INCREMENTAL SVDD TRAINING: IMPROVING EFFICIENCY OF BACKGROUND MODELING IN VIDEOS

Alireza Tavakkoli [1], Mircea Nicolescu[1], Monica Nicolescu[2], George Bebis[1]
[1] Computer Vision Laboratory [2] Robotics Laboratory
University of Nevada, Reno
NV, USA 89557
email: {tavakkol,mircea,monica,bebis}@cse.unr.edu

ABSTRACT

Tracking moving objects in videos with quasi-stationary backgrounds is one of the most important and challenging tasks in video processing applications. In order to detect moving foreground regions in such videos the background and its changes should be modeled to help detecting moving regions of interest. Support Vector Data Descriptors (SVDD) can be employed in order to analytically model the background and explicitly account for its inherent changes. The major draw back of the SVDD modeling is the issue of training of the SVDD which is a quadratic programming (QP) problem. In this paper we propose a method to efficiently train the SVDD's. The advantages of our technique are its low memory requirement and its efficiency in terms of speed. The proposed method runs in constant time with respect to the size of the training data set since its retraining is performed only on the support vector working set.

KEY WORDS

Background Modeling, Support Vector Data Description, Incremental Training, Computer Vision

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 [2].

Parametric density estimation methods, such as Mixture of Gaussians techniques (MoG), assume that the data is generated from a mixture of normal distributions with different weights, means and covariance matrices [8]. In order to compute the parameters of the Gaussians Expectation Maximization (EM) technique is adopted [1]. Due to computational complexity of the EM algorithm the MoG modeling technique is slow. An online version of MoG is presented by Lee in [5] which uses a recursive technique to update the parameters of the Gaussians. However, the parametric density estimation techniques may not be useful 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 [3], [9]. 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. In order to address these issues Tavakkoli *et al.* proposed a non-parametric recursive modeling technique in [10] to model background pixels in videos.

Support Vector Data Description (SVDD) is an elegant technique which uses support vectors to represent a data set [13]. The SVDD represents one class of known samples in such a way that for a given test sample it can be recognized as known, or rejected as novel. Tavakkoli *et al.* in [11] proposed using the support vector data description to model the background of videos. These descriptors are trained using an online learning algorithm [14].

Tax and Laskov in [14] presented an online training of SVDD using a limited sample set. Their method sweeps through the data set, but instead of solving the problem by keeping the whole data samples as the working set, it keeps a portion of the data. This system adds a new sample and removes the most irrelevant one at each run. However, this system computes the optimum on the limited working set by making approximations to solve the reduced optimization subproblem. We observed, the online SVDD training algorithm has some limitations when the number of training samples increases, since it reduces the size of the optimization problem but still solves it in a canonical manner.

In this paper we present a novel incremental learning scheme for SVDD training. The convergence of training can be achieved by optimizing on only two data points with a specific condition [4]. The condition requires that at least one of the data points does not satisfy the Karush-Kuhn-Tucker (KKT) conditions [7]. Our experimental results show that the incremental SVDD training achieves higher speed and require less memory than the online [14] and the canonical (batch) training of SVDD [13].

The rest of the paper is organized as follows. Sec-

```
1. Initialization;
   C : Confidence, σ : bandwidth
2. For each frame at time t
    For each pixel x_ij
     2.1. Training stage
          SVD_ij ← IncTrain(x_ij[t])
     2.2. Classification stage
          Value_ij ← Test(x_ij[t],SVD_ij)
          Label pixel based on Value_ij
```

Figure 1. The SVDDM algorithm.

tion 2 discusses the methodology used in this paper for the training of SVDD's. 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 method we first introduce the SVDD method and its application. Then, we present the proposed algorithm for incremental training of the SVDD's.

2.1 The 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, 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. The training of the support vector descriptors for each pixel is performed using our proposed incremental learning scheme.

2.2 Support Vector Data Description

Data domain description concerns the characteristics of a data set [13] whose boundary can be used to detect novel samples (outliers). 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. As proposed in [13] the extension to more complex distributions is straightforward using kernels. To allow the possibility of outliers in the training set, slack variables $\epsilon_i \geq 0$ are introduced. The error function to be minimized is defined as:

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

subject to:

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

Using Lagrange optimization the above results in:

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

When a sample falls in the hyper-sphere then its corresponding Lagrange multiplier is $\alpha_i \geq 0$, otherwise it is zero. After optimizing the function in (3) the following equality constraint must hold:

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

It can be observed that 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. Given the support vectors x_i, a new test sample z_t can be classified as known/novel data using:

$$\|z_t - a\|^2 = (z_t \cdot z_t) - 2\sum_i \alpha_i (z_t \cdot x_i) + \sum_{i,j} \alpha_i \alpha_j (x_i \cdot x_j) \tag{5}$$

where α_i are Lagrange multipliers and $||z_t - \mathbf{a}||$ is the distance of the new sample from the description center. The sample is classified as novel if the distance is larger than R.

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. This kernel maps the data into a higher dimensional space, where it is described by the simple hyper-sphere boundary. Instead of a simple dot product of the training samples $(x_i \cdot x_j)$, the dot product is performed using a kernel function. Several kernels have been proposed in the literature [16]. Among these, the Gaussian kernel gives a closed data description, $K(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|^2}{\sigma^2}\right)$. The SVDD function using kernels becomes:

$$L = \sum_i \alpha_i K(x_i, x_i) - \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \tag{6}$$
$$\forall \alpha_i : 0 \leq \alpha_i \leq C$$

Optimizing the functions in equations (3) and (6) 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 (6) 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.

[1]The proposed method is implemented in MATLAB 6.5, using Data Description toolbox [12].

2.3 Incremental SVDD

Our incremental training algorithm is based on the theorem proposed by Osuna *et al.* in [7]. 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 sample violates the KKT conditions.

In the incremental learning scheme at each step we add one sample to the training working set. The training working set only consists of the support vectors. This is a direct result of the above theorem. 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. This means that its inclusion to the working set does not minimize the currently minimum objective function, and thus it will be discarded. 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 [4] 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 this two sample sub set.

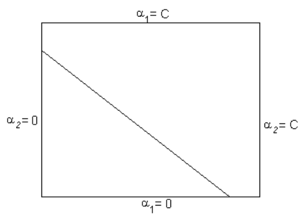


Figure 2. The two Lagrange multipliers should satisfy the inequality constraint (3) and the linear equality (7).

Solving the QP problem for two Lagrange multipliers can be done analytically. The two Lagrange multipliers should satisfy the inequality constraint in (3) and the following linear equality constraint (Figure 2):

$$\alpha_1 + \alpha_2 = \gamma \quad : \quad \gamma \leq 1 \tag{7}$$

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. From Figure 2 the two Lagrange multipliers should lie on a diagonal line (equality constraint) within the 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 we start 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 of the linear equality constraint:

$$\alpha_2^{new} = \alpha_2^{old} + \frac{E_1 - E_2}{K(x_2, x_2) + K(x_1, x_1) - 2K(x_2, x_1)} \tag{8}$$

where E_i is the error in evaluation of each multiplier in equation (5). The denominator in (8) 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:

$$\alpha_1^{new} = \alpha_1^{old} + \alpha_2^{old} - \hat{\alpha}_2^{new} \tag{9}$$

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 of the SVDD's with the traditional canonical and online training methods, on synthetic data sets. In the second set of experiments we show the performance of the proposed technique on real videos.

3.1 Comparison

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 [14] and canonical SVDD [13]. 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.

Table 1. Speed comparison of the incremental, online and canonical SVDD on the *banana* data set.

Training Set Size	Incremental SVDD	Online[14] SVDD	Canonical [13] SVDD
100	0.66	0.73	1.00
200	1.19	1.31	8.57
500	2.19	2.51	149.03
1000	4.20	6.93	1697.2

The SVVD Training Speed. In this section we compare the speed of incremental SVDD against its online and canonical counterparts. The experiments are conducted in Matlab 6.5 on a P4 Core Duo processor with 1GB RAM. The reported training times are in seconds. Table 1 Shows a report the training speed of our incremental SVDD, online and canonical versions on various sizes of data set. As seen, the proposed SVDD training technique runs faster than both canonical 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 more 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. As expected, both online and our SVDD training methods are considerably faster than the canonical training of the classifier. Notice that the training time of a canonical SVDD for 2000 training points is not available since it takes hours to finish the experiment.

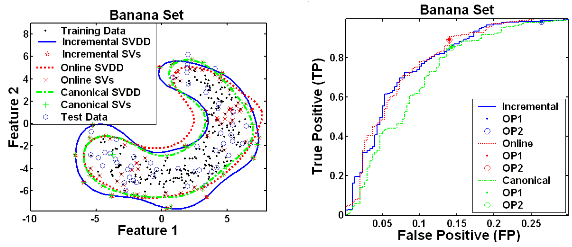
Table 2. Comparison of the number of support vectors for the incremental, online and canonical SVDD on the *banana* data set.

Training Set Size	Incremental No. of SV's	Online [14] No. of SV's	Canonical [13] No. of SV's
100	12	16	14
200	14	23	67
500	16	53	57
1000	19	104	106

Number of Support Vectors. A comparison of the number of retained support vectors for our technique and canonical and online SVDD learning methods is presented in Table 2. In this experiment the parameters of the SVDD system are $C = 0.1$ and $\sigma = 5$ with a Gaussian kernel for all three classifiers. As it can be observed both online and canonical SVDD training algorithms increase the number of support vectors as the size of the data set increases. However, 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.

Notice that by increasing the number of training samples the proposed SVDD training algorithm requires less memory than both online and canonical algorithms. This makes the proposed algorithm very suitable for applications in which the number of training sample increases by time, i.e. in the case of growing data sets. Since the number of support vectors is inversely proportional to the classification speed of the system in (5), the processing time of a classifier trained by the proposed method is constant with the number of samples compared with the canonical and online methods.

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

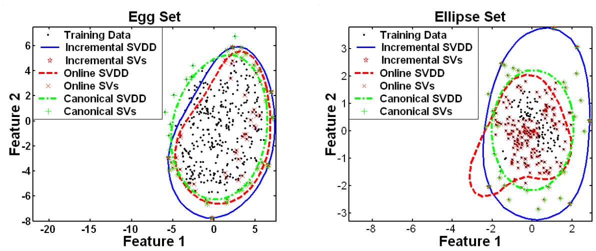


(a) (b)

Figure 3. Comparison of incremental with canonical and online SVDD: (a) Classification boundaries (b) Receiver Operating Curve (ROC).

Figure 3(b) shows the comparison between Receiver Operating Curve (ROC) of the three algorithms. The solid curve is the ROC of the incremental learning while the dotted and the 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 case of the proposed incremental learning algorithm compared to both canonical and online learning. This can be expected since the proposed method explicitly takes the small trade-off parameter (C) into account by learning the support vectors incrementally. As it can be seen from the figure, the true positive rate for the proposed method is higher than the canonical method. This shows that the proposed method, under the same conditions and with the same parameters, has higher precision and recall rates.

Figure 4 shows a comparison of the classification boundaries and support vectors between the three SVDD training algorithms. Figure 4(a) shows the result of classification on a 2-D normal distribution and Figure 4(b) is the experiment on a data set drawn from a more complex distri-



(a) (b)

Figure 4. Comparison of incremental with online and canonical SVDD: (a) Complex (egg) data set. (b) Normal data set.

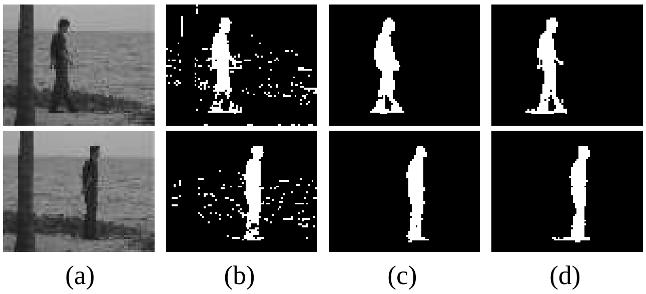


Figure 5. Water surface video: comparison of methods in presence of irregular motion. (a) Original frame, (b) MoG, (c) AKDE, (d) INCSVDD.

bution function in 2-D (egg shape). As seen from the figure the proposed 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 both data sets compared to the other two methods.

3.2 Application to Background Modeling

In this section we show the results of our method applied for 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 the traditional background modeling techniques.

Comparison in the 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 [9] and MoG [8]. For this comparison the sliding window of size L=150 is used in the AKDE method. The results of MoG are shown in Figure 5(b), the AKDE method results are shown in Figure 5(c) and the foreground masks detected by the proposed technique are shown in Figure 5(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 a threshold to classify pixels as background or foreground.

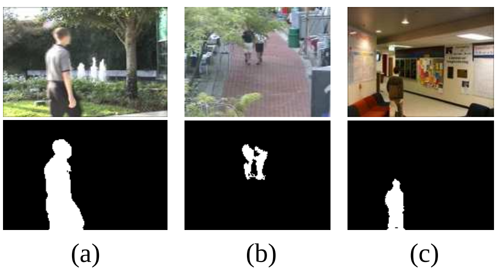


Figure 6. Results of the foreground detection using the proposed incremental SVDD to the background modeling. Top row: Original videos. Bottom row: Detection results.

Detection results in difficult scenarios. Figure 6 shows the results of foreground detection in videos using the proposed method. The water fountain in Figure 6(a), waving tree branches in Figure 6(b) and flickering lights and monitor in Figure 6(c) pose challenges in foreground detection. However, as seen 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 technique. The comparison includes the classification type, memory requirements, computation cost and type of parameter selection.

As seen in Table 3, the Wallflower method uses a K-means decision criterion where other systems except both SVDD and incremental SVDD (INCSVDD) use a Bayes classifier. The only methods which explicitly deal with the single class classification are the two SVDD techniques by fitting the description of data belonging to the background class in their rather simple training stage. Other methods shown in the table use a binary classification scheme and use heuristics ([3], [8] and [15]) or a more complex training scheme ([9] and [10]) to make it useful for the single-class classification problem of background modeling.

From the computational cost row, the only method suitable for scenarios where there is a steady and very slow motion in the background is the INCSVDD technique. 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, as it can be seen from Table 3. Also 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.

4 Conclusions

Tracking moving objects in videos with quasi-stationary backgrounds is a very 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 the background.

SVDD training is a quadratic programming (QP) problem. By increasing the number of training samples, solving this QP problem becomes intractable both in terms of memory requirements and speed. This paper proposes a method to efficiently train an SVDD. The proposed algorithm solves the optimization problem by reducing its size to the number of support vectors, thus making it run in linear time with respect to the number of training samples.

Another advantage of our technique is in its constant memory requirements. The experimental results show its superiority over both the canonical SVDD and the traditional online training methods. We showed the results of

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

	INCSVDD	SVDD[11]	AKDE[9]	KDE[3]	Spatio-temp[6]	MoG[8]	Wallflower[15]
Automated	Yes	Yes	Yes	No	No	No	No
Post proc.	No	No	No	No	Yes	No	No
Classifier	INCSVD	SVD	Bayes	Bayes	Bayes	Bayes	K-means
Memory req.*	$O(1)$	$O(1)$	$O(N)$	$O(N)$	$O(N)$	$O(1)$	$O(N)$
Comp. cost*	$O(1)$	$O(N)$	$O(N)$	$O(N)$	$O(N)$	$O(1)$	$O(N)$

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

the proposed technique in a background modeling system, while comparing the system with traditional techniques.

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 detection and recognition systems can be presented as a single-class classification applications and the proposed training algorithm can be used to train their corresponding SVDD.

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] Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press, Walton Street, Oxford OX26DP (1994)

[2] Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley, New York (2001)

[3] Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.: Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance. In proceedings of the IEEE **90**, 1151–1163. (2002)

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

[5] Lee, D.S.: Effective Gaussian Mixture Learning for Video Background Subtraction. IEEE Transactions on PAMI **27**(5), 827–832 (2005)

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

[7] Osuna, E., Freund, R., Girosi, F.: Improved Training Algorithm for Support Vector Machines. In Proc. Neural Networks in Signal Processing (1997)

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

[9] Tavakkoli, A., Nicolescu, M., Bebis, G.: Automatic Statistical Object Detection for Visual Surveillance. In proceedings of IEEE Southwest Symposium on Image Analysis and Interpretation pp. 144–148 (2006)

[10] Tavakkoli, A., Nicolescu, M., Bebis, G.: Robust Recursive Learning for Foreground Region Detection in Videos with Quasi-Stationary Backgrounds. In proceedings of 18th International Conference on Pattern Recognition (2006)

[11] Tavakkoli, A., Nicolescu, M., Bebis, G.: A support vector data description approach for background modeling in videos with quasi-stationary backgrounds. to appear in the International Journal on Artificial Intelligence Tools (2007)

[12] Tax, D.: Ddtools, the data description toolbox for matlab (2005). Version 1.11

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

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

[15] Toyama, K., Krumm, J., Brumitt, B., Meyers, B.: Wallflower: principles and practice of background maintenance. In proceedings of ICCV **1**, 255–261 (1999)

[16] Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)