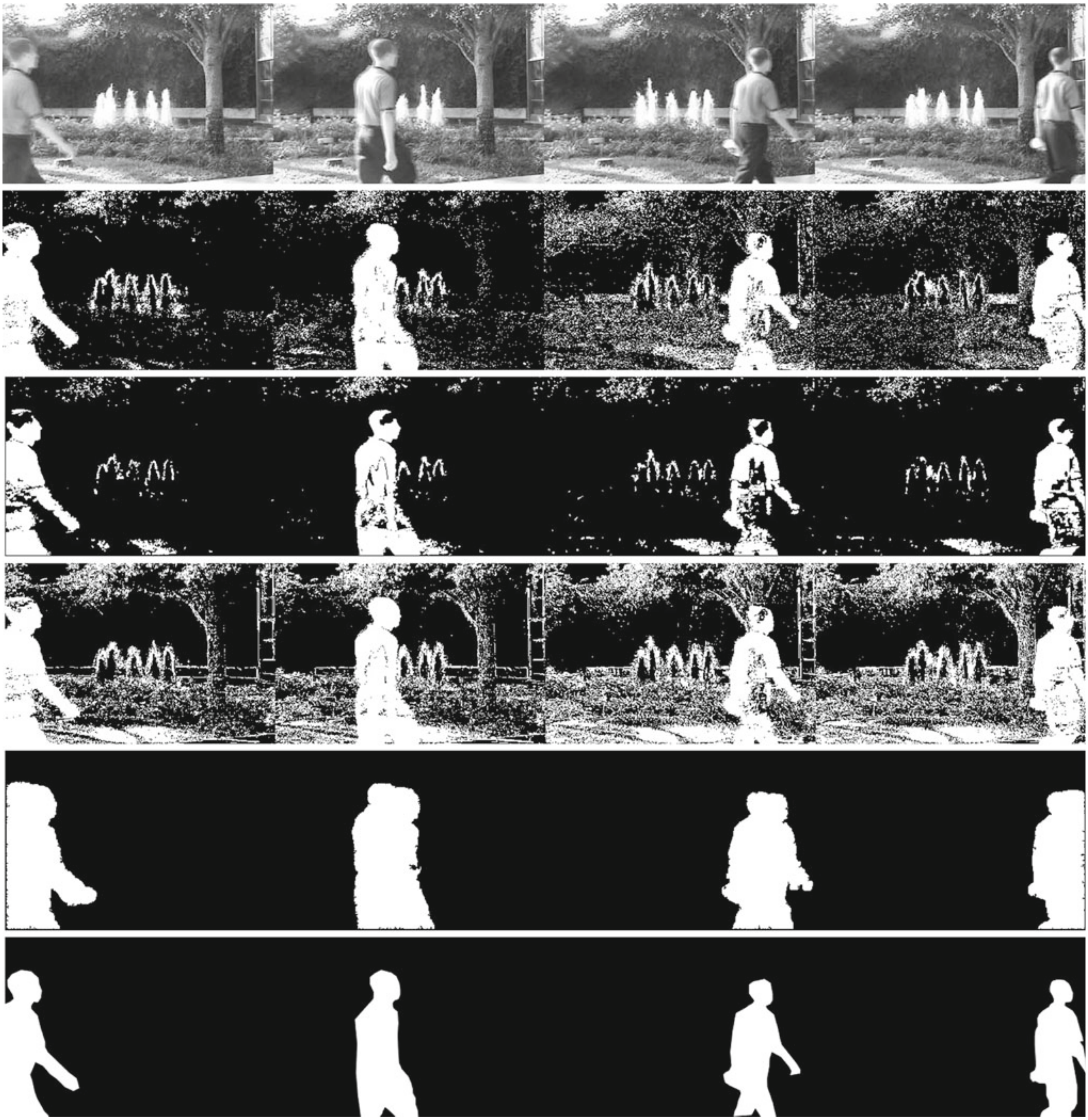

Fig. 2 Experimental results obtained from the *fountain* sequence. The *first row* shows the original images followed the results obtained by the mixture of Gaussian approach, shown *in row 2* (trained on 400 frames). The results from median-based and principle component analysis-based approaches are shown in *rows 3* and *4*, respectively. The results obtained from the proposed method are shown in *row 5* (using 75 frames only for training) and the ground truth subtraction results are shown in the *last row*. No post-processing was performed on these results

3 Experiments and results

We tested the proposed method on two dynamic natural scenes from [18]; from here on, we will refer to them as the *fountain* and the *railway* sequence. These sequences contain nominal camera motion, significant dynamic textures, and cyclic motion. In the *fountain* sequence, the dynamic texture is induced in the scene by the moving trees while the fountain in the background induces constant cyclic motion. The *railway* sequence contains periodic self-occlusion of a walking person followed by occlusion by a passing car. We compare the proposed method with the mixture of Gaussian

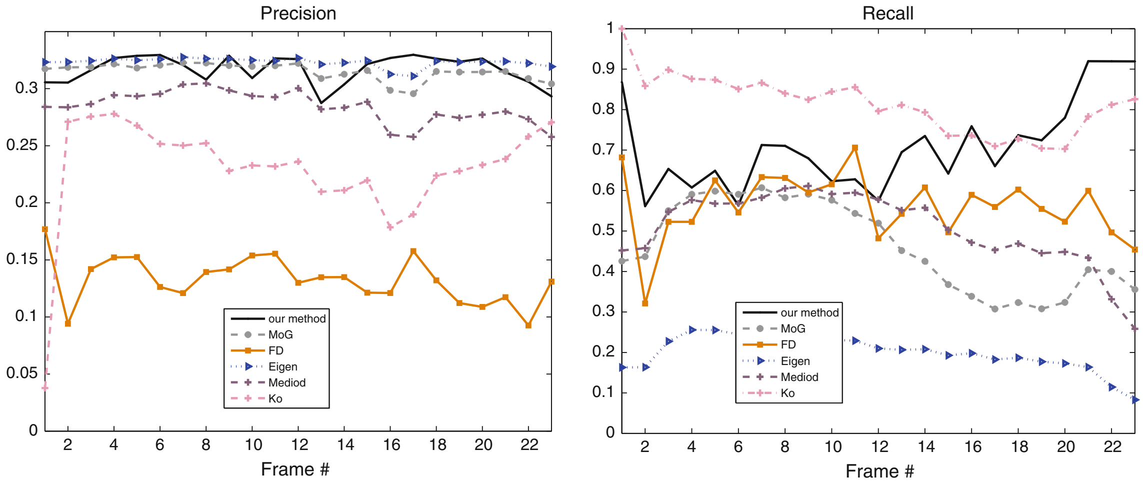

Fig. 3 Comparison of the proposed method with the state of the art MoG method [19] as well as Eigen [14], Medoid [2,16], FD [21], and Ko [10] methods using *fountain* sequence. The figure on the *left* shows the calculated precision for all seven methods while the *right* figure shows the computed recall for these methods. These figures indicate that the precision the proposed method is considerably better than Ko, Medoid, and FD methods while the recall is considerably better than most methods used here

(MoG) approach [19], principle component analysis-based approach (Eigen) [14], median filtering-based approach (Mediod) [2,16], adjacent frame difference (FD) approach [21], and color distribution-based approach [10]. We train MoG model using three (3) components and use 400 and 270 frames for *fountain* and *railway* sequences, respectively. For our method, we only used 75 frames for feature extraction and training the single-class SVM, as described in the Sect. 2. FD method was implemented as proposed in [21] and uses threshold value equal to 0.06. Eigen and Medoid methods are implemented as described in [16] and [14], respectively. The images have a resolution of 360 × 240.

3.1 Qualitative analysis

Qualitatively, the results are an improvement over the methods [2,14,19,21] , as shown in Fig. 2. The camera is mounted on a tall tripod, and the wind causes the tripod to sway back and forth, and in the background is a water fountain and swaying trees. The first row shows the original images from the test sequence. The figure depicts a person coming in from the left of the image and walking to the right. The second row shows the results obtained from the MoG method and it becomes evident that the nominal motion caused by the camera and the presence of the water fountain cause substantial degradation of the results qualitatively. A large number of moving background pixels are detected as foreground pixels. Some portions of the foreground object are also classified as background. The third and fourth rows show the results obtained from median Filtering-Based approach (Mediod) and principle component analysis-based approach (Eigen), respectively. The results from both Mediod and Eigen approaches show the same behavior as MoG where large number of background pixels are classified as foreground. The fifth row shows the results obtained by the proposed method, showing a considerable improvement over MoG [19], Medoid [2], and Eigen [14]. The last row shows the ground truth frames obtained by manually labeling some frames from the image sequence.

Qualitative results for the *railway* sequence are shown in Fig. 4. This sequence, where camera also moves due to the wind, contains periodic self-occlusion of a walking person followed by occlusion by a passing car with trees swaying due to the wind in the background. For the *railway* sequence, the proposed method demonstrates the qualitative improvements as well which can be seen in the fifth row of Fig. 4. It is important to highlight that the results from FD method were consistently worst than MoG and hence have not been shown above. We, however, have provided the comparison of all seven methods in our quantitative analysis below.

3.2 Quantitative analysis

Quantitative analysis is performed on both sequences and the results obtained from our method are compared with [2,14,19] and [21]. We compute the following two measures for assessing the goodness of the proposed method:

$$\mathtt{Precision} = \frac{\#\ \mathtt{of\ true\ positives\ detected}}{\mathtt{total}\ \#\ \mathtt{of\ positives\ detected}}$$

$$\mathtt{Recall} = \frac{\#\ \mathtt{of\ true\ positives\ detected}}{\mathtt{total}\ \#\ \mathtt{of\ true\ positives}}$$

The detection accuracy, in terms of both the precision and the recall, is considerably higher than FD, MoG, and color distribution- based approaches. As observed from the Figs. 3

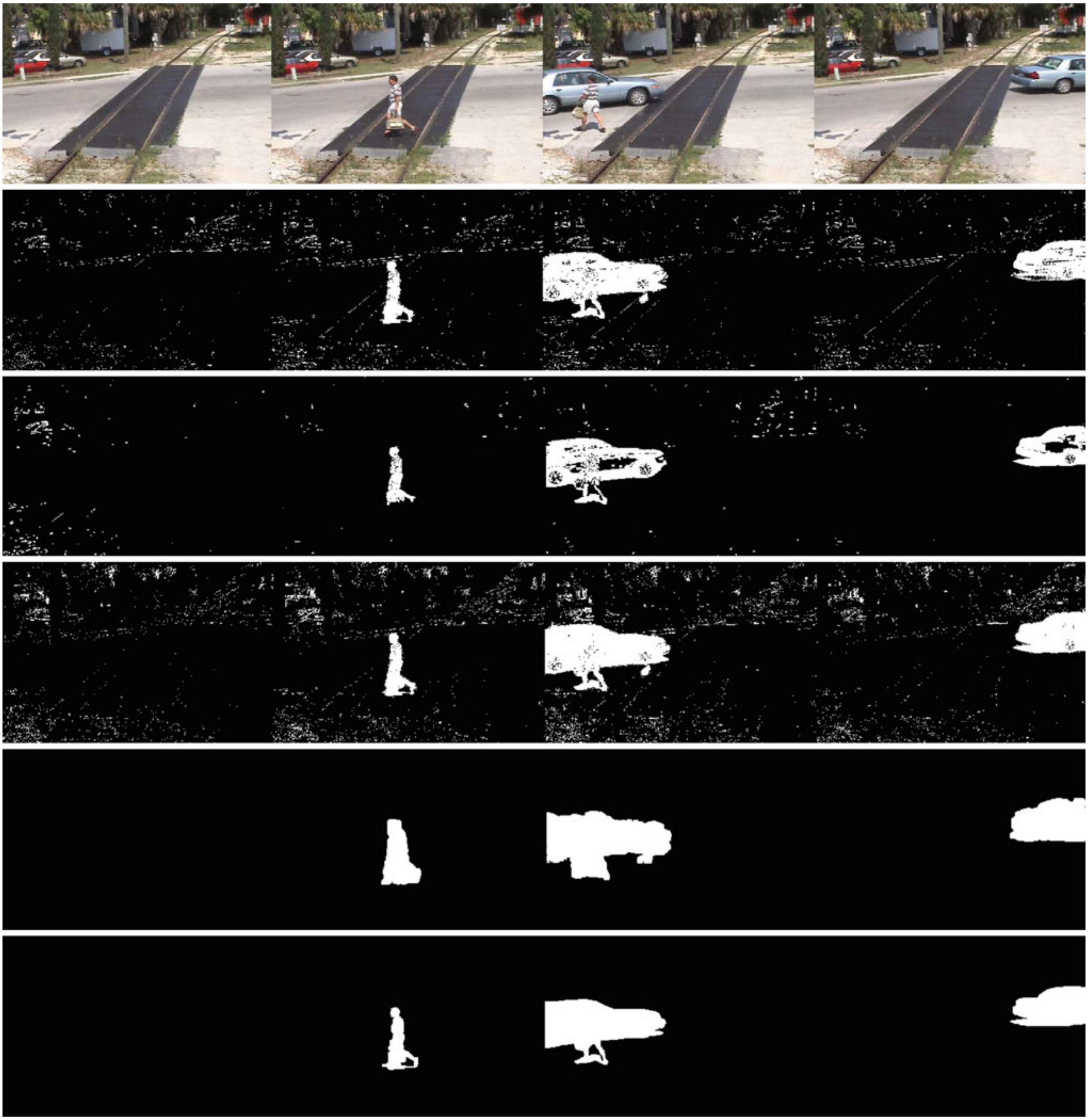

Fig. 4 Experimental results obtained from the *railway* sequence. The *first row* shows the original images followed the results obtained by the mixture of Gaussian approach, shown in *row 2* (trained on 270 images). The results from median filtering and principle component analysis-based approaches are shown in *rows 3* and *4*, respectively. The results obtained from the proposed method are shown in *row 3* (using 75 frames only for training), and the ground truth subtraction results are shown in the *last row*

and 5, the recall rate for the proposed method is consistently high for both sequences, whereas at some instances, the precision decreases due to strong motion in the image sequences. This indicates that the localized foreground is larger than the labeled ground truth, however, the background pixels such as the fountain and the swaying trees are not detected as foreground objects at all. Moreover, we are not using any post-processing techniques, such as graph cuts [18], to improve the boundaries of the foreground objects, which would improve the precision considerably.

It is important to highlight that the results were generated using SVM which was trained using a very small number of

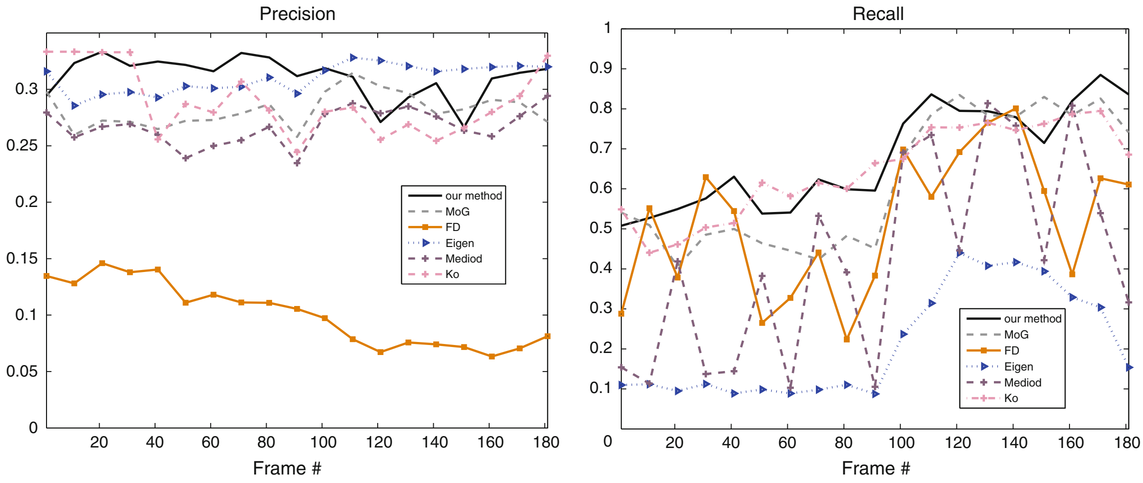

Fig. 5 Comparison of the proposed method with the state of the art MoG method [19] as well as Eigen [14], Medoid [2,16], FD [21], and Ko [10] methods using *railway* sequence. The figure on *left* shows the calculated precision for all five methods while the *right* figure shows the computed recall for these methods. These figures indicate that the precision and recall of the proposed method is considerably better than most methods including FD, Mediod, Eigen, Ko, and the standard MoG. Notice that the precision and recall values are consistently higher indicating that the proposed method outperforms these techniques

Table 1 Number of frames used for training

	Support vector machine	Mixture of Gaussian
Fountain	75	400
Railway	75	270

training images as opposed to MoG. As shown in Table 1, we use 75 frames for training of SVM as opposed to 400 frames for *fountain* and 270 frames for *railway* sequence. It underlines the distinct advantage of the proposed technique over MoG, in cases when the amount of available training data is limited.

4 Conclusion

Scene modeling is a very significant initial step for various vision-based systems. The existing methods often fail for scenes with dynamic textures or cyclic background motion. We propose treating the scene or the background modeling problem as a Single-Class classification problem, and propose using single-class SVM that is able to create the optimal class boundary from a *very limited* set of training examples. We also employ a novel, yet simple region-based features, extracted at each pixel location for training the single-class SVMs. The proposed features not only capture the dynamics at each pixel they also capture the spatial context of the region surrounding a pixel. We have presented results on challenging sequences that contain considerable amount of sensor motion, in addition to a dynamic backgrounds. We compare our results with four standard techniques including mixture of Gaussian, Principle Component Analysis, Mediod filtering and adjacent frame difference, and color distribution-based methods and notice a very significant improvement. The proposed method has successfully minimized false positives and shows considerably higher recall and precision compared with all four approaches without using any post-processing. In addition, we have the distinct advantage of using considerably less training data as opposed to other methods. These encouraging results indicate the practicality and effectiveness of the proposed method.

References

1. Benezeth, Y., Jodoin, P., Emile, B., Laurent, H., Rosenberger, C.: Review and evaluation of commonly-implemented background subtraction algorithm. In: Proceedings of International Conference on Pattern Recognition (ICPR), pp. 1–4, (2008) (8–12 December)
2. Calderara, S., Melli, R., Prati, A., Cucchiara, R.: Reliable bacground suppression using complex scenes. In: Proceedings of ACM International Workshop on video surveillance and sensor networks, pp. 211–214 (2006)
3. Cheng, L., Wang, S., Schuurmans, D.: An online discriminative approach to background subtraction. In: Proceedings of International Conference on Video and Signal Based Surveillance (2006)
4. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.S., Duraiswami, R., Harwood, D.: Background and foreground modeling using nonparametric kernel denity for visual surveillance. In: Proceedings of the IEEE, pp. 1151–1163 (2002)
5. Friedman, N., Russell, S.: Image segmentation in video sequences: a probalistic approach. Uncertainty in Artificial Intelligence, (1997)

6. Grimson, W., Stauffer, C.: Adaptive background mixture models for real time tracking. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR), pp. 246–252 (1999)
7. Javed, O., Shah, M.: Tracking and object classification for automated surveillance. In: Proceedings of European Conference on Computer Vision (ECCV) (2002)
8. Junejo, I., Bhutta, A.A., Foroosh, H.: Scene modeling for object detection using single-class svm. In: Proceedings International Conference on Image processing (ICIP), pp. 1541–1544 (2010)
9. Karmann, K., Brandt, A.V.: Moving object recognition using an adaptive background memory. Time-Varying Image Processing and Moving Object Recognitionm. pp. 289–307. Elsevier, Amsterdam (1990)
10. Ko, T., Soatto, S., Estrin, D.: Background subtraction on distributions. In: Proceedings of European Conference on Computer Vision (ECCV), pp. 276–289 (2008)
11. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Image Understanding Workshop, pp. 121–130 (1981)
12. Mittal, A., Paragios, N.: Motion-based background subtraction using adaptive kernel density estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 302–309 (2004)
13. Monnet, A., Mittal, A., Paragios, N., Ramesh, V.: Background modeling and subtraction of dynamic scenes. In: Proceedings of International Conference on Computer Vision (ICCV), pp. 1305–1312 (2003)
14. Oliver, N., Rosario, B., Pentland, A.: A bayesian computer vision system for modeling human interactions. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **22**, 831–843 (2000)
15. Pless, R., Larson, J., Siebers, S., Westover, B.: Evaluation of local Pattern Recognition (CVPR), pp. 73–78 (2003)
16. Prati, A., Mikic, I., Trivedi, M., Cucchiara, R.: Detecting moving shadows: algorithms and evaluation. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **25**(7), 918–923 (2003)
17. Ren, Y., Chua, C., Ho, Y.: Motion detection with nonstationary background. Mach. Vis. Appl. **13**(5–6), 332–343 (2003)
18. Sheikh, Y., Shah, M.: Bayesian modeling of dynamic scenes for object detection. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **27**(11), 1778–1792 (2005)
19. Stauffer, C., Eric, W., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **22**, 747–757 (2000)
20. Tian, Y., Lu, M., Hampapur, A.: Robust and efficient foreground analysis for real-time video surveillance. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR), pp. 1182–1187 (2005)
21. Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: principles and practices of background maintenance. In: Proceedings of International Conference on Computer Vision (ICCV), pp. 255–261 (1999)
22. Wren, C., Azarbayejani, A., Darell, T., Pentland, A.: Pfinder: real-time tracking of human body. IEEE Trans. Pattern Anal. Mach. Intell. (PAMI) **19**, 780–785 (1997)
23. Yu, H.: Single-class classification with mapping convergence. Mach. Learn. **61**(1–3), 49–69 (2005)
24. Zhang, S., Yao, H., Liu, S., Chen, X., Gao, W.: A covariance-based method for dynamic background subtraction. In: Proceedings of International Conference on Pattern Recognition (ICPR), pp. 1–4 (2008)
25. Zhao, T., Aggarwal, M., Kumar, R., Sawhney, H.: Real-time wide area multi-camera stereo tracking. In: Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR), pp. 976–983 (2005)
26. Zhong, J., Sclaroff, S.: segmenting foreground objects from a dynamic textured background via a robust kalman filter. In: Proceedings of International Conference on Computer Vision (ICCV), pp. 44–50 (2003)