International Journal on Artificial Intelligence Tools
Vol. 17, No. 4 (2008) 635–658

World Scientific
www.worldscientific.com

A SUPPORT VECTOR DATA DESCRIPTION APPROACH FOR BACKGROUND MODELING IN VIDEOS WITH QUASI-STATIONARY BACKGROUNDS

ALIREZA TAVAKKOLI
Department of Computer Science and Engineering, University of Nevada, Reno, Reno, Nevada 89557 USA
tavakkol@cse.unr.edu

MIRCEA NICOLESCU
Department of Computer Science and Engineering, University of Nevada, Reno, Reno, Nevada 89557 USA
mircea@cse.unr.edu

GEORGE BEBIS
Department of Computer Science and Engineering, University of Nevada, Reno, Reno, Nevada 89557 USA
bebis@cse.unr.edu

MONICA NICOLESCU
Department of Computer Science and Engineering, University of Nevada, Reno, Reno, Nevada 89557 USA
monica@cse.unr.edu

Received 12 January 2007
Accepted 11 December 2007

Video segmentation is one of the most important tasks in high-level video processing applications. Stationary cameras are usually used in applications such as video surveillance and human activity recognition. However, possible changes in the background of the video such as waving flags, fluctuating monitors, water surfaces, etc. make the detection of objects of interest particularly challenging. These types of backgrounds are called quasi-stationary backgrounds. In this paper we propose a novel, non-statistical technique to generate a background model and use this model for background subtraction and foreground region detection in the presence of such challenges. The main advantage of the proposed method over the state of the art is that unlike statistical techniques the accuracy of foreground regions is not limited to the estimate of the probability density. Also, the memory requirements of our method are independent of the number of training samples. This makes the proposed method useful in various scenarios including the presence of slow changes in the background. A comprehensive study is presented on the efficiency of the proposed method. Its performance is compared with various existing techniques quantitatively and qualitatively to show its superiority in various applications.

Keywords: Background modeling; background subtraction; quasi-stationary backgrounds; support vector data description; support vectors.

(a) (b) (c)

Fig. 1. Examples of challenges in quasi-stationary backgrounds: (a) Fluctuating monitors. (b) Rain/Snow. (c) Waving tree branches.

1. Introduction

Detecting foreground regions in videos is one of the most important tasks in high-level video processing applications. One of the major issues in detecting foreground regions using background subtraction techniques is that because of inherent changes in the background, such as fluctuations in monitors and fluorescent lights, waving flags and trees, water surfaces, etc. the background may not be completely stationary. These difficult situations are illustrated in Fig. 1.

In the presence of these types of backgrounds, referred to as quasi-stationary, a single background frame is not enough to accurately detect moving regions. Therefore the background of the video has to be modeled in order to detect foreground regions which are newly introduced objects to the scene, while allowing for quasi-stationary backgrounds.

There is also a great amount of diversity in scenarios where the background modeling techniques are used to detect foreground regions. Applications vary from indoor scenes to outdoor, from completely stationary to dynamic backgrounds, from high quality videos to low contrast scenes and so on. Therefore, a single system that addresses all possible situations while being time and memory efficient is yet to be devised.

Pless *et al.*[28] evaluated different models for dynamic backgrounds. Typically, background models are defined independently on each pixel, and depending on the complexity of the problem employ the expected pixel features (i.e. colors)[7,31] or consistent motion.[27,49] They also may employ pixel-wise information[48] or regional models of the features.[46,10,20] To improve robustness to noise, spatial[24] or spatio-temporal[19] features may be used.

In Ref. 48 a single 3-D Gaussian model for each pixel in the scene is built, where the mean and covariance of the model are learned in each frame. This system tried to model the noise and used a background subtraction technique to detect those pixels whose probabilities are smaller than a threshold. However, the system fails to label a pixel as foreground or background when it has more than one modality due to fluctuations in its values, such as a pixel belonging to a fluctuating monitor.

Kalman Filtering[16,13,14] is also used to update the model, while linear prediction using Wiener filtering is presented in Ref. 46. These background models were unable to represent multi-modal situations.

A mixture of Gaussians modeling technique was proposed in Refs. 35, 34 and 8 to address the multi-modality of the underlying background. In this modeling technique background pixels are modeled by a mixture of a number of Gaussian functions. During the training stage, parameters of each Gaussian and their weights are trained and used in the background subtraction, where the probability of each pixel is generated using the mixture of Gaussians. The pixel is labeled as foreground or background based on its probability.

There are several shortcomings for mixture learning methods. First, the number of Gaussians needs to be specified. Second, this method does not explicitly handle spatial dependencies. Even with the use of incremental-EM, the parameter estimation and its convergence is noticeably slow where the Gaussians adapt to a new cluster. The convergence speed can be improved by sacrificing memory as proposed in Refs. 21 and 22, limiting its applications where mixture modeling is pixel-based and over long temporal windows.

A recursive filter formulation is proposed by Lee in Ref. 18 to speed up the convergence. However, the problem of specifying the number of Gaussians as well as the adaptation in later stages still exists. Moreover, this model does not account for situations where the number of Gaussians changes due to occlusion or uncovered parts of the background.

In Ref. 7, El Gammal *et al.* proposed a non-parametric kernel density estimation method (KDE) 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 specifying the number of modes in the background. However, there are several issues to be addressed using non-parametric kernel density estimation.

These methods are memory and time consuming since the system has to compute the average of all kernels centered at each training sample for each pixel in each frame. Also the size of temporal window used as the background model needs to be specified. Too small a window increases speed, while it does not incorporate enough history for the pixel, resulting in a less accurate model. Another issue is that the adaptation will be problematic by using small window sizes. Increasing the window size improves the model accuracy but at the cost of higher memory requirements and slower convergence. Finally, the non-parametric KDE methods are pixel-wise techniques and do not use spatial correlation of pixel features. In order to adapt the model a sliding window is used in Ref. 23. However the model convergence is problematic in situations where the illumination suddenly changes.

In order to update the background for scene changes such as moved objects, parked vehicles or opened/closed doors, Kim *et al.* in Ref. 15 proposed a layered modeling technique. This technique needs an additional model called *cache* and

assumes that the background modeling is performed over a long period of time. It should also be used as a post-processing stage after the background is modeled.

Another approach to model variations in the background is to represent these changes as different states, corresponding to different environments, such as lights on/off, night/day, sunny/cloudy. For this purpose Hidden Markov Models (HMM) have been used in Refs. 29 and 36. However, these techniques suffer from slow model training speed and are sensitive to model selection and initialization.

Recently, we investigated two statistical methods for background modeling, based on adaptive kernel density estimation (AKDE),[39] and recursive modeling (RM).[41]

The theory behind the AKDE algorithm is to estimate the probability of each pixel being background based on a number of samples in its history. One advantage of the AKDE method over existing density estimation techniques is that in the proposed algorithm, instead of a single threshold for all pixels in the scene, for each pixel a different threshold is used. By employing these localized thresholds the system works efficiently on different video scenes and is more robust to local changes in the same scene. Also the AKDE method exploits dependency between pixel color components. This results in a more accurate background model.

The RM method is a recursive counterpart for the AKDE technique which uses pixel intensity/color values in new frames to update the background model at that pixel location. Since the update process is performed continuously, the background model converges to the actual one as more frames are processed. This gives the RM method the ability to detect foreground regions in situations where the background changes are very slow and do not fit in a small number of training frames. Also in videos without a set of empty background frames the RM technique has the ability to generate a clear background model. The proposed RM method uses a schedule for learning, which makes the background model converge to the actual one and recover from the expired model faster.

There is a major drawback in all of the statistical modeling methods including the AKDE and the RM techniques. The accuracy of these methods is limited to the accuracy of the estimated probability density function for the background pixels. In this paper we present a non-statistical method that addresses this difficulty.

Furthermore, there is an additional issue with all statistical foreground detection techniques including the AKDE and the RM methods. In all statistical methods the assumption is that there are two classes, namely foreground and background, and the model is trained on background samples which are present during a short period of time (the AKDE) or from the beginning of the video (the RM). Note that until a foreground object appears in the scene, there is no information about the foreground class. This problem is addressed by using thresholds in the classification stage to label pixels as foreground/background.

The contribution of this paper is in its ability to explicitly address the above issues, dependence to probability estimation and the single-class classification problem. We propose a non-statistical background modeling technique based on support

vector data description modeling (SVDDM). The SVDDM uses support vectors to generate a description for the known data; i.e. the background. These support vectors along with the classifier information for each pixel are stored and used in the classification stage to label pixels in new frames as foreground/background. The performance of this system is studied and its experimental results on both synthetic data and real video sequences are compared with other existing techniques in the literature.

The rest of this paper is organized as follows. In Section 2 the theory behind the support vector data description is presented. Section 3 gives a detailed algorithm of the support vector data description method in detecting foreground regions in video sequences. The performance of the proposed method is evaluated in Section 4. Section 5 presents a comparison between the performance of the proposed method and other existing techniques on real videos as well as synthetic data and a comparison summary is drawn in Section 6. Finally, Section 7 concludes the proposed method and gives future direction of research.

2. Support Vector Data Description Modeling (SVDDM)

In this section a novel technique in describing one class of known data samples, called Support Vector Data Description Modeling,[40] is presented. The backbone of the proposed method is a theory based on describing a data set using their support vectors.[44,42] In the following we present the SVDDM theory and the algorithm which detects foreground regions using this theory in detail.

A normal data description is a description which gives a closed boundary around the data. A simple normal data description can be considered as a sphere with center $\mathbf{a}$ and radius $R > 0$, which encloses all of the training samples $\mathbf{x}_i$. The data description is achieved by minimizing the error function:

$$F(R, \mathbf{a}) = R^2 \tag{1}$$

subject to the constraints:

$$\|\mathbf{x}_i - \mathbf{a}\|^2 \leq R^2, \quad \forall i\,. \tag{2}$$

In order to allow for outliers in the training data set, the distance of each training sample $\mathbf{x}_i$ to the center of the sphere $\mathbf{a}$ should not be strictly smaller than R^2. However, large distances should be penalized. Therefore, after introducing slack variables $\epsilon_i \geq 0$ the minimization problem becomes:

$$F\left(R, \mathbf{a}\right) = R^2 + C \sum_i \epsilon_i \tag{3}$$

subject to the new constraints:

$$\|\mathbf{x}_i - \mathbf{a}\|^2 \leq R^2 + \epsilon_i, \quad \forall i \tag{4}$$

where C controls the trade-off between the sphere volume and the description error.

In order to solve the minimization problem in equation (3), the constraints of equation (4) are introduced to the error function using Lagrange multipliers:

$$L(R, \mathbf{a}, \alpha_i, \gamma_i, \epsilon_i) = R^2 + C \sum_i \epsilon_i - \sum_i \alpha_i [R^2 + \epsilon_i - (\|\mathbf{x}_i - \mathbf{a}\|^2)] - \sum_i \gamma_i \epsilon_i \quad (5)$$

where $\alpha_i \geq 0$ and $\gamma_i \geq 0$ are lagrange multipliers. Optimization is achieved by minimizing L.

$$\frac{\partial L}{\partial R} = 0 : \qquad \sum_i \alpha_i = 1 \,, \quad (6)$$

$$\frac{\partial L}{\partial \mathbf{a}} = 0 : \qquad \mathbf{a} = \frac{\sum_i \alpha_i \mathbf{x}_i}{\sum_i \alpha_i} = \sum_i \alpha_i \mathbf{x}_i \,, \quad (7)$$

$$\frac{\partial L}{\partial \gamma_i} = 0 : \qquad C - \alpha_i - \gamma_i = 0 \,. \quad (8)$$

From the above equations and the fact that the Lagrange multipliers are not all negative, when we add the condition $0 \leq \alpha_i \leq C$, Lagrange multipliers γ_i can be safely removed. By replacing results of (6)-(8) into (5) we have:

$$L = \sum_i \alpha_i (\mathbf{x}_i \cdot \mathbf{x}_i) - \sum_{i,j} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) \quad \forall \alpha_i : 0 \leq \alpha_i \leq C \,. \quad (9)$$

A set of α_i values can be achieved using equation (9). If a sample $\mathbf{x}_i$ satisfies the inequality in equation (4) its corresponding Lagrange multiplier will be zero ($\alpha_i = 0$). For all the training samples for which the equality in Eq. (4) is satisfied the Lagrange multipliers become greater than zero ($\alpha_i > 0$). These are the possible scenarios for a given sample $\mathbf{y}_i$:

$$|\mathbf{y}_i - \mathbf{a}|^2 < R^2 \Rightarrow \alpha_i = 0, \gamma_i = C \,, \quad (10)$$

$$|\mathbf{y}_i - \mathbf{a}|^2 = R^2 \Rightarrow 0 < \alpha_i < C, \gamma_i > 0 \,, \quad (11)$$

$$|\mathbf{y}_i - \mathbf{a}|^2 > R^2 \Rightarrow \alpha_i = C, \gamma_i = 0 \,. \quad (12)$$

Note that from equation (7), the center of the sphere is a linear combination of the training samples. Only those training samples $\mathbf{x}_i$ which satisfy (4) by equality are needed to generate the description since their coefficients are not zero. Therefore these samples are called *Support Vectors*.

The above theory describes the normal data description which is the smallest sphere surrounding the training data. However, this simple, normal description is not enough for more complex data which dose not fit into a sphere, i.e. the description needs more complex boundaries.

Here we describe the extension of the normal data description in order to allow more general boundaries. To achieve a more flexible description, instead of a simple dot product of the training samples $(\mathbf{x}_i \cdot \mathbf{x}_j)$, we can perform the dot product using a kernel function:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \,. \quad (13)$$

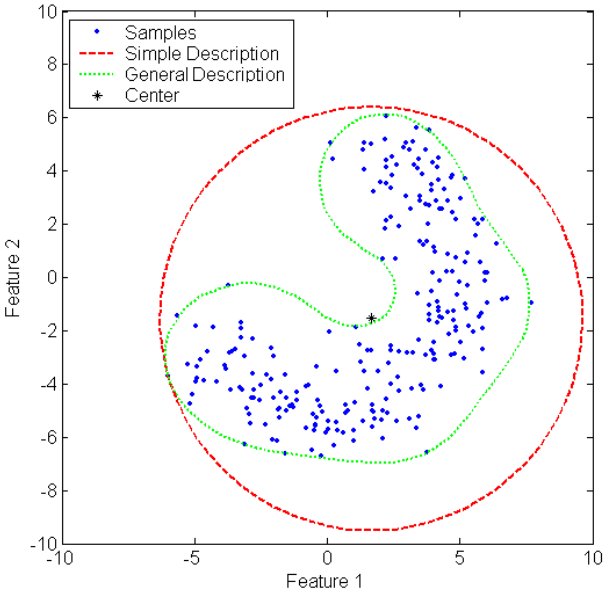


Fig. 2. General vs. simple data descriptions, using the kernel function to extend the description to higher dimensions.

This is done by using a mapping function Φ which maps the data into another (higher dimensional) space. By performing this mapping any complicated boundary (description of data) in low dimension can be modeled by a hyper-sphere in a higher dimension. Several kernel functions have been proposed in the literature,[47] among which the Gaussian kernel gives a closed data description:

$$K(\mathbf{x}_i, \mathbf{x}_j) = exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right). \tag{14}$$

This can be seen from Figure 2, where the blue dots are the samples, the red dashed line is the simple description and the green dotted line is the extended (general) description obtained by mapping the data into the kernel space using equation (14).

According to the above theory the proposed SVDDM method generates a support vector data description for each pixel in the scene using its history. These descriptions are then used to classify each pixel in new frames as a background or a novel/foreground pixel. In the following section the actual implementation of the system is presented.

3. The Algorithm

The methodology described in section 2 is used in our technique to build a descriptive boundary for each pixel in the background training frames in order to generate its model for the background. These boundaries are then used to classify

```
1. Initialization; C : Confidence, N: Number of training frames, σ : bandwidth
     For each pixel x_ij
        1.1. Training stage
             SVD(i,j)← Train(x_ij [1:N]) 2. For each frame at time t
     For each pixel x_ij
        2.1. Classification stage
             DV(i,j)← Test(x_ij [Current Frame],SVD(i,j))
             Label pixel based on DV(i,j).
        2.2. Update stage
             if !(t mod 10)
                 SVD(i,j)← Train(x_ij [t−N:t])
```

Fig. 3. The SVDDM algorithm.

their corresponding pixels in new frames as background and novel (foreground) pixels. There are several advantages in using the Support Vector Data Description (SVDD) method in detecting foreground regions:

- The proposed method, as opposed to existing statistical modeling methods, explicitly addresses the single-class classification problem. Existing statistical approaches try to estimate the probability of a pixel being background, and then use a threshold for the probability to classify it into background or foreground regions. The disadvantage of these approaches is in the fact that it is impossible to have an estimate of the foreground probabilities, since there are no foreground samples in the training frames.
- The proposed method has lower memory requirements compared to non-parametric density estimation techniques, where all the training samples for the background need to be stored in order to estimate the probability of each pixel in new frames. The proposed technique only requires a very small portion of the training samples, the *support vectors*, to classify new pixels.
- The accuracy of our method is not limited to the accuracy of the estimated probability density functions for each pixel. Also the fact that there is no need to assume any parametric form for the underlying probability density of pixels gives the proposed method superiority over the parametric density estimation techniques, i.e. mixture of Gaussians.
- The efficiency of our method can be explicitly measured in terms of false reject rates. The proposed method considers a goal for false positive rates, and generates the description of data by fixing the false positive tolerance of the system. This helps in building a robust and accurate background model.

Figure 3 shows the proposed algorithm in pseudo-code format.[a] The only critical parameter is the number of training frames (N) that needs to be initialized. The support vector data description confidence parameter C is the target false reject rate of the system, which accounts for the system tolerance. Finally the Gaussian

[a]The proposed method is implemented in MATLAB 6.5, using Data Description toolbox.[45]

kernel bandwidth, σ does not have a particular effect on the detection rate as long as it is not set to be less than one, since features used in our method are normalized pixel chrominance values. For all of our experiments we set $C = 0.1$ and $\sigma = 5$. The optimal value for these parameters can be estimated by a cross-validation stage.

3.1. *Training stage*

In order to generate the background model for each pixel the SVDDM method uses a number of training frames. The background model in this technique is the description of the data samples (color and or intensity of pixels). The background training buffer is a First In First Out (FIFO) buffer with Round Robin replacement policy. Since the buffer uses Round Robin replacement policy, once the buffer is full the new frames replace the oldest ones in the buffer.

The data description is generated in the training stage of the algorithm. In this stage for each pixel a SVDD classifier is trained using the training frames, detecting support vectors and the values of Lagrange multipliers that maximize equation (9).

The support vectors and their corresponding Lagrange multipliers are stored as the classifier information for each pixel. This information is used for the classification step of the algorithm. The training stage can be performed off-line in cases where there are no global changes in the illumination or can be performed in parallel to the classification to achieve efficient foreground detection.

In the current implementation the training stage is computationally extensive since a quadratic programming optimization is performed to compute Lagrange multipliers. However, an on-line training scheme is under investigation to train the system based on the trained values from previous frames.

3.2. *Classification stage*

In this stage, for each frame its pixels are evaluated by their corresponding classifier to label them as background or foreground. To test each pixel $\mathbf{z}_t$ the distance to the center of the description hyper-sphere is calculated:

$$\|\mathbf{z}_t - \mathbf{a}\|^2 = (\mathbf{z}_t \cdot \mathbf{z}_t) - 2\sum_i \alpha_i (\mathbf{z}_t \cdot \mathbf{x}_i) + \sum_{i,j} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j)\,. \tag{15}$$

A pixel is classified as a background pixel if its distance to the center of the hyper-sphere is smaller or equal to R^2:

$$\|\mathbf{z}_t - \mathbf{a}\|^2 \leq R^2\,. \tag{16}$$

R^2 is the distance from the center of the hyper-sphere to its boundary which is also equivalent to the distance of each support vector to the center of the hyper-sphere. All support vectors which fall outside the boundary are disregarded:

$$R^2 = (\mathbf{x}_k \cdot \mathbf{x}_k) - 2\sum_i \alpha_i (\mathbf{x}_i \cdot \mathbf{x}_k) + \sum_{i,j} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j)\,. \tag{17}$$

Note that in the implementation of the algorithm, since the boundaries of the data description are more complicated than a hyper-sphere, a kernel is used to map the training samples into a higher dimension according to equations (13) and (14). As the result the mapped samples in the higher dimension can be described by a high dimensional hyper-sphere and the above discussion can be used.

3.3. *Update stage*

In order to update the classifiers for each pixel to adapt the system to changes in the illumination a similar approach to AKDE[39] is devised for the SVDDM method. We assume that the scene illumination may undergo *gradual changes* all the time. To adapt the system to gradual changes, every 16 seconds (corresponding to 480 frames of the video) each pixel in the scene undergoes a re-training process.

This retraining process is performed in a similar fashion to the original training, but using as samples only the current support vectors and the current pixel value. To make the system adaptive to *gradual changes* in illumination, we replace pixels in the oldest background frame with those pixels belonging to the current background mask.

In order to detect *sudden global changes* in the illumination, the area of the detected foreground objects is checked. If the foreground mask area is greater than 50% of the frame size a sudden global change is detected. Upon the detection of sudden global illumination change the classification stage of the algorithm is suspended and new frames replace all those in the background training buffer.

However, because of the high computational cost of the training stage, this adaptation process is not performed at each frame.

In the current implementation the time needed for retraining is about 0.8 seconds for the whole frame. Before the results of retraining are ready the system uses the previous support vector description to detect foreground regions. When the retraining is completed its results are used in the detection stage. The detection stage is very fast since it only uses a few support vectors compared to the number of initial training sample points. The classification stage detects foreground regions at a rate of 10 frames per second.

Currently an online training scheme is under investigation to improve the retraining speed of the system and make it suitable for real-time applications. In the current implementation, the retraining process is performed in parallel to the classification stage and as soon as the new classifier is trained for each pixel it is used instead of the older classifier.

4. Performance Evaluation

In this section the SVDDM performance is evaluated in terms of memory requirements and computation cost. The key to evaluate the performance of this technique is to analyze the optimization problem solved by the system to find support vectors. This issue is discussed in detail in the following.

Table 1. Per-pixel memory requirements for the SVDDM method.

Memory Req.	Intensity	Chrominance	Both	Asymp.
Bytes/pixel	$[5 \times f(C,\sigma)] \geq 10$	$[8 \times f(C,\sigma)] \geq 24$	$[f(C,\sigma)] \geq 32$	$O(1)$
No. of SVs	$f(C,\sigma) \geq 2$	$f(C,\sigma) \geq 3$	$f(C,\sigma) \geq 4$	$O(1)$

f is a function of the description target accuracy.

- **Parameters.** As described earlier in this chapter the support vector data description (SVDD) is an elegant technique to describe one class of data samples without any information about data belonging to other (novel) classes. In order to generate the data description, a hyper-sphere of minimum size which contains most of the training samples is constructed, which represents the boundary of the known class. There are parameters involved with the construction of the class boundary (i.e. hyper-sphere), namely, the number of training samples N and trade off factor C in equation (3) and the bandwidth of the Gaussian kernel σ in (14). As mentioned in section 3 for all experiments the values for C and σ are 0.1 and 5, respectively. This leaves the system with only the number of frames as a scene-dependent parameter.
- **Memory requirements.** It is not easy to answer how many data samples are required to find a sufficiently accurate description of a target class boundary. It not only depends on the complexity of the data itself but also on the distribution of the outlier (unknown) class. However, there is a trade-off between the number of support vectors to describe the data set and the driven description accuracy. In that sense a lower limit can be found for the number of samples that can describe the data which corresponds to the rigid hyper-sphere containing most of the data samples with the target error goal.

  In theory only $d + 1$ support vectors in d dimensions are sufficient to construct a hyper-sphere and their corresponding Lagrange multipliers (α_i) sum to 1. The center of the sphere lies within the convex hull of these support vectors. This translates to the function f in Table 1. The minimum number of support vectors needed is a function of the accuracy of description that we require. The least accurate description which is a hyper-sphere requires only $d + 1$ support vectors. To preserve more accuracy one needs to retain more support vectors which results in more memory requirements and less generalization. So there is a trade off between memory requirements of the system and its generalization and accuracy.

  — *Using only intensity values.* Since by using intensity for each pixel there is only one feature value the support vectors are 1-D and therefore the minimum number of support vectors required to describe the data will be 2. For each pixel 2 bytes are required to store the intensity and 8 bytes to store the Lagrange multipliers, resulting in at least 10 bytes per pixel memory requirements.

Table 2. Per-pixel memory requirements for the AKDE, the RM and the SVDDM.

Memory Req.	Intensity	Chrominance	Both	Asymptotic
The AKDE[39]	$N + 8$	$8N + 20$	$9N + 40$	$O(N)$
The RM[38]	1024	2048	3072	$O(1)$
The SVDDM	$[f(C, \sigma) \times 5] \geq 10$	$[f(C, \sigma) \times 8] \geq 24$	$f(C, \sigma) \geq 32$	$O(1)$

— *Using chrominance values.* By using red and green chrominance values, c_r and c_g, a minimum of 3 support vectors need to be used. This results in at least 24 bytes per pixel memory requirements.

The above argument is about the lower limit of the number of support vectors. Obviously in practical applications this lower limit is far from being useful for implementation. The only fact that can be used from the above discussion is that the number of support vectors required to sufficiently describe a data set is related to the target accuracy of the description. Therefore, the memory requirements of the SVDDM method are independent of the number of training frames.

Table 1 shows memory requirements in bytes per pixel for the SVDDM method using intensity, chrominance values and their combinations respectively. In conclusion the asymptotic memory requirements of the SVDDM algorithm are constant $O(1)$ since they are independent of the number of training frames.

- **Computation cost.** Training the SVDDM system for each pixel needs to maximize (9) which is a quadratic programming (QP) optimization problem. The most common technique to solve the above QP is Platt's algorithm (sequential minimal optimization),[26,25] which runs in polynomial time.

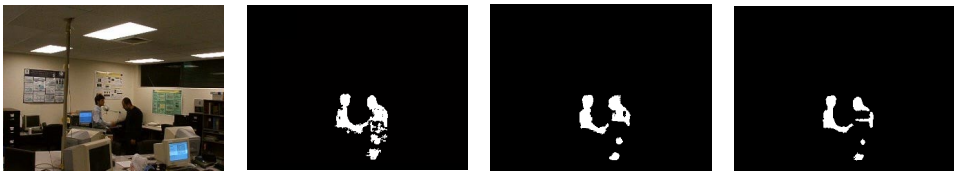
5. Experimental Results and Comparison

In this section we evaluate the performance of proposed technique using several real video sequences that pose significant challenges. The performance is also compared to the mixture of Gaussians method,[35] the spatio-temporal modeling presented in Ref. 19, the RM in Ref. 38, the AKDE method in Ref. 39 and the simple KDE method.[7] We use different scenarios to test the performance of the proposed techniques and discuss where each method appears to be particularly suitable.

In some experiments the results of the proposed method are compared with results of the AKDE method[39] and/or the RM method.[38,41] As mentioned in Refs. 38–41, these methods outperform other existing algorithms.

In the real video experiments conducted below, $N = 300$ frames are used in the background training stage unless otherwise stated, corresponding to 10 seconds of the videos sequence. The background training buffer is a First In First Out (FIFO) buffer with Round Robin replacement policy.

From Table 2 we notice that the memory requirements of the AKDE technique presented in Ref. 39 and the SVDDM are lower than those of the RM method[38] if the number of training frames is small enough. That is, for situations when a small



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

Fig. 4. Rapidly fluctuating background: (a) *Handshake* video sequence. Detected foreground regions using (b) AKDE; (c) RM; (d) SVDDM.

number of frames can cover most changes that occur in the background, by using a sliding window of limited size the AKDE method needs less memory than the RM technique. Note that SVDDM needs even less memory than the AKDE and like the RM its memory requirements are independent of the *number of training samples*.

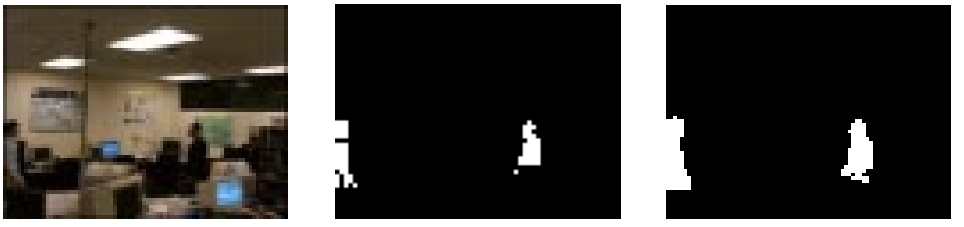
5.1. *Rapidly fluctuating backgrounds*

As described above, for videos with rapidly changing backgrounds, the AKDE method has a better performance in terms of memory requirements and speed. Our experiments showed that for videos where possible fluctuations in the background occur in about 10 seconds, the AKDE technique needs less memory and works faster compared to the RM and SVDDM methods.

Figure 4 shows the detection results of the AKDE, RM and the SVDDM algorithms on the *Handshake* video sequence where the pixel values corresponding to monitors fluctuate rapidly. As it can be seen from this figure, capturing dependencies between chrominance features results in more accurate foreground regions (Fig. 4(b)), showing that AKDE performs better than both the RM and the SVDDM. Note that in this particular frame the color of foreground objects is very close to the background in some regions. The SVDDM technique results in very smooth and reliable foreground regions because this technique uses the confidence factor C to eliminate false positives in the classification. This may lead to missing some parts of the foreground which are very close in color to the background. Also notice that in all methods fluctuations in monitors are completely modeled as a part of background and not detected as foreground regions.

5.2. *Low contrast videos*

To evaluate the accuracy of the SVDDM technique in low contrast video sequences and compare its results with those of the AKDE technique the experiment is performed again on the *Handshake* video sequence but we intentionally decreased the quality of the images by down-sampling them by a factor of 4. Figure 5 shows a frame where the background and foreground colors are reasonably different. The accuracy of the foreground region detected using the SVDDM technique is clearly better than that of the AKDE method. The reason is that the SVDDM method



(a) (b) (c)

Fig. 5. Low contrast videos: (a) *Handshake* video sequence. Detected foreground regions using (b) AKDE; (c) SVDDM.

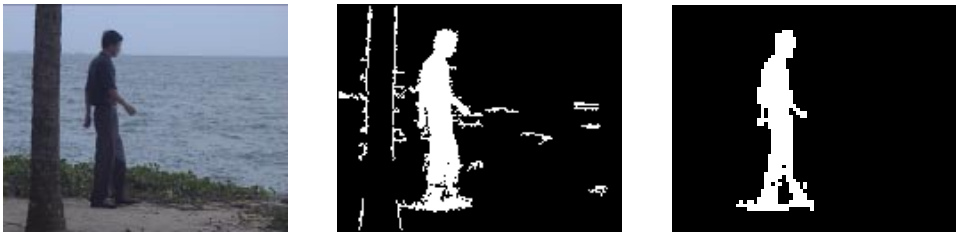
fixes the false reject rate of the classifier and generates a discrimination boundary independent of the estimated probability density function of the single known class; i.e., the background class. On the other hand, the AKDE method attempts to use a statistical binary classification for a single-class classifier introducing statistical Bayes error to the model.

5.3. *Slowly changing backgrounds*

For videos with slowly changing backgrounds or backgrounds whose changes are not periodic, the AKDE method needs more training frames to generate a good model for the background. This increases the system memory requirements and drastically decreases its training and foreground detection speed. In this example in order to get a reasonably accurate foreground, a large number of frames is required. Since asymptotic training time, detection time and memory requirements of the AKDE are $O(N)$, it is not an efficient method for this scenario. In order to achieve an efficient foreground detection in the AKDE we used $N = 300$ frames as background training frames.

In these situations the SVDDM technique is a very good alternative, since its detection speed and memory requirements are independent of the number of training frames. Note that in order to cover the possible changes in slowly changing backgrounds, both SVDDM and AKDE require a large number of training frames. However, in SVDD once the background model is trained the system only retains support vectors. As discussed earlier in section 4 the number of support vectors needed to detect foreground regions is asymptotically constant and independent of the number of training frames. The proposed retraining scheme makes the SVDDM adaptation more efficient. As seen in Fig. 6 the SVDDM results are better than the AKDE.

Figure 6(a) shows an arbitrary frame of the *Water* video sequence. In this figure the detection results of the AKDE and the SVDDM methods are presented. This example is particularly difficult because waves do not follow a regular motion pattern and their motion is slow. As it can be seen from Fig. 6(b), using the AKDE method without any post-processing results in many false positives where



(a) (b) (c)

Fig. 6. Slowly changing background: (a) *Water* video sequence. Detected foreground region using (b)AKDE; (c) SDDM.

the SVDDM (Fig. 6(b)) learns the background more accurately using more training frames.

From this figure we can conclude that the SVDDM method has a better performance compared to the AKDE in situations where the background has a slow and irregular motion. Also in the AKDE there is a sliding window of limited size which may not cover all changes in the background resulting in an inaccurate probability density estimation but the model built in the SVDDM uses the decision boundaries of the single training class instead of bounding the training accuracy to the accuracy of the probability estimation.

5.4. *Other difficult examples*

Figure 7 shows three more video sequences with challenging backgrounds. In column (a) the original frames are shown, while column (b) and (c) show the results of the AKDE and the SVDDM methods, respectively. In this figure, from top row to the bottom, heavy rain, waving tree branches and the water fountain pose significant difficulties in detecting accurate foreground regions.

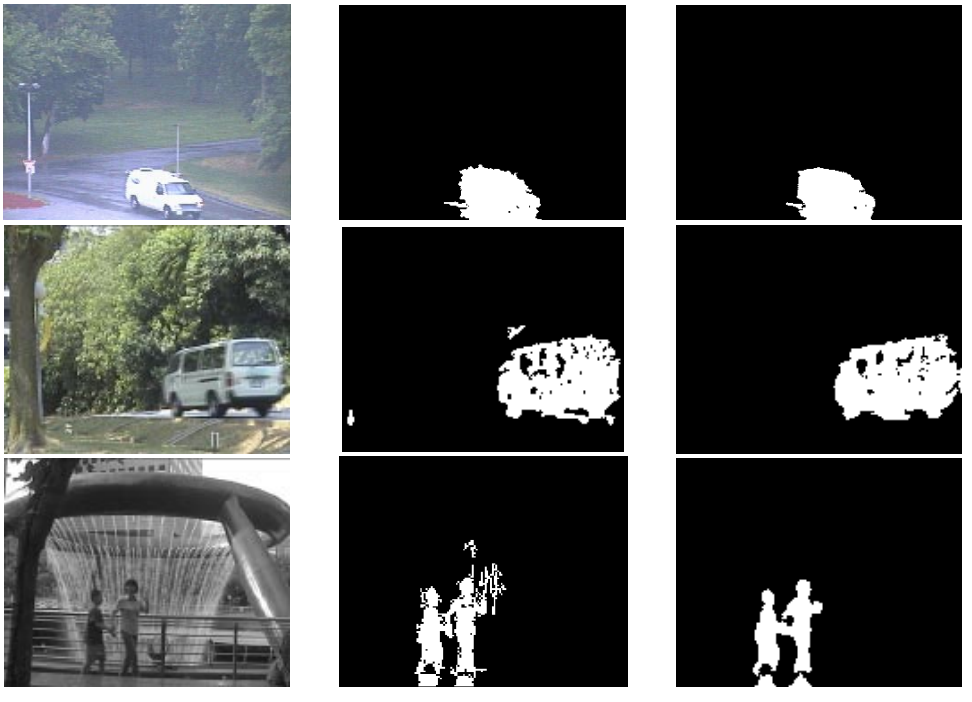
5.5. *Quantitative evaluation*

Performance of our proposed method is evaluated quantitatively on randomly selected samples from different video sequences, taken from Ref. 19. Figure 8 shows the ground truth masks for some challenging video sequences.

To evaluate the performance of each method we use the similarity measure between two regions $\mathcal{A}$ (detected foreground regions) and $\mathcal{B}$ (ground truth), defined by:

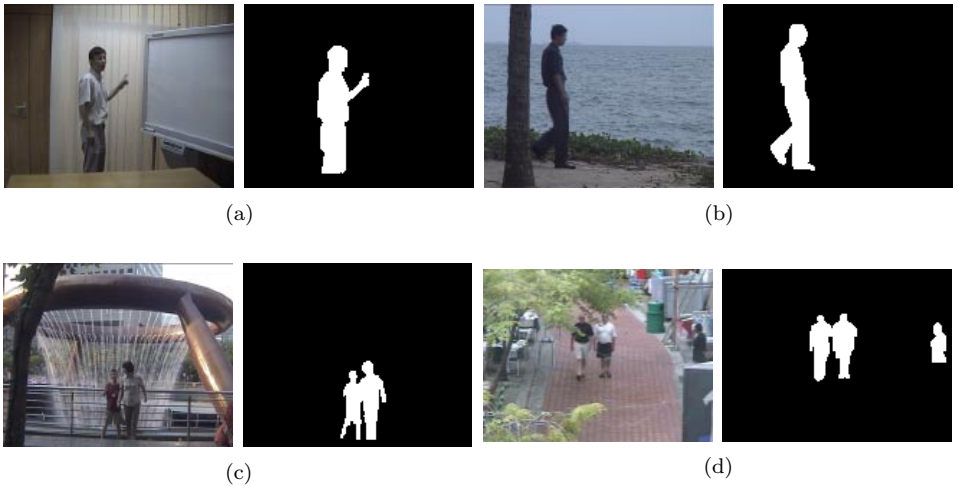
$$\mathcal{S}(\mathcal{A}, \mathcal{B}) = \frac{\mathcal{A} \cap \mathcal{B}}{\mathcal{A} \cup \mathcal{B}}. \tag{18}$$

This measure increases monotonically with the similarity between detected masks and the ground truth, ranging between 0 and 1. By using this measure we report the performance of the SVDDM method, the AKDE,[39] the spatio-temporal technique presented in Ref. 19 and the mixture of Gaussians (MoG) in Ref. 35.



(a) (b) (c)

Fig. 7. Other difficult examples: (a) Original frame. Detected foreground region using (b) AKDE; (c) SVDDM.



(a) (b)

(c) (d)

Fig. 8. Ground truth masks for some challenging video sequences: (a) Meeting Room. (b) Water Surface. (c) Fountain. (d) Side Walk.

Table 3. Quantitative evaluation and comparison. The sequences are *Meeting Room, Lobby, Campus, Side Walk, Water* and *Fountain*, from left to right.

Method \ Video	MR	LB	CAM	SW	WAT	FT	Avg. $\mathcal{S}(\mathcal{A}, \mathcal{B})$
SVDDM	0.84	0.78	0.70	0.65	0.87	0.80	0.77
AKDE[39]	0.74	0.66	0.55	0.52	0.84	0.51	0.64
Spatio-Temp[19]	0.91	0.71	0.69	0.57	0.85	0.67	0.74
MoG[35]	0.44	0.42	0.48	0.36	0.54	0.66	0.49

Table 4. Computation time.

Videos	Init. Train	Retraining†	Segmentation‡	Avg. Time*	Speed**
Handshake	24	0.8	0.085	279	12
Fountain	25.6	0.85	0.1	326	11

†: Mean time for 1 frame, every 480 frames (sec)
‡: Mean time for 1 frame, every frame (sec)
∗: Mean process time over 3000 frames (sec)
∗∗: Speed (fps)

By comparing the average of the similarity measure over different video sequences in Table 3, we can see that the SVDDM method outperforms other techniques. This also shows that the SVDDM method works consistently well on a wide range of video sequences. Also note that unlike the existing techniques the SVDDM and AKDE methods are automatic, without the need for fine-tuning a large number of parameters for each scene.

5.6. *Computation time*

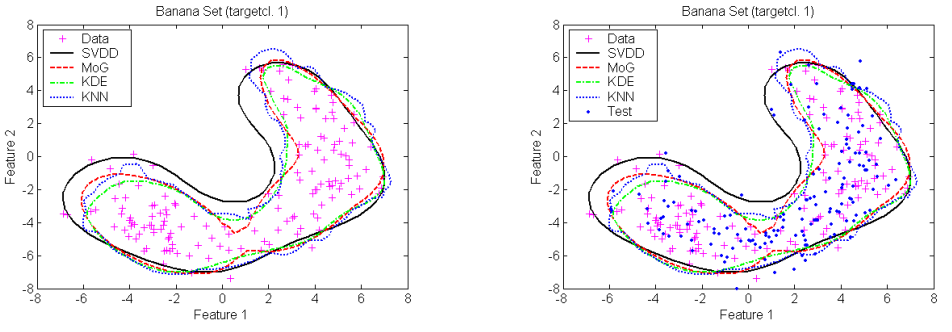
In this section we discuss the speed of our algorithm on the *Handshake* and *Fountain* video sequences. The frame size is 120×160 and it is in *RGB* color format. The system is implemented on an Intel Dual Processor Pentium 4 with 4.8 GHz cpu clock. We used $N = 300$ frames for the initial background training process. The retraining is performed every 16 seconds.

In the current implementation, the retraining stage is performed on the whole frame in parallel with detection process. Once ready, the results of the retraining process are used in the foreground detection stage.

Table 4 shows the computation time of the system. The initial training is performed once at the beginning of the process as well as after the detection of a sudden global change. The rest of the adaptation process uses the retraining stage.

5.7. *Synthetic data sets*

In this section we use a synthetic data set, which represents randomly distributed training samples with an unknown distribution function (*banana* data set). Figure 9



(a) (b)

Fig. 9. Comparison between different classifiers on a synthetic data set: (a) Decision boundaries of different classifiers after training. (b) Data points (blue dots) outside decision boundaries are false rejects.

shows a comparison between different classifiers. This experiment is performed on 150 training samples using support vector data description (SVDDM), mixture of Gaussians (MoG), kernel density estimation (AKDE) and k-nearest neighbors (KNN).

Parameters of these classifiers are manually determined to give a good performance. For all classifiers the confidence parameter is set to be 0.1. In MoG, we used 3 Gaussians. The Gaussian kernel bandwidth in the AKDE classifier is considered $\sigma = 1$, for the KNN we used 5 nearest neighbors, and for the SVDDM classifier the Gaussian kernel bandwidth is chosen to be 5.

Figure 9(a) shows the decision boundaries of different classifiers on 150 training samples from *banana* dataset. From Fig. 9(b), SVDDM generalizes better than the other three classifiers and classifies the test data more accurately. In this figure the test data is composed of 150 samples drawn from the same probability distribution function as the training data, which should be classified as the known class.

Here we need to define the False Reject Rate (FRR) and Recall Rate (RR) for a quantitative evaluation. By definition, FRR is the percentage of missed targets, and RR is the percentage of correct predictions (True Positive rate). These quantities are given by:

$$\text{FRR} = \frac{\#\text{Missed targets}}{\#\text{Samples}}, \qquad \text{RR} = \frac{\#\text{Correct predictions}}{\#\text{Samples}}. \tag{19}$$

Table 5 shows a quantitative comparison between different classifiers. In this table, FRR and RR of classifiers are compared after training them on 150 data points drawn from an arbitrary probability function and tested on the same number of samples drawn from the same distribution. As it can be seen from the above example, the FRR for SVDDM is less than that of the other three, while its RR is higher. This proves the superiority of this classifier in the case of single class classification over the other three techniques.

Table 5. Comparison of False Reject Rate and Recall Rate for different classifiers.

Method	SVDDM	MoG	AKDE	KNN
FRR	0.1067	0.1400	0.1667	0.1333
RR	0.8933	0.8600	0.8333	0.8667

Table 6. Need for manual optimization and number of parameters.

Method	SVDDM	MoG	AKDE	KNN	RM
No. of parameters	1	4	2	2	2
Need manual selection	No	Yes	Yes	Yes	Yes

Table 7. Comparison of memory requirements for different classifiers.

Method	SVDDM	MoG	AKDE	KNN	RM
Memory needs (bytes)	1064	384	4824	4840	1024

Table 6 compares the number of parameters of each classifier and the need for manually selecting these parameters for a single class classification problem. As it can be seen, the only method that automatically determines data description is the SVDDM technique. In all other classification techniques there is at least one parameter that needs to be manually chosen to give a good performance. Note that this table refers to the parameters of the classifiers not the technique used for background modeling. In methods such as AKDE and MoG there has been efforts to learn some of these parameters and make the system automatic in detecting foreground regions.

Table 7 shows memory requirements for each classifier. As it can be seen from the table, SVDDM requires much less memory than the KNN and AKDE methods, since in SVDDM we do not need to store all the training data. Only the MoG and the RM methods need less memory than the SVDDM technique, but in MoG methods we need to manually determine the number of Gaussians to be used which is not practical when we are training one classifier per pixel in real video sequences.

6. Comparison Summary

Table 8 summarizes this study and provides a comparison between different traditional methods for background modeling and our proposed method. The comparison includes the classification type, memory requirements, computation cost and type of parameter selection.

As seen in Table 8, the RM method uses a MAP decision criterion where other systems except the SVDDM use a Bayes classifier. The only method which explicitly deals with the single class classification is the SVDDM technique which fits the description of data belonging to the background class in its rather simple training

Table 8. Comparison between the proposed methods and traditional techniques.

	SVDDM	AKDE[39]	RM[38]	KDE[7]	Spatio-temp[19]	MoG[35]	Wallflower[46]
Automated	Yes	Yes	Yes	No	No	No	No
Post proc.	No	No	No	No	Yes	No	No
Classifier	SVD	Bayes	MAP	Bayes	Bayes	Bayes	K-means
Memory req.*	$O(1)$	$O(N)$	$O(1)$	$O(N)$	$O(N)$	$O(1)$	$O(N)$
Comp. cost*	$O(N)$	$O(N)$	$O(1)$	$O(N)$	$O(N)$	$O(1)$	$O(N)$

$*$: Per-pixel N: number of training frames

Table 9. Scenarios where each method appears to be particularly suitable.

	SVDDM	AKDE[39]	RM[38]	KDE[7]	Spat-tmp[19]	MoG[35]	Wallflower[46]
Low contrast	S	S	NS	S	NS	NS	NS
Close Bg/Fg color	NS	S	NS	NS	NS	NS	NS
Slow changing Bg	S	NS	S	NS	S	S	S
Fast changing Bg	S	S	S	S	S	NS	S
Sudden changes	S	NS	S	NS	S	S	NS
Non-empty Bg	NS	NS	S	NS	S	S	S
Hand-held	NS	NS	S	NS	NS	NS	NS

S: Suitable NS: Not suitable

stage. Other methods shown in the table use a binary classification scheme and use heuristics (Refs. 7, 35 and 46) or a more complex training scheme (Refs. 39 and 38) to make it useful for the single-class classification problem of background modeling.

Table 9 shows different scenarios and illustrates where each method appears to be particularly suitable for foreground region detection. As expected the SVDDM method is suitable for a wide range of applications. In case of low contrast videos the SVDDM works reasonably well in detecting foreground regions. This is due to the fact that this technique aims to build and accurate background model from a finite number of background training frames. However the accuracy of the background model is limited to the false reject rate of the classifier in the SVDDM technique. This makes the SVDDM method unable to detect foreground regions in those locations where the background color is very close to the foreground color but being more robust to noise. The AKDE is more sensitive to noise compared to the SVDDM but on the other hand works better in these situations.

As discussed earlier and also seen from Table 9, the only method suitable for scenarios where there is a steady and very slow motion in the background; such as the hand-held camera scenario, is the RM technique. The other methods fail to build a very long term model for the background model because of the fact that their cost grows linearly by the number of training background frames, as it can be seen from Table 8. Also in scenarios where there is no empty set of background, called non-empty backgrounds, only RM method is suitable and works independently without any need to perform post processing steps.

7. Conclusions and Future Work

As discussed in the previous sections, the detection of foreground regions is a crucial step in many high-level video processing applications such as video indexing, video compression, video surveillance and activity recognition. In some applications it is assumed that the camera is fixed and its parameters do not change. In this context the problem of detecting foreground regions is a challenging issue because the background may not be completely stationary due to fluctuations in the background, changes in the illumination and small camera shakes.

In this paper a novel approach is proposed to label pixels in video sequences into foreground and background classes using a support vector data description. The contributions of our method can be described along the following directions:

- The model accuracy is not bounded to the accuracy of the estimated probability density functions.
- The memory requirements of the proposed technique are lower than those of non-parametric techniques and are independent of the number of training samples.
- Because the support vector data description explicitly models the decision boundary of the known class, it is suitable for novelty detection without the need to use thresholds. This results in less parameter tuning, leading to a scene-independent algorithm.
- The classifier performance in terms of false positives is controlled explicitly.

Directions of future study related to this work include investigating an on-line training scheme for the SVDDM technique to make it more efficient in terms of speed and computational cost. One can also use tracking information combined with pixel features to achieve more accurate detection results. Incorporating detection/tracking results into useful information for high-level stages of the video processing application such as activity recognition is another important issue that needs to be formalized. As an interesting extension to this work, we are also working on a reliable foreground model using the detection results to relax the stationary camera assumption.

Acknowledgment

This work was supported by the NSF-EPSCoR Ring True III award EPS0447416 and by the Office of Naval Research award N00014-06-1-0611.

References

1. C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Walton Street, Oxford OX26DP, 1994.
2. C. Bishop. Novelty detection and neural network validation. *In IEE proceedings on Vision, Image and Signal Processing. Special Issue on Application of Neural Networks*, 141(4):217–222, 1994.
3. E. Durucan and T. Ebrahimi. Change detection and background extraction by linear algebra. *In proceedings of IEEE*, 89(10):1368–1381, Oct. 2001.

4. A. Elgammal and L. S. Davis. Probabilistic framework for segmenting people under occlusion. *In proceedings of the 8th IEEE International Conference on Computer Vision*, 2001.
5. A. Elgammal, R. Duraiswami, and L. S. Davis. Efficient computation of kernel density estimation using fast gauss transform with application for segmentation and tracking. *In proceedings of the 2nd IEEE International Workshop on Statistical and Computational Theories of Vision*, 2001.
6. A. Elgammal, R. Duraiswami, and L. S. Davis. Efficient non-parametric adaptive color modeling using fast gauss transform. *In proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2001.
7. A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *In proceedings of the IEEE*, 90:1151–1163., 2002.
8. N. Friedman and S. Russell. Image segmentation in video sequences: A probabilistic approach. *Annual Conference on Uncertainty in Artificial Intelligence*, pages 175–181, 1997.
9. I. Haritoglu, D. Harwood, and L. S. Davis. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:809–830, 2000.
10. Y. Hsu, H. Nagel, and G. Rekers. New likelihood test methods for change detection in image sequences. *Computer Vision, Graphics and Image Processing*, 26:73–106, July 1984.
11. N. Japlpwicz, C. Meyers, and M. Gluck. A novelty detection approach to classification. *In proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 518–523, 1995.
12. O. Javed, K. Shafique, and M. Shah. A hierarchical approach to robust background subtraction using color and gradient information. *In proceedings of IEEE Workshop on Motion Video Computing*, pages 22–27, Dec. 2002.
13. K. P. Karman and A. von Brandt. Moving object recognition using an adaptive background memory. *Time-varying Image Processing and Moving Object Recognition*, 1990.
14. K. P. Karman and A. von Brandt. Moving object segmentation based on adaptive reference images. *Signal Processing: Theories and Applications*, 1990.
15. K. Kim, D. Harwood, and L. S. Davis. Background updating for visual surveillance. *In proceedings of the International Symposium on Visual Computing*, 1:337–346, December 2005.
16. D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russel. Towards robust automatic traffic scene analysis in real-time. *In proceedings of ICPR*, 1:126–131, October 1994.
17. C. Lambert, S. Harrington, C. Harvey, and A. Glodjo. Efficient on-line non-parametric kernel density estimation. *Algorithmica*, (25):37–57, 1999.
18. D. Lee. Effective gaussian mixture learning for video background subtraction. *IEEE Transactions on PAMI*, 27(5):827–832, May 2005.
19. L. Li, W. Huang, I.Y. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing.*, 13(11):1459–1472, November 2004.
20. T. Matsuyama, T. Ohya, and H. Habe. Background subtraction for non-stationary scenes. *In proceedings of Asian Conference on Computer Vision*, pages 662–667, July 2000.
21. S.J. McKenna, Y. Raja, and S. Gong. Object tracking using adaptive color mixture

models. *In proceedings of Asian Conference on Computer Vision*, 1:615–622, January 1998.
22. S.J. McKenna, Y. Raja, and S. Gong. Tracking color objects using adaptive mixture models. *Image and Vision Computing*, 17:223–229, 1999.
23. A. Mittal and N. Paragios. Motion-based background subtraction using adaptive kernel density estimation. *In proceedings of CVPR*, 2:302–309, July 2004.
24. N. Paragios and V. Ramesh. A mrf-based approach for real-time subway monitoring. *IEEE Transactions on PAMI*, 1:1030–1040, December 2001.
25. J. Platt. *Advances in Kernel Methods - Support Vector Learning.* Editors: B. Schlkopf, C. Burges, A. J. Smola, MIT Press, 1998.
26. J. Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. *Microsoft Research Technical Report MSR-TR-98-14*, 1998.
27. R. Pless, T. Brodsky, and Y. Aloimonos. Detecting independent motion: The statistics of temporal continuity. *IEEE Transactions on PAMI*, 22(8):68–73, 2000.
28. R. Pless, J. Larson, S. Siebers, and B. Westover. Evaluation of local models of dynamic backgrounds. *In proceedings of the CVPR*, 2:73–78, June 2003.
29. J. Rittscher, J. Kato, S. Joga, and A. Blake. A probabilistic background model for tracking. *In proceedings of 6th European Conference on Computer Vision*, 1843:336–350, 2000.
30. D. W. Scott. *Multivariate Density Estimation.* Wiley Interscience, 1992.
31. Y. Sheikh and M. Shah. Bayesian object detection in dynamic scenes. *In proceedings of the CVPR*, 1:74–79, June 2005.
32. B. Sholkopf. *Support Vector Learning.* Ph.D. Thesis, Technischen Universitat Berlin, 1997.
33. B. Sholkopf, C. Burges, and V. Vapnik. Extracting support data for a given task. *In proceedings of the 1st International Joint Conference on Knowledge Discovery and Data Mining*, pages 252–257, 1995.
34. C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. *In proceedings of CVPR*, 2:246–252, 1999.
35. C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on PAMI*, 22(8):747–757, August 2000.
36. B. Stenger, V. Ramesh, N. Paragios, F.Coetzee, and J.M. Buhmann. A probabilistic background model for tracking. *In proceedings of ICCV*, pages 294–301, July 2001.
37. A. Tavakkoli, M. Nicolescu, and G. Bebis. Automatic robust background modeling using multivariate non-parametric kernel density estimation for visual surveillance. *In proceedings of the International Symposium on Visual Computing*, LNSC 3804:363–370, December 2005.
38. A. Tavakkoli, M. Nicolescu, and G. Bebis. An adaptive recursive learning technique for robust foreground object detection. *In proceedings of the International Workshop on Statistical Methods in Multi-image and Video Processing (in conjunction with ECCV06)*, May 2006.
39. A. Tavakkoli, M. Nicolescu, and G. Bebis. Automatic statistical object detection for visual surveillance. *In proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation*, pages 144–148, March 2006.
40. A. Tavakkoli, M. Nicolescu, and G. Bebis. A novelty detection approach for foreground region detection in videos with quasi-stationary backgrounds. *In proceedings of the 2nd International Symposium on Visual Computing*, November 2006.
41. A. Tavakkoli, M. Nicolescu, and G. Bebis. Robust recursive learning for foreground region detection in videos with quasi-stationary backgrounds. *In proceedings of 18th International Conference on Pattern Recognition*, August 2006.

42. D. Tax and R. Duin. Support vector domain description. *Pattern Recognition Letters*, (20):1191–1199, 1999.
43. D. Tax and R. Duin. Data description in subspaces. *In proceedings of the International Conference on Pattern Recognition*, 2:672–675, 2000.
44. D. Tax and R. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.
45. D.M.J. Tax. Ddtools, the data description toolbox for matlab, Augustus 2005. version 1.11.
46. K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: principles and practice of background maintenance. *In proceedings of ICCV*, 1:255–261, September 1999.
47. V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.
48. C. Wern, A. Azarbayejani, T. Darrel, and A.P. Pentland. Pfinder: real-time tracking of human body. *IEEE Transactions on PAMI*, 19(7):780–785, July 1997.
49. L. Wixson. Detecting salient motion by accumulating directionary-consistent flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:774–780, August 2000.