Efficient Background Modeling through Incremental Support Vector Data Description

Alireza Tavakkoli and Mircea Nicolescu and George Bebis
Computer Vision Laboratory
University of Nevada, Reno
Email: {tavakkol,mircea,bebis}@cse.unr.edu

Monica Nicolescu
Robotics Laboratory
University of Nevada, Reno
Email: monica@cse.unr.edu

Abstract

Many high-level video processing applications such as visual surveillance require the detection and tracking of objects of interest in the video. However, due to inherent changes such as waving trees, water surfaces, flickering lights, etc., the background may not be completely static even with a fixed camera. Therefore, background modeling becomes an essential and important part of such applications. Recently, the Support Vector Data Description (SVDD) has been introduced to address the issue of single-class classification when samples of only one class of data are available, i.e. background pixels. This paper proposes a method to efficiently train an SVDD and compares the performance of this training algorithm with the traditional SVDD training techniques. We extensively compare the results and performance of our proposed method with traditional SVDD and other classification algorithms on various data sets including real video sequences.

1 Introduction

Background modeling is one of the most effective and widely used techniques to detect moving objects in videos with a quasi-stationary background. In these scenarios, although the camera is considered to be fixed, the background is not completely stationary due to inherent changes, such as water fountains, waving flags, etc. In order to detect moving objects in such scenes the background of the video needs to be modeled. There are several statistical modeling approaches proposed in the literature. These approaches can be used to estimate the probability density function from which the data points are generated.

Parametric density estimation methods, such as Mixture of Gaussians techniques (MoG) are proposed in [4]. However, the parametric density estimation techniques may not be useful in applications when the data is not drawn from normal distributions. As an alternative, non-parametric density estimation approaches – also known as Parzen window – can be used to estimate the probability of a given sample belonging to the same distribution function as the data set [5]. However, the memory requirement of the non-parametric approach is high. These techniques are also computationally expensive since they require the evaluation of a kernel function for each data sample in the training data set.

Support Vector Data Description (SVDD) is an elegant technique which uses support vectors in order to represent a data set [7]. The SVDD represents one class of known data samples in such a way that for a given test sample it can be recognized as known, or rejected as novel.

In this paper we present a novel incremental learning scheme for training SVDDs. The proposed learning technique employs two factors in learning the support vector machines which guarantee its convergence. The convergence can be achieved by optimizing on only two data points with a specific condition [1]. The condition requires that at least one of the data points does not satisfy the KKT conditions [3]. Our experimental results show that the incremental SVDD training achieves higher speed and require less memory than the online [8] and the canonical training [7].

The rest of the paper is organized as follows. Section 2 discusses the methodology used in this paper for the training of SVDDs. In Section 3 a comprehensive quantitative and qualitative set of experiments is carried out to compare the proposed incremental SVDD with the online and canonical training algorithms. Finally, Section 4 concludes the paper and proposes future directions of study.

2 Methodology

In this section we present the background modeling algorithm employed by our approach. In order to discuss the proposed algorithm we first introduce the SVDD method and its application. Then, we present the proposed algorithm for incremental training of the SVDDs.

2.1 Support Vector Data Description

A normal data description gives a closed boundary around the data which can be represented by a hyper-sphere (i.e. $F(R, a)$). The volume of this hyper-sphere with center a and radius R should be minimized while containing all the training samples x_i. To allow the possibility of outliers in the training set, slack variables $\epsilon_i \geqslant 0$ are introduced. The error function to be minimized is defined as:

$$F(R,a) = R^2 + C\sum_i \epsilon_i \left\|x_i - a\right\|^2 \leqslant R^2 + \epsilon_i \quad \forall i. \tag{1}$$

subject to:

$$\left\|x_i - a\right\|^2 \leqslant R^2 + \epsilon_i \quad \forall i. \tag{2}$$

In order to have a flexible data description, as opposed to the simple hyper-sphere discussed above, a kernel function $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$ is introduced. After applying the kernel and using Lagrange optimization, the SVDD function using kernels becomes:

$$L = \sum_i \alpha_i K(x_i, x_i) - \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \tag{3}$$
$$\forall \alpha_i : 0 \leqslant \alpha_i \leqslant C$$

When a sample falls in the hyper-sphere then its corresponding Lagrange multiplier is zero. Only data points with non-zero α_i are needed in the description of the data set, therefore they are called *support vectors* of the description. After optimizing the function in (3) the following equality constraint must hold:

$$\sum_i \alpha_i = 1 \tag{4}$$

Optimizing the functions in equation (3) is a Quadratic Programming (QP) problem. Generally the SVDD is used to describe large data sets. In such applications solving the above problem via standard QP techniques becomes intractable. The quadratic form of (3) needs to store a matrix whose size is equal to the square of the number of training samples. Due to this fact several algorithms have been proposed to present faster solutions to the above QP problem.

2.2 Incremental SVDD

Our incremental training algorithm is based on the theorem proposed by Osuna *et al.* in [3]. According to this theorem a large QP problem can be broken into series of smaller sub-problems. The optimization on these sub-problems converges when new samples are added as long as at least one violates the KKT conditions.

In the incremental learning scheme, at each step we add one sample to the training working set consisting of only support vectors. Assume we have a working set which minimizes the current SVDD objective function for the current data set. If a new sample belongs to the description then it satisfies the KKT conditions and its inclusion to the working set does not minimize the currently minimum objective function. If the KKT conditions do not hold for this sample, the SVDD optimization is solved for the new working set which includes the new sample. Since the working set contains only support vectors of the data set, its size is considerably smaller than the actual data set and the optimization can be performed efficiently.

From (4) it can be observed that Lagrange multipliers have a linear relationship. In order to further increase the optimization efficiency, we propose to solve the smallest possible sub-problem [1] which consists of only two samples. Since only the new sample violates the KKT conditions, at every step, our incremental learning algorithm chooses one sample from the working set along with the new sample and solves the optimization on these two samples.

Solving the QP problem for two Lagrange multipliers can be done analytically. This fact greatly reduces the burden of solving numerical QP problems and decreases the cost of the algorithm. Because there are only two multipliers at each step, the minimization constraint can be displayed in 2-D. The two Lagrange multipliers should satisfy the inequality constraint in (3) and the following linear equality constraint:

$$\alpha_1 + \alpha_2 = \gamma \quad : \quad \gamma \leqslant 1 \tag{5}$$

The main component of our incremental learning algorithm is based on an analytical method to solve for the two Lagrange multipliers. We first compute the constraints on each of the two multipliers. The two Lagrange multipliers should lie on a diagonal line in 2-D (equality constraint) within a rectangular box (inequality constraint). By expressing the two ends of this line we can easily find bounds for one of the two multipliers and from there proceed to the optimization process. Without loss of generality we consider that the algorithm starts with finding the upper and lower bounds on α_2 which are $H = min(C, \alpha_1^{old} + \alpha_2^{old})$ and $L = max(0, \alpha_1^{old} + \alpha_2^{old})$, respectively. The new value for α_2^{new} is computed by finding the maximum along the direction given by the linear equality constraint:

$$\alpha_2^{new} = \alpha_2^{old} + \frac{E_1 - E_2}{K(x_2, x_2) + K(x_1, x_1) - 2K(x_2, x_1)} \tag{6}$$

where E_i is the error in evaluation of each multiplier. The denominator in (6) is a step size (second derivative of objective function along the linear equality constraint). Next, we determine whether the new value for α_2^{new} has exceeded the bounds and needs to be clipped. We call this $\hat{\alpha}_2^{new}$. Finally, the new value for α_1 is computed using the linear equality constraint:

$$\alpha_1^{new} = \alpha_1^{old} + \alpha_2^{old} - \alpha_2^{new} \tag{7}$$

The process of optimizing the objective function for a multiplier pair iterates until all Lagrange multipliers satisfy the KKT conditions within a small error range.

```
1. Initialization
   C : Confidence, σ : bandwidth
2. For each frame at time t
    For each pixel x_ij
2.1. Training stage
     ISVD_i,j ← Inc. train(x_ij[t])
       % ISVD: Incremental SVD
2.2. Classification stage
     DV_i,j ← Test(x_ij[t],ISVD_i,j)
       % DV: Description Value
     Label pixel based on DV_i,j.
```

Figure 1. The SVDDM algorithm.

2.3 The Background Modeling Algorithm

Figure 1 shows the proposed algorithm in pseudo-code format[1]. The support vector data description confidence parameter C is the target false reject rate of the system, which accounts for the system tolerance. The Gaussian 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. 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. The training of the support vector descriptors for each pixel is performed using our proposed incremental learning scheme.

3 Experimental Results and Comparison

In this section we present a set of qualitative and quantitative experiments. The experiments are conducted in two main categories. The first set compares the performance of the proposed method in training the SVDDs with traditional methods on synthetic data sets. In the second set of experiments we show the performance of the proposed technique in a real background modeling application.

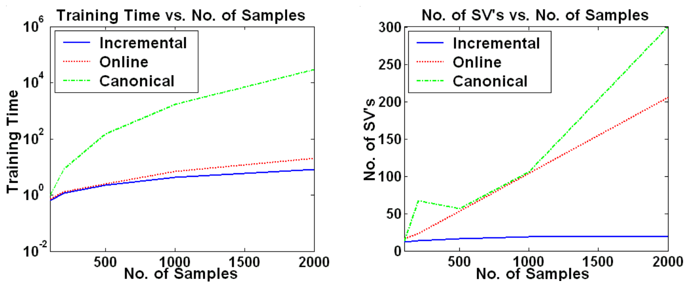
3.1 Comparison on Synthetic Data

In order to show the performance of the proposed method and its efficiency we compare the results obtained by our technique with those of the online SVDD [8] and batch SVDD [7]. We compare the speed of the algorithms as well as several error values for these techniques using different number of training samples and different data sets.

The SVVD Training Speed. In this section we compare the speed of incremental SVDD against its online and batch counterparts. The experiments are conducted in Matlab 6.5 on a P4 Core Duo processor with 1GB RAM.

Figure 2(a) shows the training speed of our incremental SVDD, online and batch versions against the number of training samples. As seen, the proposed SVDD training technique runs faster than both batch and online algorithms and its asymptotic speed is linear with the data set size. The online SVDD runs in linear time but for larger data sets its training time is higher than the proposed method. Our observation showed that this is due to the fact that online SVDD retains more unnecessary support vectors than the proposed technique. Notice that the training of a batch SVDD is in the order of magnitude slower than the proposed method.

[1]The proposed method is implemented in MATLAB 6.5, using Data Description toolbox [6].



(a) (b)

Figure 2. Comparison of the training algorithms on *banana* data set: (a) Training Speed. (b) No. of SV's.

Number of Support Vectors. A comparison of the number of retained support vectors for our technique and batch and online SVDD learning methods is presented in Figure 2(b). In this experiment, the parameters of the SVDD system are $C = 0.1$ and $\sigma = 5$ with a Gaussian kernel for all three classifiers. Our method keeps almost a constant number of support vectors. This can be interpreted as mapping to the same higher dimensional feature space for any given number of samples in the data set. Thus the proposed algorithm is suitable for applications in which the number of training samples increases in time, i.e. in the case of growing data sets such as long-term background modeling in videos.

Classification Performance. In Figure 3(a) the classification boundaries of the three SVDD training algorithms are shown. In this figure the dots are the training samples drawn from the *banana* data set and the circles represent the test data set drawn from the same distribution function. The $\star$, $\times$, and $+$ symbols are the support vectors of the incremental, online and canonical SVDD training algorithms, respectively. The proposed incremental learning retains fewer support vectors compared to both online and canonical training algorithms. From Figure 3(a), the decision boundaries of the classifier trained using the incremental algorithm (solid curve) is objectively more accurate than those trained by online (dotted curve) and canonical (dashed curve) methods.

Figure 3(b) shows the Receiver Operating Curves (ROC) of the three algorithms. The solid curve is the ROC of the incremental learning while dotted and dashed curves correspond to the online and canonical learning algorithms, respectively. Notice that the true positive rate is higher for small false positive rates in the our learning algorithm compared to both canonical and online learning. The ROC curves for other two data sets are similar to Figure 3(b) and are not included due to lack of space.

Figure 3(c) and (d) show a comparison of the classification boundaries between the three SVDD training algorithms on a 2-D normal distribution (*ellipse* data set) a complex distribution function in 2-D (the *egg* data set), respectively. From Figure 3, the incremental SVDD results in more accurate classification boundaries than both online and canonical versions. Notice that the proposed method keeps a smaller number of support vectors to describe data sets compared to the other two methods.

Error Evaluation. Table 1 compares the classification error, F_1 measure, number of the support vectors, and learning time for the three learning methods. The experiments are performed on three data sets ('*banana*', '*normal*', '*egg*') with 1000 training samples and 1000 test samples. The F_1 measure combines both the recall and the precision rates of a classifier:

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{8}$$

Classification Comparison. Table 2 compares the classification error, F_1 measure and training and classification asymptotic time for various classifiers. As seen, the proposed training of the SVDDs reaches very good classification rates compared to other methods. The SVDDs trained with batch and online training techniques give good classification accuracy but are outperformed by our method both in terms of accuracy and efficiency. In all systems, the trade-off parameter is set to be $C = 0.1$. Kernel bandwidth for the three SVDD methods and the Parzen window is $\sigma = 3.8$. $K = 3$ is selected for the number of Gaussians in the MoG and number of nearest neighbors in the K-NN method.

Table 1. Comparison of the proposed training algorithm with, online and batch methods on *banana*, *ellipse* and *egg* data sets of size 1000.

Training	Data Set	Error	F_1	No. SV's	Time
Banana	Proposed	0.005	0.997	19	4.2
	Online	0.075	0.961	104	6.9
	Canonical	0.085	0.956	106	1697
Ellipse	Proposed	0.013	0.993	6	3.72
	Online	0.100	0.947	105	4.1
	Canonical	0.110	0.994	108	2314
Egg	Proposed	0.065	0.966	8	3.85
	Online	0.095	0.950	101	3.7
	Canonical	0.128	0.932	87	1581

Table 2. Comparison of the classification error, F_1 measure, and asymptotic speeds with various classifiers on a *complex* data set of size 1000.

Classifier	Error	F_1	Training	Classification
Proposed	0.015	0.992	$O(1)$	$O(1)$
Batch SVD	0.100	0.947	$O(N)$	$O(N)$
Online SVD	0.103	0.945	$O(N)$	$O(N)$
KDE(Parzen)	0.114	0.940	$O(N)$	$O(N)$
MoG	0.143	0.923	$O(1)$	$O(1)$
K-means	0.150	0.919	$O(1)$	$O(1)$

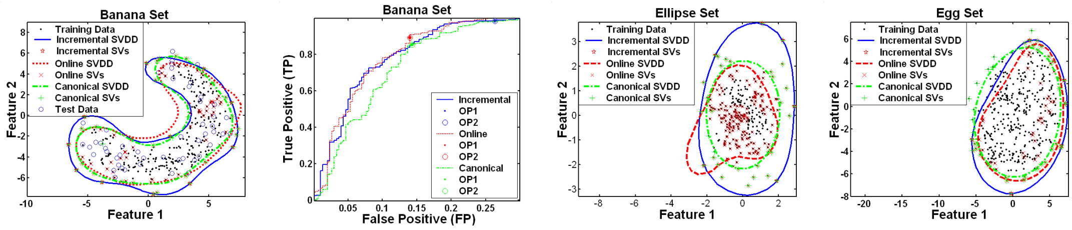
3.2 Application to Background Modeling

In this section we show the results of our method applied to background modeling in video sequences. We applied the incremental SVDD (INCSVDD) to speed up the process in section 2. We also compare the proposed method with traditional background modeling techniques.

Presence of Irregular Motion. By using the *water surface* video sequence, we compare the results of foreground region detection using our proposed method with a typical AKDE [5] and MoG [4]. For this comparison the sliding window of size L=150 is used in the AKDE method. The results of MoG are shown in Figure 4(b), the AKDE method results are shown in Figure 4(c) and the foreground masks detected by the proposed technique are shown in Figure 4(d). As it can be seen, the proposed method gives better detection since it generates a more accurate descriptive boundary on the training data, and does not need an explicit threshold to classify pixels as background or foreground.

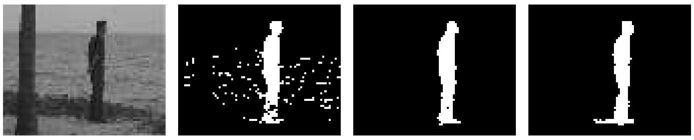
Difficult Scenarios. Figure 5 shows the results of foreground detection in videos using the proposed method. The water fountain in Figure 5(a), waving tree branches in Figure 5(b) and flickering lights in Figure 5(c) pose challenges in foreground detection. However, our method detects the foreground regions reliably and models the inherent changes in the background explicitly.

Comparison Summary. Table 3 provides a comparison between different traditional background modeling methods and our incremental SVDD (INCSVDD) technique. As seen in Table 3, the Wallflower method uses a K-means decision criterion, while



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

Figure 3. Comparison of incremental with batch and online SVDD: (a) *banana* **data set. (b) Receiver Operating Curve (ROC) for** *banana* **data set. (c)** *ellipse* **data set. (d)** *egg* **data set.**



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

Figure 4. Comparison of methods in presence of irregular motion in *water surface* **video: (a) Original frame. (b) MoG results. (c) AKDE results. (d) INCSVDD results.**

(a) (b) (c)

Figure 5. Results of the foreground detection using the proposed incremental SVDD for background modeling. Top row: Original videos. Bottom row: Detection results.

other systems –except INCSVDD– use a Bayes classifier. The only method which explicitly deal with the single-class classification is the proposed SVDD technique by fitting the description of data belonging to the background class. Other methods shown in the table use a binary classification scheme and use heuristics ([4] and [9]) or a more complex training scheme ([5]) to make it useful for the single-class classification problem of background modeling.

Only the INCSVDD technique is suitable for scenarios where there is a steady and slow change in the background. Other methods fail to build a long term representation for the background model because of the fact that their cost grows linearly by the number of training background frames. In scenarios where there is no empty set of background frames, called non-empty backgrounds, the INCSVDD method is suitable and works independently without any need to perform post-processing steps.

Table 3. Comparison between the proposed methods and traditional techniques.

Method	Classifier	Memory Req.*	Comp. Cost*
INSVDD	SVDD	$O(1)$	$O(1)$
AKDE[5]	Bayes	$O(N)$	$O(N)$
Spatio-temp[2]	Bayes	$O(N)$	$O(N)$
MoG[4]	Bayes	$O(1)$	$O(1)$
Wallflower[9]	K-means	$O(N)$	$O(N)$

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

4 Conclusions

Tracking moving objects in videos with quasi-stationary backgrounds is a challenging task. In order to detect moving foreground regions in such videos the background and its changes should be modeled. Support Vector Data Descriptors (SVDD) can be employed in order to analytically model a single class of data (the background pixels). This paper proposes a method to efficiently train an SVDD by solving the optimization problem. Another advantage of our technique is its constant memory requirements, since the proposed method only requires the support vectors for its incremental retraining. We showed the results of the proposed technique in a real world background modeling application, while comparing the system with traditional techniques, both quantitatively and qualitatively.

The proposed incremental training of the SVDD is a general method that can be employed in many novelty detection applications such as face detection. The issue in face detection systems is that samples of only one class of the data (faces) are available. Most object recognition systems can be represented as a single-class classification application, hence the proposed training algorithm can be used to train their corresponding SVDD.

References

[1] J. Platt. Fast Training of Support Vector Machines using Sequential Minimal Optimization. *Advances in Kernel Methods - Support Vector Learning.*, MIT Press:185–208., 1998.

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

[3] E. Osuna, R. Freund, and F. Girosi. Improved Training Algorithm for Support Vector Machines. *In Proc. Neural Networks in Signal Processing*, 1997.

[4] C. Stauffer and W. Grimson. Learning Patterns of Activity using Real-Time Tracking. *IEEE Transactions on PAMI*, 22(8):747–757, August 2000.

[5] 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.

[6] D. Tax. Ddtools, the data description toolbox for matlab, Augustus 2005. version 1.11.

[7] D. Tax and R. Duin. Support Vector Data Description. *Machine Learning*, 54(1):45–66., 2004.

[8] D. Tax and P. Laskov. Online SVM Learning: from Classification and Data Description and Back. *Neural Networks and Signal Processing*, (1):499–508., 2003.

[9] 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.