Proceedings of the IEEE Symposium on Image Analysis and Interpretation,
pages 144-148, Denver, Colorado, March 2006.

Automatic Statistical Object Detection for Visual Surveillance

Alireza Tavakkoli, Mircea Nicolescu, and George Bebis
Computer Vision Laboratory
Department of Computer Science and Engineering
University of Nevada, Reno, USA 89557
{tavakkol, mircea, bebis}@cse.unr.edu

Abstract

Detection and tracking of foreground objects in a video scene requires a robust technique for background modeling. Modeling issues such as noise robustness, adaptation and model accuracy must be addressed while allowing for the automatic choice of relevant parameters. In this paper three major contribution are presented. First, the representative background model is a general multivariate kernel density estimation to address the model accuracy issue as well as capturing color dependencies without any knowledge about the underlying probability density of the pixel colors. Second, a single-class classifier is trained adaptively and independently for each pixel, using its estimated densities during the training stage. Finally, noise robustness is achieved by enforcing spatial consistency of the background model.

1. Introduction

In visual surveillance systems, stationary cameras are typically used. However, due to camera shake or inherent changes in the background itself such as fluctuations in monitors, waving flags and trees, water surfaces etc., the background of the video may not be completely stationary. In these types of backgrounds, referred to as dynamic or quasi-stationary backgrounds, a single background frame is not useful to detect moving regions. Pless *et al.* [6] evaluated different models for dynamic backgrounds. Typically background models are defined independently on each pixel and, depending on the complexity of the problem, use the expected pixel features (i.e. colors) [1] or consistent motion. Also they may use pixel-wise information [9] or regional models of the features [8].

In [9], a single 3-dimensional Gaussian model for each pixel in the scene is built, where the mean and covariance of the model were learned in each frame. Kalman Filtering [3] is also used to update the model. These background models were unable to follow and represent multi-modal situations. A Mixture of Gaussians modeling technique was proposed in [7] and [2] to address the multi-modality of the underlying background. There are several shortcomings for the mixture learning methods. First of all, the number of Gaussians needs to be specified. Second, these methods do not specifically deal with spatial dependencies. Also, even with the use of incremental-EM, the parameter estimation and its convergence is noticeably slow where the Gaussians adapt to a new cluster. A recursive filter formulation is proposed by Lee in [4]. However the problem of specifying the number of Gaussians as well as the adaptation in later stages still exists. Also this model does not account for the situations where the number of Gaussians change due to occlusion or uncovered parts of the background.

In [1], El Gammal *et al.* proposed a non-parametric kernel density estimation for pixel-wise background modeling without making any assumption on its probability distribution. Therefore, this method can easily deal with multi-modality in background pixel distributions without determining the number of modes in the background. However there are several issues to be addressed using non-parametric kernel density estimation. First, the non-parametric KDE methods are pixel-wise techniques and do not use the spatial correlation of the pixel features. In order to adapt the model a sliding window is used in non-parametric methods. However the model convergence is critical in situations where the illumination suddenly changes. Second, in [1] a single threshold is used to detect foreground regions which is determined heuristically and it is not adaptive to different changes in the scene. And finally, the model in traditional non-parametric techniques does not enforce spatial consistency of the model explicitly.

In this paper we propose a generalized and adaptive multi-variate nonparametric density estimation. There are three major contributions presented in our proposed method. (i) Dependencies between the pixel features are exploited in our implementation, resulting in more accurate models. In [1], the KDE is used by assuming that the color features used for each pixel are independent, and therefore

the kernel covariance matrix is assumed to be diagonal. The observations show that RGB color space is not independent, and therefore we need to use a general covariance matrix for kernel to capture the color dependencies. (ii) In the proposed method instead of a global threshold for all the pixels in the scene an independent threshold is trained over time to effectively perform the classification. (iii) We use the spatial correlation of the neighboring pixels to achieve the spatial consistency of the background and foreground models.

The rest of this paper is organized as follows: in Section 2 we present the building block of the proposed background modeling technique and we explain how the model is trained to incorporate the dependencies between features. In Section 3, classification as well as enforcing the spatial consistency and adaptation of the neighboring models are discussed. In Section 4 the experimental results of the proposed method are presented and the performance of this method is compared with existing techniques. Finally the conclusion of this paper is drawn in Section 5 and its future directions are discussed.

2. The Proposed Algorithm

Figure 1 shows the pseudo-code for the proposed algorithm, consisting of three major parts: training, classification and update. The first and most important part of the algorithm is the training stage. In this stage the background model is generated, and for each pixel its model values are used to estimate the probability of that pixel in new frames being background. The propose method detects foreground regions by solving a classification problem. However, notice that we only have samples of background class, before any foreground appears in the scene. In this paper we present an automatic, adaptive and robust method to train this classifier.

In the proposed technique we build a non-parametric kernel density estimation classifier for each pixel. This classifier uses a history of the pixel value as training samples and estimates the probability of that pixel in new frames as the classification criteria. In the classification stage, the pixel is classified as foreground or background based on its estimated probability, computed as:

$$P_t(\mathbf{x}) = \frac{1}{N2\pi|\mathbf{\Sigma}|^{1/2}} \sum_{i=1}^{N} e^{\left[-\frac{1}{2}(\mathbf{x}_t-\mathbf{x}_i)^T\mathbf{\Sigma}^{-1}(\mathbf{x}_t-\mathbf{x}_i)\right]} \quad (1)$$

where $\mathbf{x}_t$ and $\mathbf{x}_i$ are the feature vector of the pixel at time t and its history in the training sequence. $\mathbf{\Sigma}$ is a positive definite symmetric matrix which is the kernel bandwidth matrix and N is the number of frames that we use to train the background model. In order to capture the dependencies between features for each pixel $\mathbf{\Sigma}$ has to be a full matrix as opposed to existing methods that assume it to be diagonal.

```
1. Initialization:
2. For each new frame at time t:
  2.1. Training stage:
   for each pixel (i,j):
    - Calculate kernel covariance
    - Calculate threshold: Th_ij
  2.2. Testing stage:
   for each pixel (i,j):
    - Estimate probability
    - Compute median of
      probabilities in its
      neigborhood: Med_ij
    - if (Th_ij ≥ Med_ij)
      I(i,j,t) = 1  % (FG Mask)
    - else
      I(i,j,t) = 0  % (BG Mask)
  2.3. Update:
    - if (i>N-1): i=1
    - else: i = i+1
    - if (size(FG)>0.5size(Image))
      training frame_i ← I
    - else
      training frame_i^BGMask ← I^BGMask
3. Proceed to next frame
```

Figure 1. The proposed background modeling algorithm.

Due to limited memory and computational power, we need to store a rather short memory of the background frames as the training samples. This makes the non-parametric kernel density estimation dependent on the choice of its kernel bandwidth. In order to achieve an accurate and automatic background model, which is adaptive to the spatial information in the scene such as different changes in the background, we need to train the kernel bandwidth matrix. By training Σ for each pixel independently, we automatically update the classifier for each pixel.

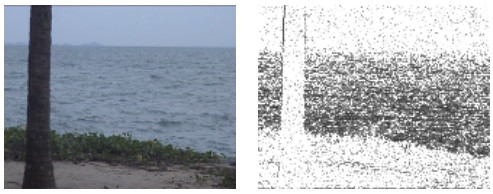
For each pixel the training samples are vectors $\mathbf{X}_N = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, consecutively. To use the temporal information in this sequence of samples, we define the successive deviation of the above vectors as:

$$\Delta_X = \{\mathbf{y}_i | \mathbf{y}_i = \mathbf{x}_i - \mathbf{x}_{i-1}; i = 1, 2, \cdots, N\} \quad (2)$$

For each pixel, the kernel bandwidth matrix is defined such that it represents the scatter of the training samples, by time. Thus the kernel bandwidth is defined by:

$$\Sigma = \text{cov}(\Delta) = (\Delta - \mu_\Delta)(\Delta - \mu_\Delta)^T \quad (3)$$

where μ_Δ is the mean of successive deviation matrix. From equations (2) and (3) it can be seen that for pixels



(a) Arbitrary frame (b) Threshold map

Figure 2. Adaptive threshold map

with more feature changes through time, such as flickering pixels, the kernel bandwidth matrix has larger elements, while for pixels that do not change much, its elements are smaller. Also notice that the kernel bandwidth is drawn from the training samples without any assumption of features and their underlying probability density function. The estimated probability density function by using this adaptive kernel bandwidth is more accurate, even with small number of background frames as the training samples.

In the traditional foreground detection techniques, usually the foreground regions are detected by comparing the values or model of each pixel with its values or models in the background, and if this deviation is larger than a heuristically selected threshold it is selected as a foreground region. If we estimate the probability of each pixel in all of the background frames, as all of these pixels are background, their probabilities should have large values, close to 1. But because of noise and inherent background changes, the pixels do not take a single value, hence their probabilities become smaller. The suitable probability of a pixel to be a background is related to the amount of changes that its features undergo by time. Therefore a single global threshold does not work quite well, because pixels in the scene experience different amounts of change.

This can be seen in Figure 2, where (a) shows an arbitrary frame of a sequence containing water surface and (b) shows the threshold map for this frame. Darker pixels in Figure 2 (b) represent smaller threshold values, and lighter pixels are corresponding to larger threshold values. As it can be observed, the thresholds in the areas that tend to change more, such as water surface, are lower than in those areas with less amount of change, such as the sky. Thus we need to train these threshold values for each pixel during the training stage, to build an accurate and automatic classifier. For each pixel, a value such that 95% of estimated probabilities are higher than that is found. This value is selected to be the threshold for that pixel.

3. Classification and Adaptation

In this section the classification stage of the system is discussed, as well as two types of adaptation approaches we propose for gradual and sudden changes in the scene.

3.1. Classification and Spatial Consistency

After the training stage, for each pixel we have $\mathbf{x}_1, \cdots, \mathbf{x}_N$ vectors of its features in the background training sequence , Σ_{ij} its kernel bandwidth matrix and the Th_{ij} its classification decision criterion. The probability of each pixel in the new frame is estimated using equation (1) with its corresponding trained values. In order to classify the pixels into foreground and background, we compare their estimated probability with their trained threshold. However, if we directly apply the trained threshold of each pixel to its estimated probability, due to strong noise, pixels may be erroneously classified. This happens when one pixel has been affected by noise, and the amount of noise is so strong that changes its feature vector $\mathbf{x}_t$, and thus its estimated probability affects the decision results.

One of the properties of this type of noise is that, if strong noise affects a pixel, it is less likely to affect its neighborhood with the same strength. If a pixel in a region belonging to background produces a fairly small probability because of noise, its neighboring pixels are expected to produce larger probabilities. Thus, using the median of the estimated probabilities in a region around a pixel enforces the spatial consistency of its neighborhood. After estimating the probability of each pixel in new frame, the median of the probabilities in its neighborhood is compared with it threshold to make the classification decision:

$$\text{Mask}_{ij}^{t} = \begin{cases} 1 & \text{med}\big(Prob_{ij}^{t} \geq Th_{ij}\big) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

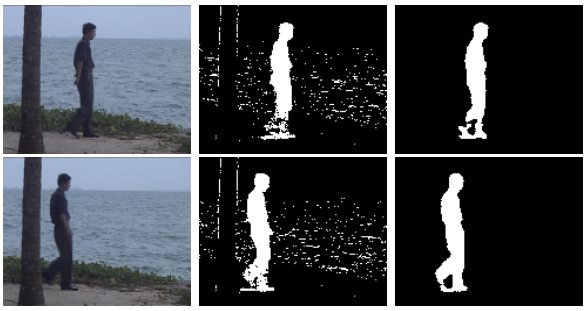
3.2. Adaptation

In order to make the system adapt to gradual and sudden changes in the scene two procedures are devised. The adaptation to sudden change adaptation checks for the size of detected foreground regions. If the size of the foreground is too large; (i.e., more than 50% of the image size), the system detects that there is a major change in the scene. Then the oldest background frame in the training dataset is replaced with current frame. If such change in foreground masks is not detected, the system makes the gradual change adaptation. In this stage, only the pixels in the oldest background frame which do not belong to the current foreground mask are replaced with their corresponding pixels from the current frame.

4. Experimental Results and Comparison

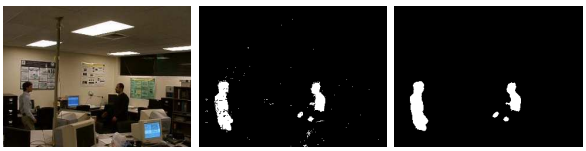
In this section the experimental results of the proposed method are evaluated and compared with current existing methods, both quantitatively and qualitatively.

Irregular motion. By using the *water surface* video sequence in Figure 3, we compare the results of foreground



(a) (b) (c)

Figure 3. Comparison of the foreground masks detected by KDE (b), and our method (c).



(a) (b) (c)

Figure 4. Accuracy of the proposed algorithm.

(a) Meeting room (b) Water (c) Campus

Figure 5. Result of the proposed foreground region detection.

Figure 6. Result of long-term detection using the proposed method.

region detection using our proposed method with a typical non-parametric kernel density estimation [1]. Column (a) shows the original frames of the video, while columns (b) and (c) show the results of the non-parametric method and our proposed technique, respectively. As it can be observed in Figure 3, the proposed method gives more accurate foreground masks which are spatially more consistent.

Detection accuracy. The foreground detection accuracy in low contrast video sequences is checked by using the *Handshake* video in Figure 4. Figure 4 (a), shows the original frame of the video sequence. In Figure 4 (b), the foreground masks detected by the non-parametric density estimation, with diagonal kernel bandwidth matrix and a heuristically selected global threshold are shown and Figure 4 (c) shows the result of the proposed automatic robust method. The accuracy of the detected masks using the proposed method can be observed in those areas where the features of foreground object (the person to the left) pixel values are similar to the background pixel values.

Challenging environments. Shown in Figure 5 are the results of our foreground detection method on several challenging video sequences. In Figure 5(a), the *Meeting room* video, an indoor situation with moving blinds is shown. In Figure 5(b), the *Water* video sequence, there are waves and rain drops in the background of the scene generating a dynamic texture and in Figure 5(c), the *Campus* video, there are waving trees as a typical outdoor scenario causing an irregular dynamic pattern in the background. In all of the above cases, our proposed method is able to detect the foreground regions accurately and ignores the background movements.

Long-term detection. In Figure 6, the results of a long term foreground region detection on the *Lobby* video sequence, using the proposed method are shown. As it can be seen from the figure, foreground regions are detected with the same accuracy throughout time. This shows that the system is able to adapt to the gradual and sudden changes that may occur, avoiding the degradation of the detected foreground.

Quantitative evaluation. The performance of our proposed method is evaluated quantitatively on randomly selected samples from different video sequences, taken from [5]. The similarity measure between two regions $\mathcal{A}$ and $\mathcal{B}$ is defined by $\mathcal{S}(\mathcal{A},\mathcal{B}) = \frac{\mathcal{A}\cap\mathcal{B}}{\mathcal{A}\cup\mathcal{B}}$. This measure is monotonically increasing with the similarity of the detected masks to the ground truth, with values between 0 and 1. We calculated the average of similarity measure of the foreground masks detected by our proposed method, the Mixtures of Gaussians in [7] and the method proposed in [5].

By comparing the average of the similarity measure over different video sequences in Table 1, we can see that the proposed method outperforms the technique proposed in [7]. As it can be seen from the first and second rows of the table, the results of [5] are better than the proposed method.

Table 1. Quantitative evaluation and comparison. The sequences are Meeting Room, Lobby, Campus, Side Walk, Water Surface and Fountain, from left to right from [5].

Videos	MR	LB	CAM	SW	WS	FT	Avg
Proposed	0.74	0.66	0.55	0.60	0.84	0.51	0.63
[5]	0.91	0.71	0.69	0.57	0.85	0.67	0.74
[7]	0.44	0.42	0.48	0.36	0.54	0.66	0.49

However, in [5] the foreground masks are refined by a morphological post-processing step, while the first row shows the unprocessed results of our proposed method. After using a morphological process on the detected masks from our method they became more similar to the ground truth masks and the average of similarity is slightly more than 0.74. However, we should emphasize that in our method the thresholds and the covariance matrices for every pixel have been estimated automatically and there is no assumption on the background model or the feature dependencies, while in other methods there are various parameters that have to be adjusted for different environments and applications. Also the parameters for other existing methods need to be selected globally for each video scene, while in our proposed technique the parameters will be estimated locally for each pixel position in the scene from its history. This can also be observed by the fact that performance of the proposed method is more consistent on different video sequences.

All the experiments have been carried out on a Pentium 4 PC, 2.54 GHz, using Matlab 6.5. The initial training stage of the algorithm takes about 4 seconds for 150 frames as the background training buffer with size of 160×120. In order to speed up the training, the updating process retrains only the covariance matrix of the kernels. The retraining process only takes about 0.2 second. The retraining of the threshold map is performed with a lower rate, as it takes longer to be retrained. The foreground region detection takes about 0.3 seconds, therefore our proposed method is able to extract moving objects at the rate of about 3 frames per second.

5. Conclusion and Future Work

In this paper we propose an automatic statistical object detection framework based on a single-class classification technique. Our method contains three major contributions. First, a generalized non-parametric density estimation with a full kernel bandwidth matrix is trained for each pixel to build up a single-class classifier without any assumption on the underlying probability of the data and the dependencies of features. Second, to achieve an accurate and automatic foreground detection, for each pixel a single threshold is trained from the history of its values. Finally, a spatial process is performed on the classification phase to enforce spatial consistency of the model and detected foreground regions. Temporal adaptation is achieved by using a gradual change and a sudden change adaptation stage.

As a major future direction, we are looking into using support vector data descriptors to make a robust and accurate single-class classifier to label pixels in the video as foreground or background. The preliminary results of such technique are promising although it needs more optimization for speed. Another extension to this study is to use online learning techniques, and combine the detection results with tracking information to achieve more robust and accurate foreground masks.

6. Acknowledgements

This work was supported in part by a grant from the University of Nevada Junior Faculty Research Grant Fund and by NASA under grant # NCC5-583. This support does not necessarily imply endorsement by the University of research conclusions.

References

[1] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proceedings of the IEEE*, 90:1151–1163., 2002.

[2] N. Friedman and S. Russell. Image segmentation in video sequences: A probabilistic approach. *Annual Conference on Uncertainty in Artificial Intelligence*, pages 175–181, 1997.

[3] D. Koller, J. W. adn T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russel. Towards robust automatic traffic scene analysis in real-time. *ICPR*, 1:126–131, October 1994.

[4] D.-S. Lee. Effective gaussian mixture learning for video background subtraction. *IEEE Transactions on PAMI*, 27(5):827–832, May 2005.

[5] L. Li, W. Huang, I. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Trans. on Image Processing.*, 13(11):1459–1472, November 2004.

[6] R. Pless, J. Larson, S. Siebers, and B. Westover. Evaluation of local models of synamic backgrounds. *In proceedings of the CVPR*, 2:73–78, June 2003.

[7] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on PAMI*, 22(8):747–757, August 2000.

[8] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: principels and practive of background maintenance. *In proceedings of ICCV*, 1:255–261, September 1999.

[9] C. Wern, A. Azarbayejani, T. Darrel, and A. Petland. Pfinder: real-time tracking of human body. *IEEE Transactions on PAMI*, 19(7):780–785, July 1997.