

Article

Online Least Squares One-Class Support Vector Machines-Based Abnormal Visual Event Detection

Tian Wang ^{1,*}, Jie Chen ², Yi Zhou ³ and Hichem Snoussi ¹

¹ Institut Charles Delaunay-LM2S-UMR STMR 6279 CNRS, University of Technology of Troyes, Troyes 10004, France; E-Mail: hichem.snoussi@utt.fr

² Observatoire de la Côte d'Azur-UMR 7293 CNRS, University of Nice Sophia-Antipolis, Nice 06108, France; E-Mail: jie.chen@unice.fr

³ College of Information Science and Technology, Dalian Maritime University, Dalian 116026, China; E-Mail: yi.zhou@dlmu.edu.cn

* Author to whom correspondence should be addressed; E-Mail: wangtian8704@gmail.com; Tel.: +33-3-2571-8567; Fax: +33-3-2571-5649.

Received: 30 October 2013; in revised form: 29 November 2013 / Accepted: 29 November 2013 / Published: 12 December 2013

Abstract: The abnormal event detection problem is an important subject in real-time video surveillance. In this paper, we propose a novel online one-class classification algorithm, online least squares one-class support vector machine (online LS-OC-SVM), combined with its sparsified version (sparse online LS-OC-SVM). LS-OC-SVM extracts a hyperplane as an optimal description of training objects in a regularized least squares sense. The online LS-OC-SVM learns a training set with a limited number of samples to provide a basic normal model, then updates the model through remaining data. In the sparse online scheme, the model complexity is controlled by the coherence criterion. The online LS-OC-SVM is adopted to handle the abnormal event detection problem. Each frame of the video is characterized by the covariance matrix descriptor encoding the moving information, then is classified into a normal or an abnormal frame. Experiments are conducted, on a two-dimensional synthetic distribution dataset and a benchmark video surveillance dataset, to demonstrate the promising results of the proposed online LS-OC-SVM method.

Keywords: abnormal detection; optical flow; covariance matrix descriptor; online least squares one-class SVM

1. Introduction

Visual surveillance is one of the major research areas in computer vision. After recording events by a visual sensor, such as a camera, obtaining detailed information of individual or crowd behavior is a challenging object in this area; automatic abnormal event detection is required to provide convenience, safety and an efficient lifestyle for humanity [1]. An abnormal event is defined as behavior deviating from what one expects. For example, a pedestrian panic in a public region: the people are running in the plaza, where people are usually strolling. As shown in Figure 1a, a normal scene is illustrated, the people are walking. In Figure 1b, the people are suddenly running in different directions; this scene is considered abnormal. If a system can detect this event, which might imply a safety risk, security staff can be alerted to take emergency response procedures. The abnormal event detection is a content-based video analysis problem; it includes two technologies: a feature representation of an event model and an abnormal event detection approach.

Figure 1. Examples of the normal and abnormal scenes. (a) A normal lawn scene: all the people are walking; (b) An abnormal lawn scene: all the people are running.



In [2–4], abnormal detection approaches with behavioral models were introduced. The behavior pattern modeled by adopting optical flow or pixel change history (PCH) was represented by Bayesian models. In [5], the motion feature, including the position, direction and velocity, was modeled by latent Dirichlet allocation. In [6], the abnormal vehicle behavior at intersections was detected via a stochastic graph model based on the Markovian approach. The behavior was labeled as abnormal when the current motion pattern cannot be recognized as any state of the system or a particular sequence of states cannot be parsed with the stochastic model. These works relied on an explicit signal statistical model, and the abnormal events were the ones interpreted as statistical model abrupt changes, maximum likelihood or Bayesian estimation theory [7]. The signal model together with probabilistic assumption techniques are usually extremely powerful insofar as an accurate model exists; these methods are effective in several scenarios. However, there are various situations where a robust and tractable model cannot be obtained. This raises the need for model-free methods.

On the other hand, low-level motion features were employed. In [8], the authors presented an algorithm monitoring optical flow in a set of fixed spatial positions. The similarity of the model was

computed to detect abnormal patches. In [9], the irregular behavior of images and videos was detected by comparing the likelihood of patches via a probabilistic graphical model. These methods based on separated patches, benefiting from the partial knowledge of the image, do not exploit the global information of the frame.

Trajectory information is also adopted to detect abnormal events. In [10,11], the authors presented a method for anomalous event detection by means of trajectory analysis. The trajectories were subsampled to a fixed-dimension vector representation and clustered with an one-class support vector machine (SVM). In [12], alarm detection of traffic was performed on the basis of the parameters of the moving objects and their trajectories by using semantic reasoning and ontologies. In [13], vision-based abnormal events for home healthcare systems were detected by using shape feature variation and 3D trajectory. Tracking-based algorithms are likely to fail in crowded scenes.

We consider the model-free approach, which does not require an explicit statistical model. To be accurate, the support vector machine (SVM) classification method is relied on in this paper. Inspired by the satisfactory performance of a covariance feature descriptor representing object in a tracking problem, a covariance descriptor characterizes the moving information of a global frame. In a tracking problem, the covariance descriptor is constructed of the blob intensity or color for template matching. In this paper, covariance encodes the optical flow of the global frame.

The rest of the paper is organized as follows. In Section 2, related works are briefly reviewed. In Section 3, the online least squares one-class support vector machine (online LS-OC-SVM) classification method is originally derived. In Section 4, a covariance matrix descriptor is described to provide feature vectors for the classification algorithm. In Section 5, we propose abnormal detection methods based on the online LS-OC-SVM. In Section 6, we present the results on synthetic data and real-world video scenes. Finally, Section 7 concludes the paper.

2. Related Work

SVM is usually trained in a batch model, *i.e.*, all training data are given *a priori* and are learned together. If additional training data arrive afterward, the SVM must be retrained from scratch [14]. In the problem of abnormal event detection in video surveillance, the normal sequence for training may last for a long time. It is impractical to train the big training set of normal samples as one batch together. If a new datum is added to a large training set, it will likely have only a minimal effect on the previous decision surface. Resolving the problem from scratch seems computationally wasteful. Considering these two aspects, the online strategy is considered in our work to adapt to the computational and the memory requirement.

Some online learning algorithms for SVM were derived based on analyzing the change of Karush–Kuhn–Tucker (KKT) conditions while updating the classifier. In [15], new arrival data along with the data violating the KKT conditions, and the support vectors from the last iteration, were considered as a new training dataset to train the classifier at the current step. The iteration will be stopped when all data satisfy the KKT conditions. In [16,17], the authors analyzed the change of the KKT conditions when one datum was included into, or removed from, the training set; then, a so-called

bookkeeping step was used to compute the new coefficients of the classifier to achieve an online update for a two-class SVM. Useful implementation issues on incremental SVM were presented in [18].

In [7], it was argued that the binary classification algorithm in [16] cannot be directly implemented for a one-class problem. In [7,19], the authors considered the change of the normal model over time and online identified outliers using previous data vectors in a sliding time window. Two one-class SVM classifiers, which preceded and followed the present instant, were compared. A change in the statistics of the time series was likely to occur when the resulting machines were different. The sliding time window approach was considered in [20], with an application on wireless sensor networks. This method, adopting sliding window formulation, is not inherently online, since it requires repeated batch training of new machines.

In [21], an online one-class SVM was presented following the idea of [22]: an exponential window was applied to the data to suit it to an adaptive scenario where the solution was able to track the changes of the data distribution and to forget old patterns. This algorithm is based on the slow-varying assumption.

Some online one-class SVM classification methods were proposed based on support vector data description (SVDD) [23,24], the hypersphere one-class SVM formulation. In [25], an online one-class classification method was proposed, a least squares optimization problem was considered and the model complexity was controlled by the coherence criterion. In [26], a method was proposed to reduce space and time complexities. It reduced the training set size during the training procedure by removing data having a high probability of becoming non-support-vectors.

In order to sidestep the difficulty in the nature of the constrained quadratic optimization problem, we derive an online version of the hyperplane one-class SVM [27] based on the least squares regularization. In the least squares SVM version, one finds the solution by solving a linear system instead of a quadratic programming problem. This advantage comes from the use of equality instead of inequality constraints in the problem formulation [28]. Least squares one-class SVM (LS-OC-SVM) was proposed in [29], without considering the sparsity of the hyperplane representation. It is thus inappropriate to detect abnormal events online. In the following, we shall derive an online version of the least squares one-class SVM, then propose a sparsification representation of the detector.

3. Classification

In this section, we introduce the derivation of the proposed online least square one-class support vector machine (online LS-OC-SVM). In abnormal detection problems, it is supposed that the samples from a positive class are obtainable. A density will only exist if the underlying probability measure possesses an absolutely continuous distribution function, but the general problem of estimating the measure for a large class of sets is not solvable [27,30]. The one-class SVM framework is then suitable to the specificity of the abnormal event detection where only normal scene data are available. Support vector machine (SVM) was initially proposed by Vapnik and Lerner [31], attempting to find a compromise between the minimization of empirical risk and the prevention of overfitting. By applying a kernel trick, SVM can handle nonlinear classification problems [10,32–34]. Based on the theoretical foundation of SVM and the soft-margin trick [35,36], one-class SVM is proposed to address the problem where only one-category (the positive) samples with a few outliers are available. In this section, after a brief review

of one-class SVM and least squares one-class SVM on a batch model, an online training algorithm is proposed. A sparsified version of the algorithm will then be provided for further adapting to critical online requirements.

3.1. One-Class SVM

One-class SVM (OC-SVM) aims to determine a suitable region in the input data space, \mathcal{X} , which includes most of the samples drawn from an unknown probability distribution, P . It detects objects that resemble training samples. The hypersphere one-class SVM was proposed in [23,24]. It identified outliers by fitting a hypersphere with a minimal radius. The hyperplane one-class SVM was an extended version of the original SVM to one-class problems [27,36]. It identified outliers by fitting a hyperplane from the origin. In our work, we adopt the hyperplane one-class SVM, which is formulated as a constrained minimization optimization problem:

$$\min_{\mathbf{w}, \xi, \rho} \frac{1}{2} \|\mathbf{w}\|^2 - \rho + C \sum_{i=1}^n \xi_i \quad \text{subject to} \quad \langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle \geq \rho - \xi_i, \quad \xi_i \geq 0 \quad (1)$$

where $\mathbf{x}_i \in \mathcal{X}, i \in \{1 \dots n\}$, are n training samples in the input data space, \mathcal{X} , and ξ_i is the slack variable for penalizing the outliers. The hyperparameter, C , is the weight for restraining the slack variable. It tunes the number of acceptable outliers and, thus, enables the analyzing of noisy data points. $\|\cdot\|$ denotes the Euclidean norm of a vector. The decision hyperplane is given by the equation:

$$\langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle - \rho = 0 \quad (2)$$

The nonlinear function, $\Phi : \mathcal{X} \rightarrow \mathcal{H}$, maps datum \mathbf{x}_i from the input space, \mathcal{X} , into the feature space, \mathcal{H} , which allows us to solve a nonlinear classification problem by designing a linear classifier in the feature space. \mathbf{w} defines a hyperplane in the feature space separating the projections of training data from the origin. A positive definite kernel function, κ , is defined as $\kappa(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$, which implicitly maps the training or testing data, \mathbf{x} , into a higher (possibly infinite) dimensional feature space. Introducing the Lagrangian multipliers, α_i , the decision function in the input data space, \mathcal{X} , is given by:

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^n \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}) - \rho\right) \quad (3)$$

if $f(\mathbf{x}) = -1$, the datum, \mathbf{x} , is classified as abnormal; otherwise, \mathbf{x} is classified as normal.

3.2. Least Squares One-Class SVM

Least squares SVM (LS-SVM) was proposed by Suykens in [37,38]. By using the quadratic loss function, Choi proposed least squares one-class SVM (LS-OC-SVM) [29]. LS-OC-SVM extracts a hyperplane as an optimal description of training objects in a regularized least squares sense. It can be written as the following objective function:

$$\min_{\mathbf{w}, \xi, \rho} \frac{1}{2} \|\mathbf{w}\|^2 - \rho + \frac{1}{2} C \sum_{i=1}^n \xi_i^2 \quad \text{subject to} \quad \langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle = \rho - \xi_i \quad (4)$$

The condition for the slack variables in OC-SVM, $\xi_i \geq 0$, is no longer in need. The variable, ξ_i , represents an error caused by a training object, \mathbf{x}_i , with respect to the hyperplane. The definitions of the other parameters in Equation (4) are the same as the ones in OC-SVM. The associated Lagrange is:

$$L = \frac{1}{2} \|\mathbf{w}\|^2 - \rho + \frac{C}{2} \sum_{i=1}^n \xi_i^2 - \sum_{i=1}^n \alpha_i (\mathbf{w}^\top \Phi(\mathbf{x}_i) - \rho + \xi_i) \quad (5)$$

Setting derivatives of Equation (5) with respect to primal variables, \mathbf{w} , ξ_i , ρ and α_i , to zero, we have the following stationarity conditions:

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^n \alpha_i \Phi(\mathbf{x}_i) \quad (6)$$

$$\frac{\partial L}{\partial \xi_i} = 0 \quad \Rightarrow \quad C \xi_i = \alpha_i \quad (7)$$

$$\frac{\partial L}{\partial \rho} = 0 \quad \Rightarrow \quad \sum_{i=1}^n \alpha_i = 1 \quad (8)$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \quad \Rightarrow \quad \mathbf{w}^\top \Phi(\mathbf{x}_i) + \xi_i - \rho = 0 \quad (9)$$

Substituting Equations (6)–(8) into (9) yields:

$$\sum_{i,j=1}^n \alpha_i \Phi^\top(\mathbf{x}_i) \Phi(\mathbf{x}_j) + \frac{\alpha_i}{C} - \rho = 0 \quad (10)$$

For all $i = 1, 2, \dots, n$, we can rewrite Equation (10) in matrix form as:

$$\begin{bmatrix} \mathbf{K} + \frac{\mathbf{I}}{C} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ -\rho \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \quad (11)$$

where \mathbf{K} is the Gram matrix with (i, j) -th entry $\kappa(\mathbf{x}_i, \mathbf{x}_j)$, \mathbf{I} is the identity matrix with the same dimension as Gram matrix \mathbf{K} and $\boldsymbol{\alpha}$ is the column vector with i -th entry α_i for training sample \mathbf{x}_i . $\mathbf{1}$ and $\mathbf{0}$ are all-one and all-zero column vectors, respectively, with compatible lengths. The parameters, $\boldsymbol{\alpha}$ and ρ , could be obtained by:

$$\begin{bmatrix} \boldsymbol{\alpha} \\ -\rho \end{bmatrix} = \begin{bmatrix} \mathbf{K} + \frac{\mathbf{I}}{C} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \quad (12)$$

The hyperplane is then described by:

$$f(\mathbf{x}) = \sum_{i=1}^n \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}) - \rho = 0 \quad (13)$$

The distance, $dis(\mathbf{x})$, of a datum, \mathbf{x} , with respect to the hyperplane is calculated by:

$$dis(\mathbf{x}) = \frac{|f(\mathbf{x})|}{\|\boldsymbol{\alpha}\|} = \frac{|(\sum_{i=1}^n \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}) - \rho)|}{\|\boldsymbol{\alpha}\|} \quad (14)$$

where \mathbf{x}_i is a training sample, $\|\boldsymbol{\alpha}\|$ is the two-norm of vector $\boldsymbol{\alpha}$. An object with a low $dis(\mathbf{x})$ value lies close to the hyperplane thus resembles the training set better than other objects with high $dis(\mathbf{x})$ values. The distance, $dis(\mathbf{x})$, is used as a proximity measure to determine the normal and abnormal class of the data [29].

3.3. Online Least Squares One-Class SVM

In an online learning scheme, the training data continuously arrive. We thus need to tune hyperparameters in the objective function and the hypothesis class in an online manner [17]. Let α_n , \mathbf{K}_n and \mathbf{I}_n denote the coefficient, Gram matrix and identity matrix at the time step, n , respectively. The parameters of LS-OC-SVM $[\alpha_n \ -\rho_n]^\top$ at the time step, n , could be calculated as:

$$\begin{bmatrix} \alpha_n \\ -\rho_n \end{bmatrix} = \begin{bmatrix} \mathbf{K}_n + \frac{\mathbf{I}_n}{C} & \mathbf{1}_n \\ \mathbf{1}_n^\top & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0}_n \\ 1 \end{bmatrix} \quad (15)$$

In order to proceed, recall the matrix inverse identity for matrices A , B , C and D with suitable sizes [39]:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} A^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} -A^{-1}B \\ 1 \end{bmatrix} \times (D - CA^{-1}B)^{-1} \times [-CA^{-1} \quad 1] \quad (16)$$

The matrix, \mathbf{K}_n , with diagonal loading $\frac{\mathbf{I}_n}{C}$ can be calculated recursively with respect to time step n by:

$$\begin{bmatrix} \mathbf{K}_{n+1} + \frac{\mathbf{I}_{n+1}}{C} \end{bmatrix}^{-1} \quad (17)$$

$$= \begin{bmatrix} \mathbf{K}_n + \frac{\mathbf{I}_n}{C} & \boldsymbol{\kappa}_{n+1} \\ \boldsymbol{\kappa}_{n+1}^\top & \kappa_{n+1} + \frac{1}{C} \end{bmatrix}^{-1} \quad (18)$$

$$= \begin{bmatrix} (\mathbf{K}_n + \frac{\mathbf{I}_n}{C})^{-1} & \mathbf{0}_n \\ \mathbf{0}_n^\top & 0 \end{bmatrix} + \frac{1}{(\kappa_{n+1} + \frac{1}{C}) - \boldsymbol{\kappa}_{n+1}^\top (\mathbf{K}_n + \frac{\mathbf{I}_n}{C})^{-1} \boldsymbol{\kappa}_{n+1}} \begin{bmatrix} -(\mathbf{K}_n + \frac{\mathbf{I}_n}{C})^{-1} \boldsymbol{\kappa}_{n+1} \\ 1 \end{bmatrix} \begin{bmatrix} -\boldsymbol{\kappa}_{n+1}^\top (\mathbf{K}_n + \frac{\mathbf{I}_n}{C})^{-1} & 1 \end{bmatrix} \quad (19)$$

where $\boldsymbol{\kappa}_{n+1}$ is the column vector with i -th entry $\kappa(\mathbf{x}_i, \mathbf{x}_{n+1})$, $i \in \{1, 2, \dots, n\}$, and $\kappa_{n+1} = \kappa(\mathbf{x}_{n+1}, \mathbf{x}_{n+1})$. Based on Equations (15) and (17), we arrive at an online implementation of LS-OC-SVM.

3.4. Sparse Online Least Squares One-Class SVM

The procedures for calculating the parameters, α and ρ , of LS-OC-SVM in Section 3.3 lose sparseness, due to the quadratic loss function in the objective function Equation (4). This formulation is inappropriate for large-scale data and unsuitable for online learning, as the number of training samples grows infinitely [25]. We propose a sparse solution to provide a robust formulation. A dictionary is adopted to address the sparse approximation problem [40].

Instead of Equation (6), where \mathbf{w} is expressed with all available data, we intend to approximate it by adopting a dictionary in a sparse way. Consider a dictionary, $\mathbf{x}_{\mathcal{D}}$, $\mathcal{D} \subset \{1, 2, \dots, n\}$, of size D with elements \mathbf{x}_{w_j} , $j \in \mathcal{D}$. Instead of Equation (6), we approximate \mathbf{w} with these D dictionary elements:

$$\mathbf{w} = \sum_{j=1}^D \beta_j \Phi(\mathbf{x}_{w_j}) \quad (20)$$

The hyperplane becomes:

$$f(\mathbf{x}) = \sum_{j=1}^D \beta_j \kappa(\mathbf{x}, \mathbf{x}_{w_j}) - \rho = 0 \quad (21)$$

In sparse online LS-OC-SVM, the distance, $dis_{\mathcal{D}}(\mathbf{x})$, of a datum, \mathbf{x} , to the hyperplane is:

$$dis_{\mathcal{D}}(\mathbf{x}) = \frac{|\sum_{j=1}^D \beta_j \kappa(\mathbf{x}, \mathbf{x}_{w_j}) - \rho|}{\|\boldsymbol{\beta}\|} \quad (22)$$

where \mathbf{x}_{w_j} is a dictionary element and $\boldsymbol{\beta}$ is the column vector with the entries, β_j . Replacing Expression (20) into Lagrange Function (5), we have:

$$L = \frac{1}{2} \boldsymbol{\beta}^\top K_{\mathcal{D}} \boldsymbol{\beta} - \rho + \frac{C}{2} \sum_{i=1}^n \xi_i^2 - \sum_{i=1}^n \alpha_i \left(\sum_{j=1}^D \beta_j \Phi^\top(\mathbf{x}_{w_j}) \Phi(\mathbf{x}_i) + \xi_i - \rho \right) \quad (23)$$

Taking the derivatives of the Function (23) with respect to primal variables, $\boldsymbol{\beta}$, ξ_i , ρ and α_i , yields:

$$\frac{\partial L}{\partial \boldsymbol{\beta}} = 0 \quad \Rightarrow K_D \boldsymbol{\beta} = K_D^\top(\mathbf{x}) \boldsymbol{\alpha} \quad (24)$$

$$\frac{\partial L}{\partial \xi_i} = 0 \quad \Rightarrow C \xi_i = \alpha_i \quad (25)$$

$$\frac{\partial L}{\partial \rho} = 0 \quad \Rightarrow \sum_{i=1}^n \alpha_i = 1 \quad (26)$$

$$\frac{\partial L}{\partial \alpha_i} = 0 \quad \Rightarrow \sum_{j=1}^D \beta_j \kappa(\mathbf{x}_{w_j}, \mathbf{x}_i) + \xi_i - \rho = 0 \quad (27)$$

The matrix form for Condition (27) is written:

$$\mathbf{K}_{\mathcal{D}}(\mathbf{x}) \boldsymbol{\beta} + \boldsymbol{\xi} - \rho = 0 \quad (28)$$

Replacing Conditions (24) and (25) into (28) leads to:

$$\mathbf{K}_{\mathcal{D}}(\mathbf{x}) \mathbf{K}_{\mathcal{D}}^{-1} \mathbf{K}_{\mathcal{D}}^\top(\mathbf{x}) \boldsymbol{\alpha} + \frac{\boldsymbol{\alpha}}{C} - \rho = 0 \quad (29)$$

Combining Equations (26) and (29), the equation for computing coefficients $\left[\boldsymbol{\alpha} \quad -\rho \right]^\top$ becomes:

$$\begin{bmatrix} \mathbf{K}_{\mathcal{D}}(\mathbf{x}) \mathbf{K}_{\mathcal{D}}^{-1} \mathbf{K}_{\mathcal{D}}^\top(\mathbf{x}) + \frac{I}{C} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ -\rho \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 1 \end{bmatrix} \quad (30)$$

After providing these relations with the dictionary, we now discuss the dictionary construction. The coherence criterion is adopted to characterize a dictionary in sparse approximation problems. It provides an elegant model reduction criterion with a less computationally-demanding procedure [25,40,41]. The coherence of a dictionary is defined as the largest correlation between the elements in the dictionary, *i.e.*,

$$\mu = \max_{i,j \in \mathcal{D}, i \neq j} |\kappa(\mathbf{x}_i, \mathbf{x}_j)| \quad (31)$$

In the online case, the coherence between a new datum and the current dictionary is calculated by:

$$\epsilon_t = \max_{j \in \mathcal{D}} |\kappa(\mathbf{x}_t, \mathbf{x}_{w_j})| \quad (32)$$

where \mathbf{x}_{w_j} is the element in the dictionary, $\mathbf{x}_{\mathcal{D}}$. Presetting a threshold, μ_0 , the new arrival sample, \mathbf{x}_t , at the time step, t , is tested with the coherence criterion to judge whether the dictionary remains unchanged or is incremented by including the new element. For n training samples, the subset, which includes m ($1 \leq m \ll n$) samples, is considered the initial dictionary. Then, each remaining sample is tested with Equation (32) to determine the relation between itself and the previous dictionary. If $\epsilon_t \leq \mu_0$, it will be included into the dictionary. Concretely, the algorithm is performed with two cases described herein below.

First case: $\epsilon_t > \mu_0$

In this case, at time step $n + 1$, the new data, \mathbf{x}_{n+1} , is not included into the dictionary. The Gram matrix, $\mathbf{K}_{\mathcal{D}}$, with the entries, $\kappa(\mathbf{x}_i, \mathbf{x}_j)$, $i, j \in \{1, 2, \dots, D\}$, is unchanged. When a new sample, \mathbf{x} , arrives, we need to compute:

$$\left[\begin{bmatrix} \mathbf{K}_{\mathcal{D}}(\mathbf{x}) \\ \boldsymbol{\kappa}^\top \end{bmatrix} \mathbf{K}_{\mathcal{D}}^{-1} \begin{bmatrix} \mathbf{K}_{\mathcal{D}}(\mathbf{x})^\top & \boldsymbol{\kappa} \end{bmatrix} + \frac{\mathbf{I}}{C} \right]^{-1} = \begin{bmatrix} \mathbf{K}_{\mathcal{D}}(\mathbf{x})\mathbf{K}_{\mathcal{D}}^{-1}\mathbf{K}_{\mathcal{D}}^\top(\mathbf{x}) + \frac{\mathbf{I}}{C} & \mathbf{K}_{\mathcal{D}}(\mathbf{x})\mathbf{K}_{\mathcal{D}}^{-1}\boldsymbol{\kappa} \\ \boldsymbol{\kappa}^\top\mathbf{K}_{\mathcal{D}}^{-1}\mathbf{K}_{\mathcal{D}}^\top(\mathbf{x}) & \boldsymbol{\kappa}^\top\mathbf{K}_{\mathcal{D}}^{-1}\boldsymbol{\kappa} + \frac{\mathbf{I}}{C} \end{bmatrix}^{-1} \quad (33)$$

where at time step $n + 1$, $\boldsymbol{\kappa}$ is the column vector with entries $\kappa(\mathbf{x}_{n+1}, \mathbf{x}_{w_j})$, $j \in \{1, 2, \dots, D\}$. $\mathbf{K}_{\mathcal{D}}(\mathbf{x})$ is the matrix with the (i, j) -th entry $\kappa(\mathbf{x}_i, \mathbf{x}_{w_j})$, $i \in \{1, 2, \dots, n\}$, $j \in \{1, 2, \dots, D\}$.

Second case: $\epsilon_t \leq \mu_0$

In this case, the new data, \mathbf{x}_{n+1} , is added into the dictionary, $\mathbf{x}_{\mathcal{D}}$. Then, the Gram matrix should be changed by:

$$\overline{\mathbf{K}}_{\mathcal{D}} = \begin{bmatrix} \mathbf{K}_{\mathcal{D}} & \mathbf{d} \\ \mathbf{d}^\top & d \end{bmatrix} \quad (34)$$

where $\overline{\mathbf{K}}_{\mathcal{D}}$ is the Gram matrix of the dictionary, including the new arrival dictionary sample, \mathbf{x}_{n+1} , and $\mathbf{K}_{\mathcal{D}}$ is the Gram matrix of the dictionary at the last time step, n . Let $\mathbf{x}_{\mathcal{D}} = \{\mathbf{x}_{w_1}, \mathbf{x}_{w_2}, \dots, \mathbf{x}_{w_D}\}$ denote the dictionary at time step n ; \mathbf{d} is the column vector with entries $d_j = \kappa(\mathbf{x}, \mathbf{x}_{w_j})$, $j \in \{1, 2, \dots, D\}$, and $d = \kappa(\mathbf{x}_{n+1}, \mathbf{x}_{n+1})$. By adopting the matrix inverse identity Equation (16), we have:

$$\overline{\mathbf{K}}_{\mathcal{D}}^{-1} = \begin{bmatrix} \mathbf{K}_{\mathcal{D}}^{-1} + \mathbf{A} & \mathbf{b} \\ \mathbf{b}^\top & c \end{bmatrix} \quad (35)$$

where:

$$c = \frac{1}{d - \mathbf{d}^\top \mathbf{K}_{\mathcal{D}}^{-1} \mathbf{d}} \quad (36)$$

$$\mathbf{A} = c \mathbf{K}_{\mathcal{D}}^{-1} \mathbf{d} \mathbf{d}^\top \mathbf{K}_{\mathcal{D}}^{-1} \quad (37)$$

$$\mathbf{b} = -c \mathbf{K}_{\mathcal{D}}^{-1} \mathbf{d} \quad (38)$$

Because the dictionary changes, the value of $\mathbf{K}_{\mathcal{D}}(\mathbf{x})$ and also $[\mathbf{K}_{\mathcal{D}}(\mathbf{x})\mathbf{K}_{\mathcal{D}}^{-1}\mathbf{K}_{\mathcal{D}}^{\top}(\mathbf{x}) + \frac{\mathbf{I}}{C}]^{-1}$ should be updated. Let the S denote the updated $[\mathbf{K}_{\mathcal{D}}(\mathbf{x})\mathbf{K}_{\mathcal{D}}^{-1}\mathbf{K}_{\mathcal{D}}^{\top}(\mathbf{x}) + \frac{\mathbf{I}}{C}]^{-1}$ at time step $n + 1$; we have:

$$S = \left[\begin{bmatrix} \mathbf{K}_{\mathcal{D}}(\mathbf{x}) & \mathbf{q} \end{bmatrix} \overline{\mathbf{K}}^{-1} \begin{bmatrix} \mathbf{K}_{\mathcal{D}}^{\top}(\mathbf{x}) \\ \mathbf{q}^{\top} \end{bmatrix} + \frac{\mathbf{I}}{C} \right]^{-1} \quad (39)$$

$$= [\mathbf{K}_{\mathcal{D}}(\mathbf{x})\mathbf{K}_{\mathcal{D}}^{-1}\mathbf{K}_{\mathcal{D}}(\mathbf{x})^{\top} + \frac{\mathbf{I}}{C} + \mathbf{K}_{\mathcal{D}}(\mathbf{x})\mathbf{A}\mathbf{K}_{\mathcal{D}}^{\top}(\mathbf{x}) + \mathbf{q}\mathbf{b}^{\top}\mathbf{K}_{\mathcal{D}}^{\top}(\mathbf{x}) + \mathbf{K}_{\mathcal{D}}(\mathbf{x})\mathbf{b}\mathbf{q}^{\top} + c\mathbf{q}\mathbf{q}^{\top}]^{-1} \quad (40)$$

where at time step $n + 1$, \mathbf{q} is the column vector with entries $q_i = \kappa(\mathbf{x}_i, \mathbf{x}_{D+1})$, $i \in \{1, 2, \dots, n\}$, and \mathbf{x}_{D+1} is the new arrival datum \mathbf{x}_{n+1} , which is included into the dictionary. The matrix inverse in Equation (39) can be calculated by using four-times Woodbury identity:

$$(A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1} \quad (41)$$

with proper choices of matrices A , U , C and V , such that U and V should be chosen as two vectors, and A should be chosen as a scalar. Thus, the inverse, $(C^{-1} + VA^{-1}U)$, is a scalar; Equation (39) can be calculated very efficiently. For instance, for computing the inverse, including the term, $(\mathbf{K}_{\mathcal{D}}(\mathbf{x})\mathbf{b}\mathbf{q}^{\top})$, we regard two vectors, $(\mathbf{K}_{\mathcal{D}}(\mathbf{x})\mathbf{b})$ and \mathbf{q}^{\top} , as vector U and V , respectively, while C in Equation (41) is one.

Once knowing S , using Equation (33) to add the new κ with entries $\kappa(\mathbf{x}_{n+1}, \mathbf{x}_{w_j})$, $j \in \{1, 2, \dots, D, D + 1\}$, \mathbf{x}_{w_j} is an element of the dictionary.

4. Covariance Descriptor of Frame Behavior

The optical flow is chosen as the basic low-level feature to represent the movement direction and amplitude. We apply the Horn–Schunck (HS) method to compute optical flow in this paper. The optical flow can provide important information about the spatial arrangement of the object and the change rate of this arrangement [42]. The optical flow of a gray image is formulated as the minimization of the following global energy functional:

$$E = \int \int [(I_x u + I_y v + I_t)^2 + \gamma^2(\|\nabla u\|^2 + \|\nabla v\|^2)] dx dy \quad (42)$$

where I is the intensity of the image, I_x, I_y and I_t are the derivatives of the image intensity value along the x , y and time t dimension, u and v are the components of the optical flow in horizontal and vertical direction and γ represents the weight of the regularization term.

The covariance feature descriptor was originally proposed by Tuzel [43] for pattern matching in a target tracking problem. Owing to its good performance, the covariance descriptor encoding the optical flow is introduced to represent the global movement of the frame. A feature is defined as:

$$F(x, y, i) = \phi_i(I, x, y) \quad (43)$$

where I is the image (which could be gray, red-green-blue (RGB), hue-saturation-value (HSV), hue-lightness-saturation (HLS), etc), ϕ_i is a mapping relating the image with the i -th feature, F

is the $W \times H \times d$ dimension feature, W is the image width, H is the image height and d is the number of the chosen features. For each frame, the feature, F , can be represented as the $d \times d$ covariance matrix:

$$C = \frac{1}{n-1} \sum_{k=1}^n (z_k - \mu)(z_k - \mu)^T \tag{44}$$

where n is the number of the pixels sampled in the frame, μ is the mean of n feature vectors of the selected points and z_k is the feature vector of the k -th point. C is the covariance matrix of the feature vector, F . The covariance matrix descriptor proposes a way to merge multiple features. Different choices of feature vectors are shown in Table 1, where u and v are horizontal and vertical components of optical flow, u_x and v_x are the first derivatives of horizontal and vertical optical flow in the x direction, respectively, u_y and v_y are the first derivatives of the corresponding feature in the y direction, u_{xx} and v_{xx} are the second derivatives in x direction and u_{yy} and v_{yy} are the second derivatives in y direction. The flowchart of covariance matrix descriptor computation is shown in Figure 2. The optical flow and corresponding partial derivative characterize the inter-frame information or can be regarded as the movement information.

Figure 2. Covariance descriptor computation based on the features of a video frame.

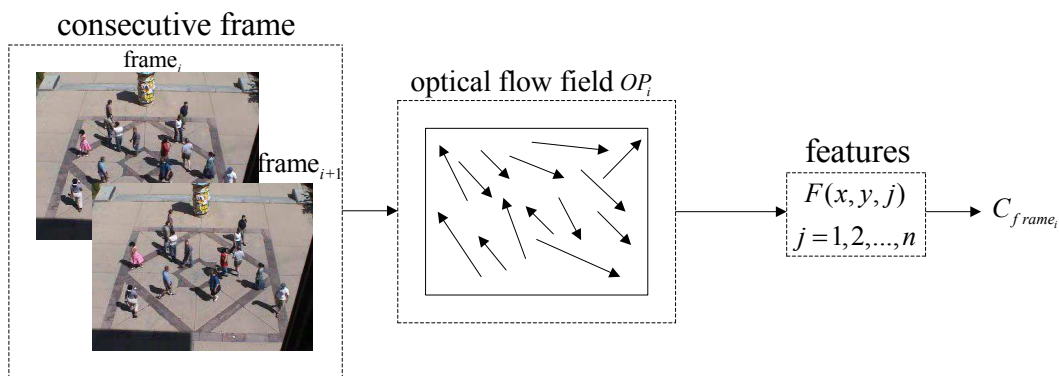


Table 1. Different choices of feature F to construct the covariance descriptor.

Feature Vector F	
$F_1(6 \times 6)$	$[y \ x \ u \ v \ u_x \ u_y]$
$F_2(6 \times 6)$	$[y \ x \ u \ v \ v_x \ v_y]$
$F_3(8 \times 8)$	$[y \ x \ u \ v \ u_x \ u_y \ v_x \ v_y]$
$F_4(12 \times 12)$	$[y \ x \ u \ v \ u_x \ u_y \ v_x \ v_y \ u_{xx} \ u_{yy} \ v_{xx} \ v_{yy}]$

If proper parameters are given, classical kernels, such as Gaussian, polynomial and sigmoidal kernels, have similar performances [44]. In our work, the Gaussian kernel $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})$ is used. The covariance matrix is an element in the Lie group; the Gaussian kernel in Euclidean spaces is not suitable. The Gaussian kernel in the Lie group is defined as [45,46]:

$$\kappa(\mathbf{X}_i, \mathbf{X}_j) = \exp(-\frac{\|\log(\mathbf{X}_i^{-1}\mathbf{X}_j)\|}{2\sigma^2}), (\mathbf{X}_i, \mathbf{X}_j) \in G \times G \tag{45}$$

where \mathbf{X}_i and \mathbf{X}_j are matrices in Lie group G ; the parameter σ determines the scale at which the data is probed.

5. Abnormal Event Detection

In an abnormal event detection problem, it is assumed that a set of training frames, $\{I_1, I_2, \dots, I_n\}$ (the positive class), describing the normal behavior is obtained. The abnormal detection strategies relative to the online algorithms proposed in Section 3.3 and Section 3.4 are introduced below.

5.1. Online LS-OC-SVM Strategy

The general architecture of the abnormal event detection method via online least squares one-class SVM (online LS-OC-SVM) proposed in Section 3.3 is summarized in Algorithm 1; the flowchart is shown in Figure 3 and explained below.

Algorithm 1: Visual abnormal event detection via online least squares one-class support vector machine (LS-OC-SVM) and sparse online LS-OC-SVM.

Require

n training frames $\{I_i\}_{i=1}^n$ and the corresponding optical flow $\{OP_i\}_{i=1}^n$.
Compute the covariance matrix of each frame.

$$\{OP_1, OP_2, \dots, OP_n\} \longrightarrow \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_n\} \quad (46)$$

(a) *Online strategy:* Applying LS-OC-SVM on the small subset of training samples to calculate the coefficient matrix.

$$\{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_m\}, 1 \leq m \ll n \xrightarrow{\text{online}} \text{coefficient matrix } [\mathbf{K}] \begin{bmatrix} \boldsymbol{\alpha} & -\rho \end{bmatrix}^\top \quad (47)$$

(b) *Sparse online strategy:* Applying LS-OC-SVM to train the initial dictionary, $\mathcal{C}_{\mathcal{D}}$, offline.

$$\mathcal{C}_{\mathcal{D}} = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_m\}, 1 \leq m \ll n \xrightarrow{\text{offline}} \text{coefficient matrix } [\mathbf{K}] \begin{bmatrix} \boldsymbol{\beta} & -\rho \end{bmatrix}^\top \quad (48)$$

(a) *Online strategy:* Applying online LS-OC-SVM on the remaining samples to calculate the coefficient matrix.

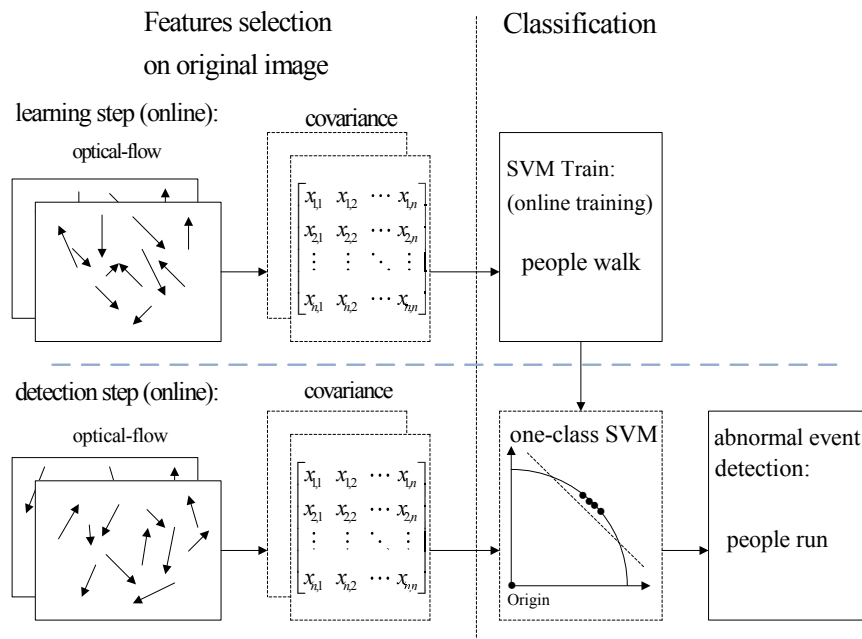
$$\{\mathbf{C}_{m+1}, \mathbf{C}_{m+2}, \dots, \mathbf{C}_n\}, [\mathbf{K}] \xrightarrow{\text{online}} \text{coefficient matrix } [\mathbf{K}] \begin{bmatrix} \boldsymbol{\alpha} & -\rho \end{bmatrix}^\top \quad (49)$$

(b) *Sparse online strategy:* Applying sparse online LS-OC-SVM on the remaining samples to calculate the coefficient matrix and to update the dictionary.

$$\begin{cases} \{\mathcal{C}_{\mathcal{D}}, \mathbf{C}_k\}, m < k \leq n \xrightarrow{\text{sparse online}} \text{coefficient matrix } \begin{bmatrix} \boldsymbol{\beta} & -\rho \end{bmatrix}^\top \\ \mathcal{C}_{\mathcal{D}} := \mathcal{C}_{\mathcal{D}} \cup \mathbf{C}_k, & \text{if } \epsilon_t \geq \mu_0 \\ \mathcal{C}_{\mathcal{D}} := \mathcal{C}_{\mathcal{D}}, & \text{if } \epsilon_t < \mu_0 \end{cases} \quad (50)$$

Each frame \mathbf{C}_{n+l} is classified via LS-OC-SVM.

Figure 3. Major processing states of the proposed abnormal frame event detection method. The covariance of the *frame* is computed.



Step 1: The first step consists of calculating the covariance matrix descriptor of the training frames. The features could be chosen as any form shown in Table 1. This step can be generalized as:

$$\{OP_1, OP_2, \dots, OP_n\} \longrightarrow \{C_1, C_2, \dots, C_n\} \tag{51}$$

where $\{OP_1, OP_2, \dots, OP_n\}$ are the image optical flows of the 1st to $n - th$ frames; $\{C_1, C_2, \dots, C_n\}$ are the covariance matrix descriptors.

Step 2: The second step is applying LS-OC-SVM on a small subset of the training samples to calculate the coefficient parameters, α and ρ , in Equation (11). Consider a subset $\{C_i\}_{i=1}^m, 1 \leq m \ll n$ of data selected from the training set $\{C_i\}_{i=1}^n$. Without loss of generality, assume that the first m frames are chosen. These m samples are trained offline. This step can be described in the following equation:

$$\{C_1, C_2 \dots C_m\}, 1 \leq m \ll n \xrightarrow{\text{offline}} \text{coefficient matrix } [K] \quad [\alpha \quad -\rho]^T \tag{52}$$

where $[K]$ and $[\alpha \quad -\rho]^T$ are defined in Equation (11).

Step 3: After learning the first m samples, the coefficient matrices, K and $[\alpha \quad -\rho]^T$, are obtained. The online LS-OC-SVM method (Section 3.3) is applied to learn the remaining $n - m$ samples $\{C_{m+1}, C_{m+2} \dots C_n\}$. This step can be expressed as:

$$\{C_{m+1}, C_{m+2} \dots C_n\}, [K] \xrightarrow{\text{online}} \text{coefficient matrix } [K] \quad [\alpha \quad -\rho]^T \tag{53}$$

Step 4: Based on the coefficient matrix, $[\alpha \quad -\rho]^T$, the distance of the training samples $\{C_i\}_{i=1}^n$ and the incoming test sample, C_{n+l} , with respect to the decision plane is computed. By comparing the distances of the samples, an abnormal event is detected:

$$dis(\mathbf{C}_{n+l}) = \frac{|(\sum_{i=1}^n \alpha_i \kappa(\mathbf{C}, \mathbf{C}_i) - \rho)|}{\|\boldsymbol{\alpha}\|} \quad (54)$$

$$= \begin{cases} 1 & \text{if } f(\mathbf{C}_{n+l}) \geq T_{dis} \\ -1 & \text{if } f(\mathbf{C}_{n+l}) < T_{dis} \end{cases} \quad (55)$$

where \mathbf{C}_{n+l} is the covariance matrix descriptor of the $(n + l) - th$ frame needed to be classified, and \mathbf{C}_i is the sample of the training data. “1” corresponds to an abnormal frame; “-1” corresponds to a normal frame. T_{dis} is the threshold of the distance, it is the maximum distance of the training samples to the hyperplane.

5.2. Sparse Online LS-OC-SVM Strategy

The abnormal event detection via sparse online least squares one-class SVM (sparse online LS-OC-SVM) is introduced below. A subset of the samples is chosen to form the dictionary, $\mathbf{C}_{\mathcal{D}}$, making a sparse representation of the training data. The initial dictionary, $\mathbf{C}_{\mathcal{D}}$, is learned offline. Each remaining training sample is learned one-by-one online. Meanwhile, it is checked to be included, or not, into the dictionary. The test datum is classified based on the dictionary. The feature extraction step (Step 1) and the detection step (Step 4) are the same as the ones presented in Section 5.1. Owing to the dictionary, the training steps are different.

Step 2-sparse: The second step is applying LS-OC-SVM to train the initial dictionary offline. The first m samples are the initial dictionary denoted as $\mathbf{C}_{\mathcal{D}}$. This step can be generalized as:

$$\mathbf{C}_{\mathcal{D}} = \{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_m\}, 1 \leq m \ll n \xrightarrow{\text{offline}} \text{coefficient matrix } \begin{bmatrix} \mathbf{K} \\ \boldsymbol{\beta} - \rho \end{bmatrix}^{\top} \quad (56)$$

Step 3-sparse: After learning the initial dictionary, $\mathbf{C}_{\mathcal{D}}$, including the first m ($1 \leq m \ll n$) samples, the remaining training samples, $\{\mathbf{C}_{m+1}, \mathbf{C}_{m+2}, \dots, \mathbf{C}_n\}$, are learned via sparse online LS-OC-SVM described in Section 3.4. This step can be described in the following equations:

$$\begin{aligned} & \{\mathbf{C}_{\mathcal{D}}, \mathbf{C}_k\}, m < k \leq n \xrightarrow{\text{sparse online}} \\ & \text{coefficient matrix } \begin{bmatrix} \boldsymbol{\beta} - \rho \end{bmatrix}^{\top} \\ & \begin{cases} \mathbf{C}_{\mathcal{D}} := \mathbf{C}_{\mathcal{D}} \cup \mathbf{C}_k & \text{if } \epsilon_t \geq \mu_0 \\ \mathbf{C}_{\mathcal{D}} := \mathbf{C}_{\mathcal{D}} & \text{if } \epsilon_t < \mu_0 \end{cases} \end{aligned} \quad (57)$$

where $\mathbf{C}_{\mathcal{D}}$ is the dictionary and \mathbf{C}_k is a new incoming remaining sample in the training dataset. According to the coherence criterion introduced in Section 3.4, if the new sample, \mathbf{C}_k , satisfies the dictionary updated condition, it will be included into the dictionary, $\mathbf{C}_{\mathcal{D}}$.

6. Abnormal Event Detection Results

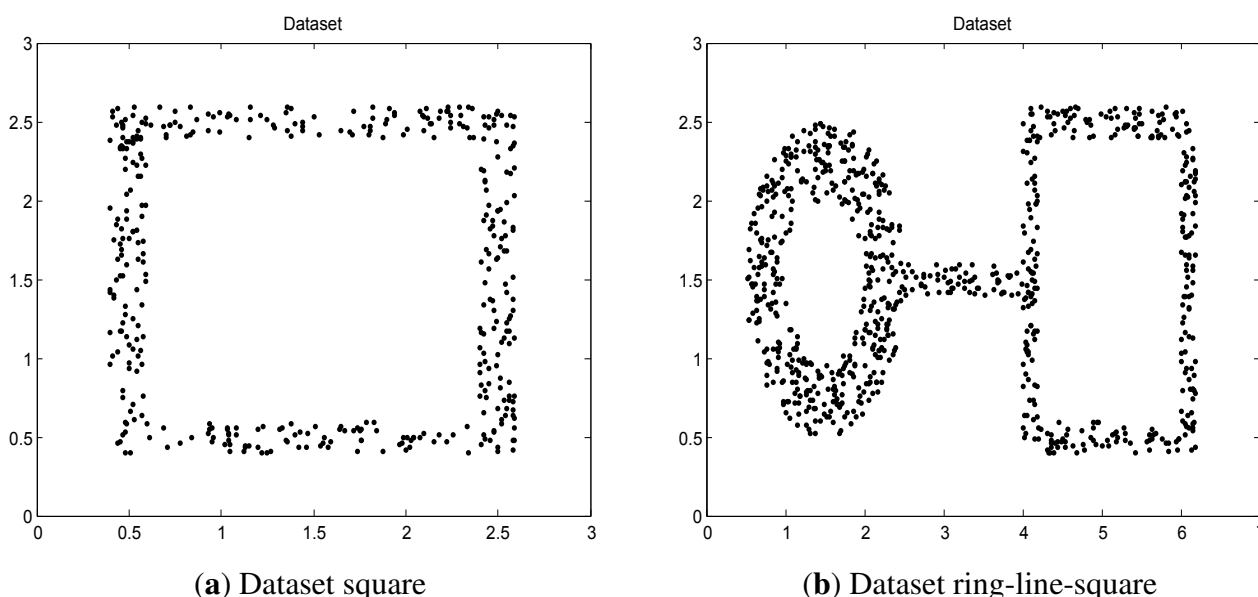
This section presents the results of experiments conducted to illustrate the performance of the two proposed classification algorithms, online least square one-class SVM (online LS-OC-SVM) and

sparse online least square one-class SVM (sparse online LS-OC-SVM). The two-dimensional synthetic distribution dataset and the University of Minnesota (UMN) [47] dataset are used.

6.1. Synthetic Dataset via Online LS-OC-SVM and Sparse Online LS-OC-SVM

Two synthetic data, “square” and “ring-line-square” [48], are used. The “square” consists of four lines, 2.2 in length and 0.2 in width. In the area of these lines, 400 points were randomly dispersed with a uniform distribution. The “ring-line-square” distribution is composed of three parts: a ring with an inner diameter of 1.0 and an outer diameter of 2.0, a line of 1.6 in length and 0.2 in width, and a square the same as dataset “square” introduced above. 850 points are randomly dispersed with a uniform distribution. These two data are shown in Figure 4.

Figure 4. Synthetic dataset. (a) Square; (b) ring-line-square.



The first sample is used for initializing the online LS-OC-SVM proposed in Section 3.3; the 399 remaining samples in “square” and 849 remaining samples in “ring-line-square” are learned in the online manner.

Via the sparse online LS-OC-SVM method proposed in Section 3.4, the first sample is trained offline, and this sample is considered the initial dictionary. Then, each arrival sample in 399 remaining samples in “square” and 849 remaining samples in “ring-line-square” are checked by the coherence criterion to determine whether the dictionary should be retained or updated by including the new element.

The distances are shown in contours illustrating the boundary. The contours of “square” and “ring-line-square” are shown in Figures 5 and 6, respectively. Gaussian kernel was used in these two data, with bandwidth $\sigma = 0.065$. The preset threshold of the coherence criterion is $\mu_0 = 0.08$. The detection results obtained by these two online training algorithms are the same as the ones when training data were learned in a batch model.

Figure 5. Offline, online LS-OC-SVM and sparse online LS-OC-SVM results of “square”. The figures might be viewed better electronically, in color and enlarged. **(a)** The contours when all the training data are learned as one batch offline; **(b)** The contours when the training data are learned via online LS-OC-SVM; **(c)** The blue circle (pointed out by the arrow) shows the original dictionary. The red points show the 232 new data included into the dictionary via sparse online LS-OC-SVM; **(d)** The contours when the training data are learned via sparse online LS-OC-SVM.

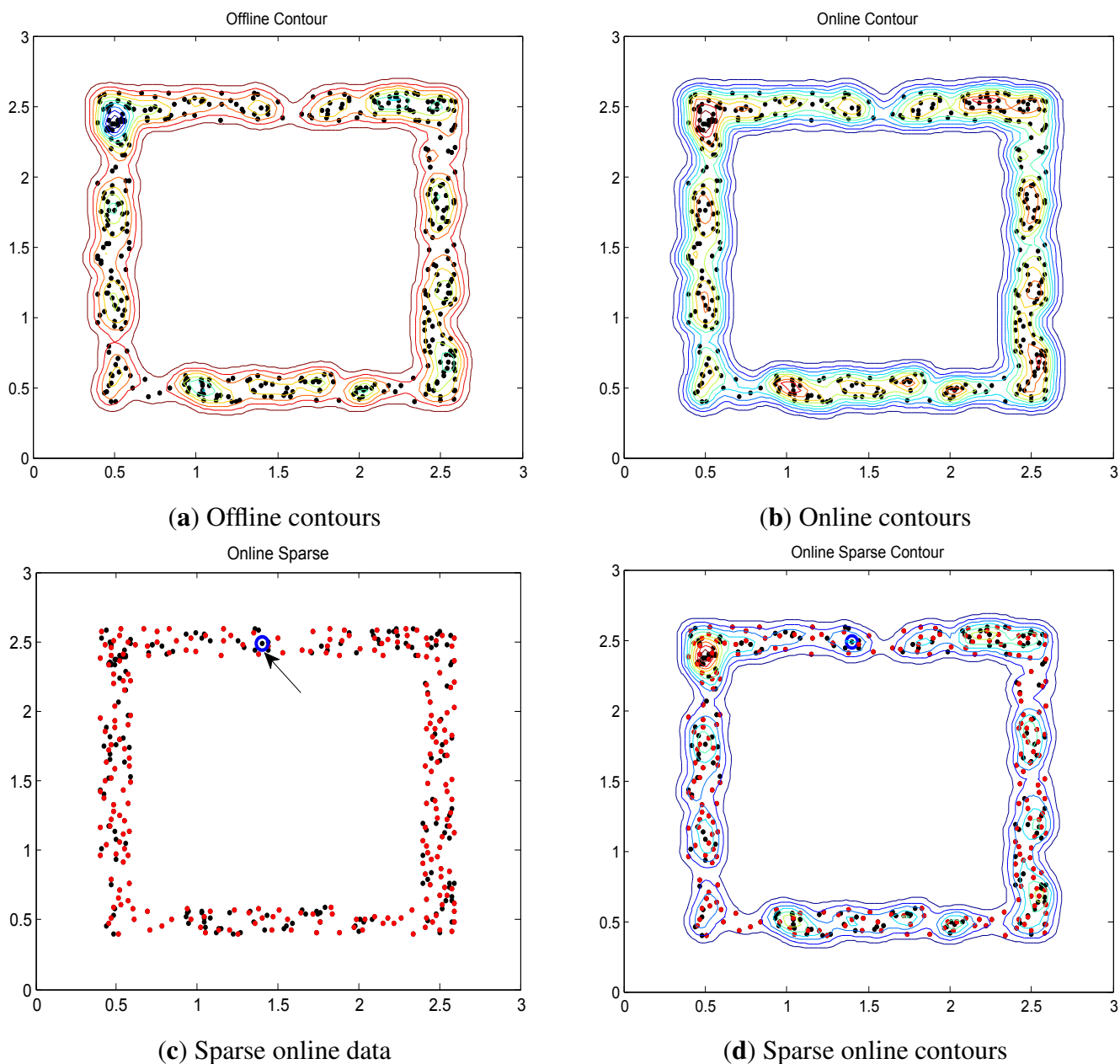
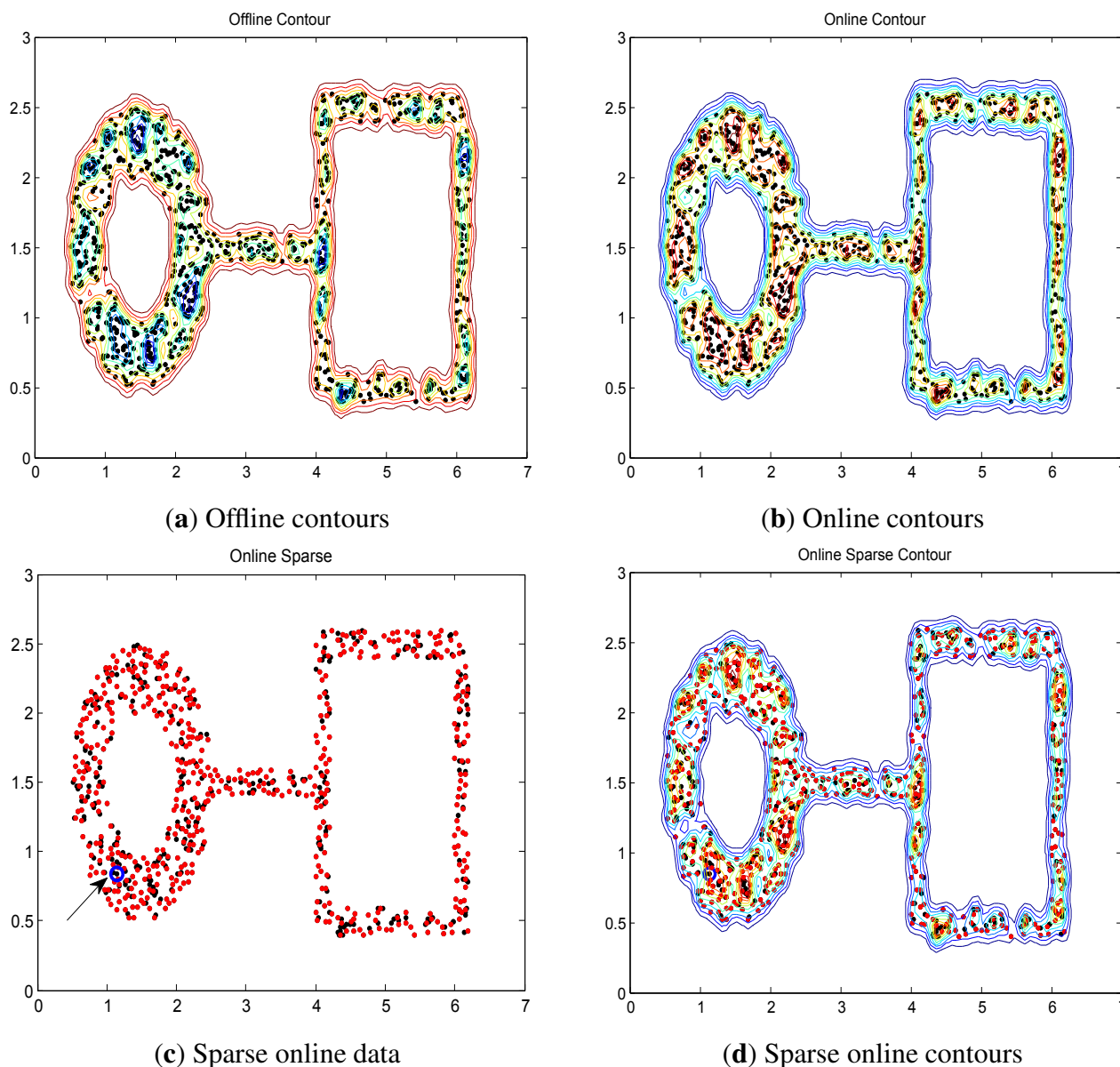


Figure 6. Offline, online LS-OC-SVM and sparse online LS-OC-SVM results of “ring-line-square”. (a) The contours when all the training data are learned as one batch offline; (b) The contours when the training data are learned via online LS-OC-SVM; (c) The blue circle (pointed out by the arrow) shows the original dictionary. The red points show the 534 new data included into the dictionary via sparse online LS-OC-SVM; (d) The contours when the training data are learned via sparse online LS-OC-SVM.



6.2. Abnormal Visual Event Detection via Online LS-OC-SVM

UMN dataset detection results via online LS-OC-SVM proposed in Section 3.3 are shown below. The UMN dataset consists of eleven sequences of crowded panic escape events, which are recorded in a lawn, an indoor and a plaza scene. A frame where the people are walking in different directions is considered as a normal sample for training or for normal testing. A scene where the people are running is taken as an abnormal sample for testing. The detection results of the lawn scene, the indoor scene and the plaza scene

are shown in Figures 7, Figure 8 and Figure 9, respectively. A Gaussian kernel for the covariance matrix in the Lie group is used. Various values of the variance, σ , in the Gaussian function and the penalty factor, C , are chosen to form the receiver operating characteristic (ROC) curve. In the indoor scene, time lags of the frame labels lead to the lower area under the ROC curve (AUC) value. In the last few frames, labeled as abnormal of abnormal sequences, there are no people, while, in the training samples, there are no people in the upper half of the image. The covariance of the training frame is similar to the covariance of the abnormal frame without people. Our covariance feature descriptor-based classification method cannot distinguish between these two situations. However, this issue can be resolved by utilizing the foreground information. For example, if there are no moving objects in the frame, this frame is immediately classified as abnormal. The results of these three scenes show that the covariance descriptor can distinguish between normal and abnormal events. The performance of online LS-OC-SVM is almost the same as that of the offline method.

Figure 7. Detection results of the lawn scene. (a) The detection result of a normal frame; (b) The detection result of an abnormal panic frame; (c) Receiver operating characteristic (ROC) curve of covariance descriptors constructed from different features F of the lawn scene detection results via LS-OC-SVM. All the training samples are learned together offline. The biggest AUC value is 0.9874; (d) ROC curve of detection results via online LS-OC-SVM. The biggest AUC value is 0.9874.

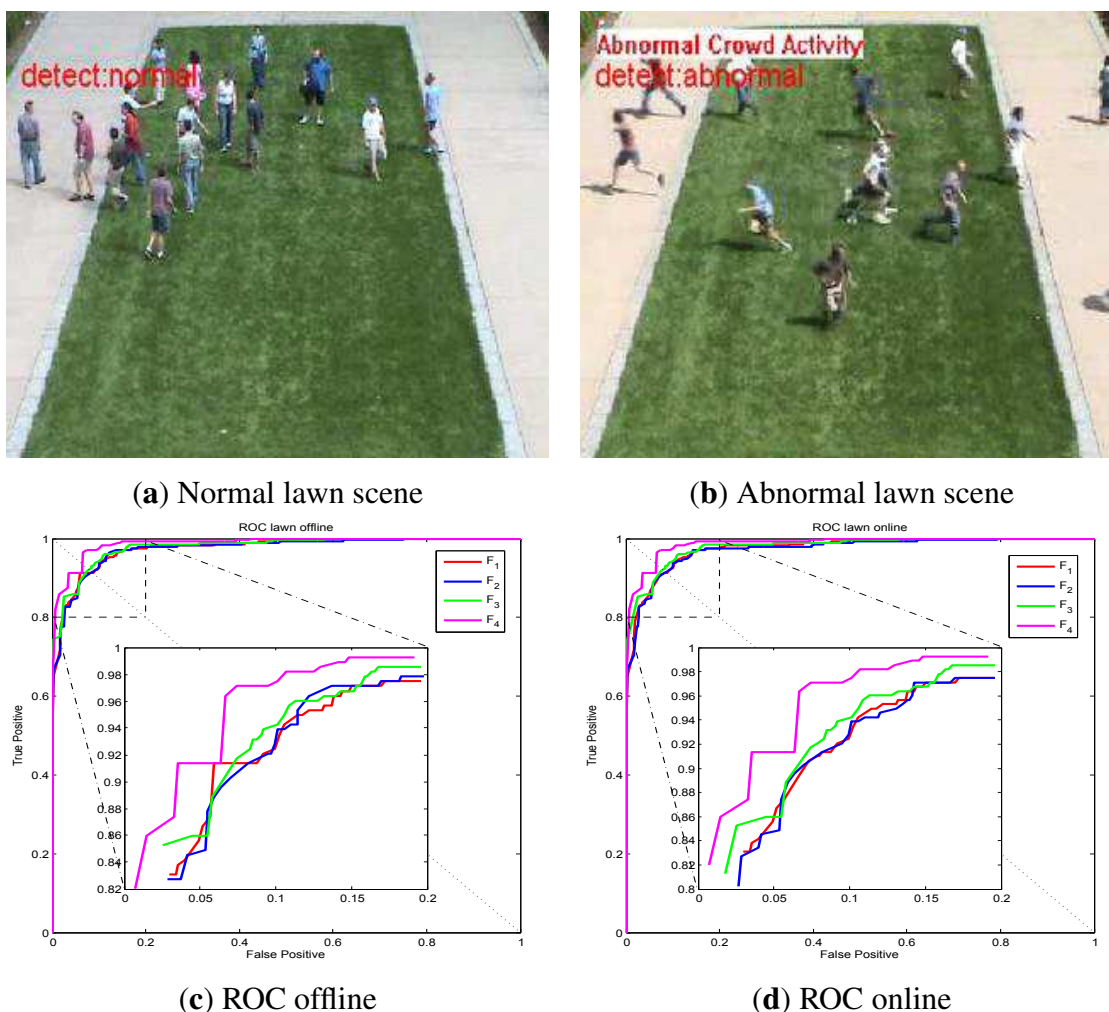


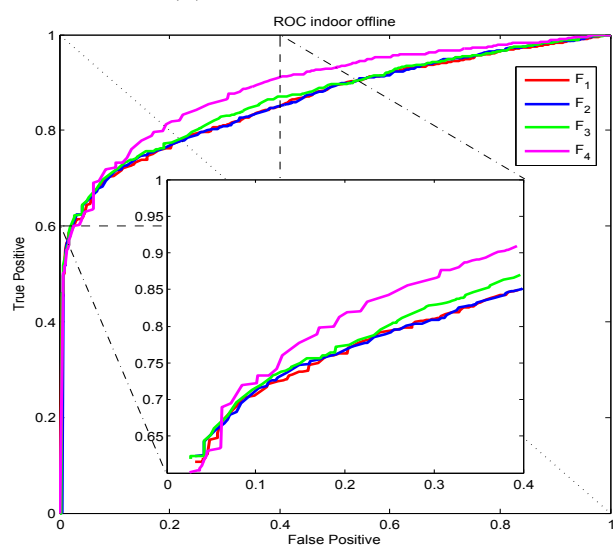
Figure 8. Detection results of the indoor scene. (a) The detection result of a normal frame; (b) The detection result of an abnormal panic frame; (c) ROC curve of covariance descriptors constructed from different features F of the indoor scene results via LS-OC-SVM. All the training samples are learned together offline. The biggest AUC value is 0.8900; (d) ROC curve of detection results via online LS-OC-SVM. The biggest AUC value is 0.8904.



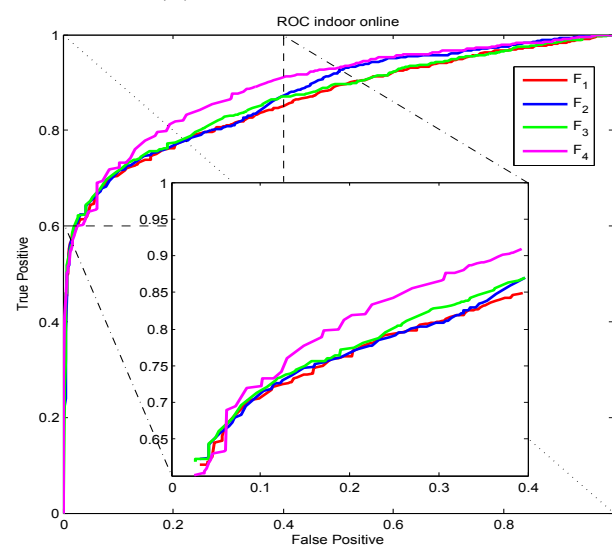
(a) Normal indoor scene



(b) Abnormal indoor scene

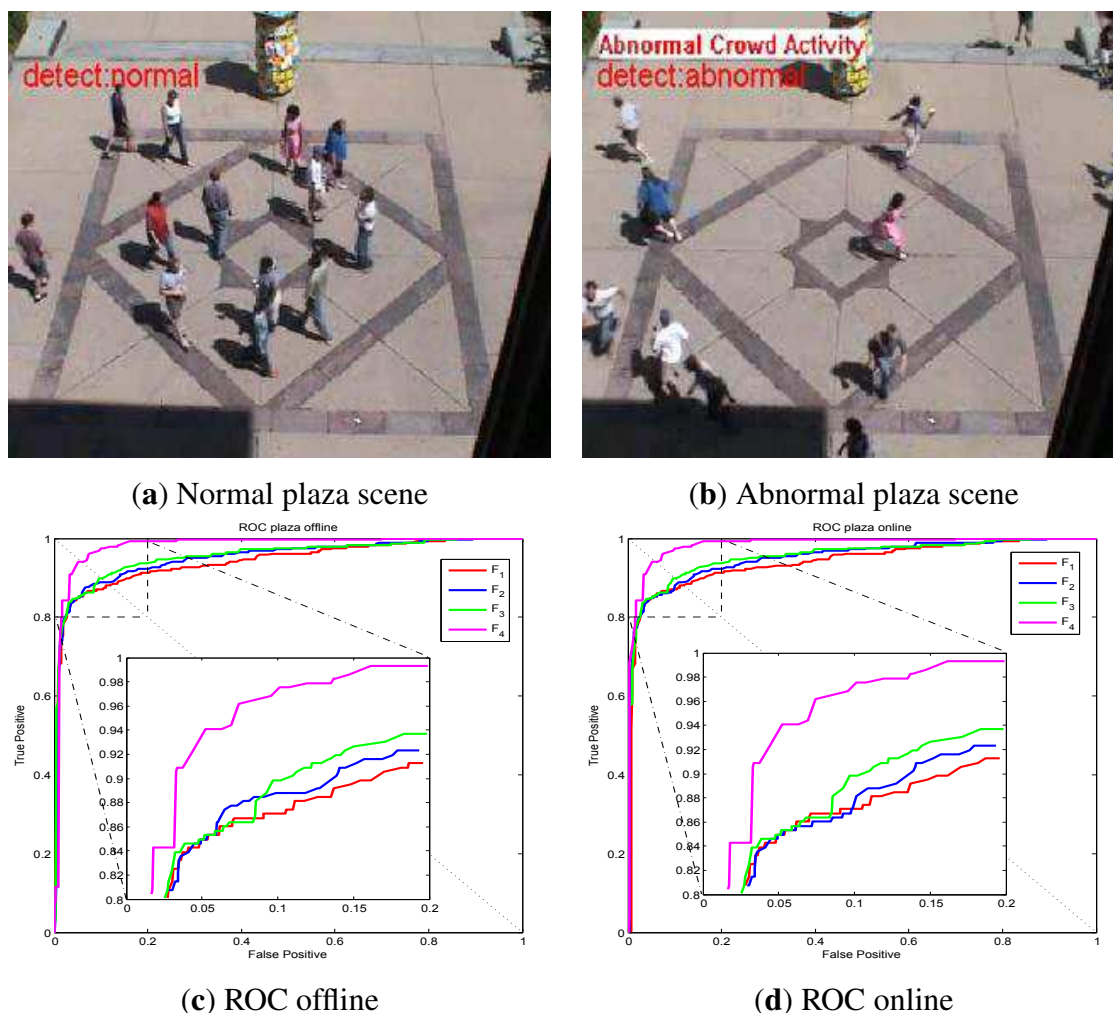


(c) ROC offline



(d) ROC online

Figure 9. Detection results of the plaza scene. (a) The detection result of a normal frame; (b) The detection result of an abnormal panic frame; (c) ROC curve of covariance descriptors constructed from different features F of the plaza scene results via LS-OC-SVM. All the training samples are learned together offline. The biggest AUC value is 0.9800; (d) ROC curve of detection results via online LS-OC-SVM. The biggest AUC value is 0.9839.



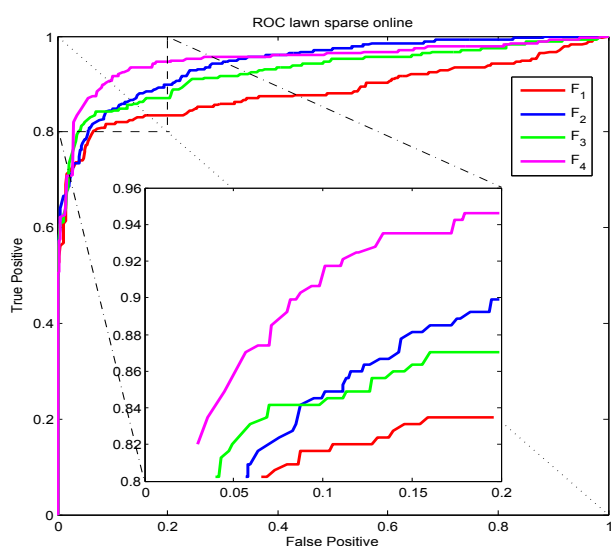
6.3. Abnormal Visual Event Detection via Sparse Online LS-OC-SVM

UMN dataset abnormal event detection results via sparse online LS-OC-SVM proposed in Section 3.4 are presented. Taking the lawn scene as an example, the first normal covariance matrix descriptor from the training samples is included into the dictionary firstly; then, the remaining training covariance descriptors are learned online by the sparse online LS-OC-SVM method. The ROC curve of the detection results of the lawn scene, the indoor scene and the plaza scene are shown in Figure 10a–c, respectively.

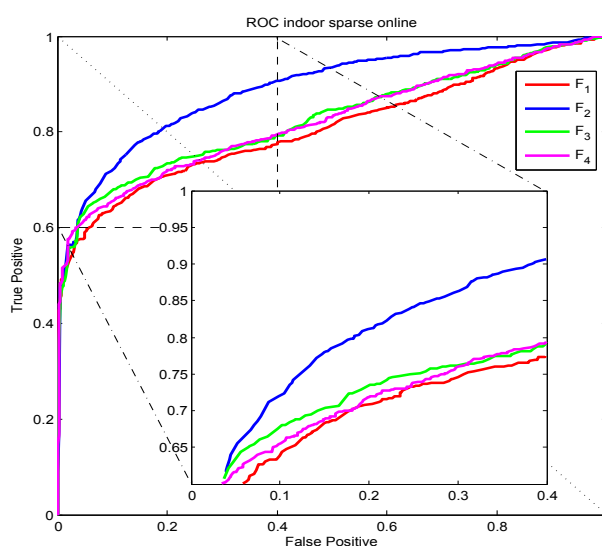
The resulting performances when all training samples are learned offline via one-class SVM (OC-SVM), learned via least squares one-class SVM (LS-OC-SVM), learned via online least squares one-class SVM (online LS-OC-SVM) and learned via sparse online least squares one-class SVM (sparse LS-OC-SVM), are shown in Table 2. The LS-OC-SVM algorithm obtains better performance than the original OC-SVM. The performances of online and sparse online strategy results are similar to the

resulting performances when all training samples are learned offline. The sparse online strategy can be computed efficiently and can adapt to the memory requirement.

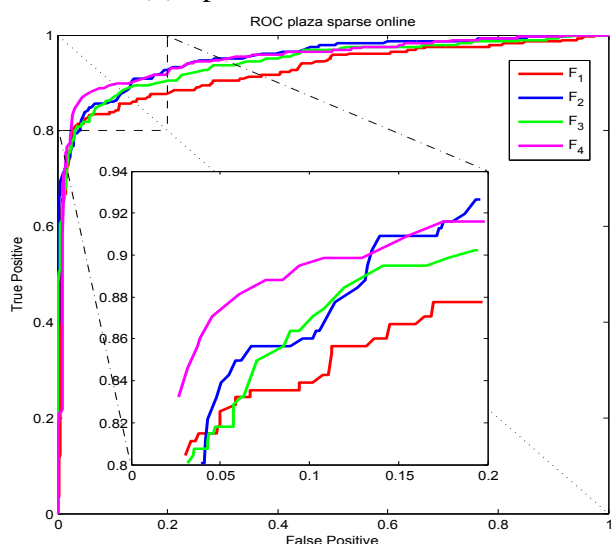
Figure 10. ROC curve of University of Minnesota (UMN) dataset. (a) Sparse online LS-OC-SVM results in the lawn scene. The biggest AUC value is 0.9510; (b) Sparse online LS-OC-SVM results in the indoor scene. The biggest AUC value is 0.8886; (c) Sparse online LS-OC-SVM results in the plaza scene. The biggest AUC value is 0.9515; (d) The ROC curve of the best performance of the lawn, plaza and indoor scene when the training samples are learned via LS-OC-SVM offline. The biggest AUC values of the lawn, indoor and plaza scene are 0.9874, 0.8900 and 0.9800.



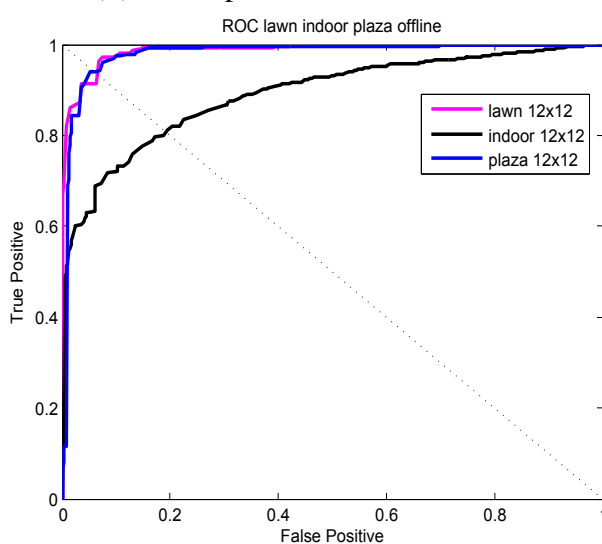
(a) Sparse online lawn scene



(b) ROC sparse online indoor scene



(c) ROC sparse online plaza scene



(d) ROC offline

Table 2. AUC of the abnormal event detection method based on covariance descriptors constructed by different features F via OC-SVM (Section 3.1), original LS-OC-SVM learning training samples together offline (Section 3.2), online LS-OC-SVM (Section 3.3) and sparse online LS -OC-SVM (Section 3.4). The biggest value of each method is shown in bold.

Features	Area under ROC		
	lawn	indoor	plaza
offline OC-SVM			
$F_1(6 \times 6)$	0.9474	0.8381	0.9148
$F_2(6 \times 6)$	0.9583	0.8410	0.9192
$F_3(8 \times 8)$	0.9656	0.8483	0.9367
$F_4(12 \times 12)$	0.9798	0.8744	0.9782
offline LS-OC-SVM			
$F_1(6 \times 6)$	0.9755	0.8605	0.9422
$F_2(6 \times 6)$	0.9738	0.8603	0.9489
$F_3(8 \times 8)$	0.9788	0.8662	0.9538
$F_4(12 \times 12)$	0.9874	0.8900	0.9800
Online LS-OC-SVM			
$F_1(6 \times 6)$	0.9755	0.8616	0.9403
$F_2(6 \times 6)$	0.9720	0.8730	0.9517
$F_3(8 \times 8)$	0.9795	0.8670	0.9563
$F_4(12 \times 12)$	0.9874	0.8904	0.9839
Sparse Online LS-OC-SVM			
$F_1(6 \times 6)$	0.8840	0.8077	0.9245
$F_2(6 \times 6)$	0.9435	0.8886	0.9515
$F_3(8 \times 8)$	0.9269	0.8266	0.9428
$F_4(12 \times 12)$	0.9510	0.8223	0.9501

The resulting performances of the covariance matrix descriptor-based online least squares one-class SVM method, and of state-of-the-art methods, are shown in Table 3. The covariance matrix-based online abnormal frame detection method obtains competitive performance. In generally, our sparse online LS-OC-SVM method is better than others, except sparse reconstruction cost (SRC) [49]. In that paper, multi-scale histogram of optical flow (HOF) was taken as a feature and a testing sample was classified by its sparse reconstruction cost, through a weighted linear reconstruction of the over-complete normal basis set. However, the computation of the HOF takes more time than the computation of covariance. By adopting the integral image [43], the covariance matrix descriptor of the subimage can be computed conveniently. The covariance descriptor can appropriately be used to analyze partial image movement. In [49], the whole training dataset was saved in the memory in advance; then, the dictionary was chosen as an optimal subset for reconstructing. Our sparse online LS-OC-SVM strategy enables one to train the

classifier with sequential inputs. This property makes our proposed method extremely suitable to handle large volumes of training data, while the method in [49] fails to work due to lack of memory.

Table 3. The comparison of our proposed method with state-of-the-art methods for abnormal event detection in the UMN dataset. NN, nearest neighbor. SRC, sparse reconstruction cost. STCOG, spatial-temporal co-occurrence Gaussian mixture models.

Method	Area under ROC		
	lawn	indoor	plaza
Social Force [50]	0.96		
Optical Flow [50]	0.84		
NN [49]	0.93		
SRC [49]	0.995	0.975	0.964
STCOG [51]	0.9362	0.7759	0.9661
LS-SVM (Ours)	0.9874	0.8900	0.9800
Online (Ours)	0.9874	0.8904	0.9839
Sparse Online(Ours)	0.9510	0.8886	0.9515

7. Conclusions

In this paper, we proposed a method to detect abnormal events via online least squares one-class SVM (online LS-OC-SVM) and sparse online least squares one-class SVM (sparse online LS-OC-SVM). Online LS-OC-SVM learns training samples sequentially; sparse online LS-OC-SVM incorporates the coherence criterion to form the dictionary for a sparse representation of the detector. The covariance matrix descriptor encodes the movement feature of the frame to distinguish between normal and abnormal events. The proposed detection algorithms have been tested on a synthetic dataset and a real-world video dataset yielding successful results in detecting abnormal events.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work is partially supported by the China Scholarship Council of Chinese Government and the SURECAP CPER project (fonction de surveillance dans les réseaux de capteurs sans fil via contrat de plan Etat-Région) and the Platform CAPSEC (capteurs pour la sécurité) funded by Région Champagne-Ardenne and FEDER (fonds européen de développement régional).

References

1. Suriani, N.S.; Hussain, A.; Zulkifley, M.A. Sudden event recognition: A survey. *Sensors* **2013**, *13*, 9966–9998.

2. Kosmopoulos, D.; Chatzis, S.P. Robust visual behavior recognition. *IEEE Signal Process. Mag.* **2010**, *27*, 34–45.
3. Utasi, Á.; Czúni, L. Detection of unusual optical flow patterns by multilevel hidden Markov models. *Opt. Eng.* **2010**, doi:10.1117/1.3280284.
4. Xiang, T.; Gong, S. Incremental and adaptive abnormal behaviour detection. *Comput. Vis. Image Underst.* **2008**, *111*, 59–73.
5. Kwak, S.; Byun, H. Detection of dominant flow and abnormal events in surveillance video. *Opt. Eng.* **2011**, doi:10.1117/1.3542038.
6. Jiménez-Hernández, H.; González-Barbosa, J.J.; Garcia-Ramírez, T. Detecting abnormal vehicular dynamics at intersections based on an unsupervised learning approach and a stochastic model. *Sensors* **2010**, *10*, 7576–7601.
7. Davy, M.; Desobry, F.; Gretton, A.; Doncarli, C. An online support vector machine for abnormal events detection. *Signal Process.* **2006**, *86*, 2009–2025.
8. Adam, A.; Rivlin, E.; Shimshoni, I.; Reinitz, D. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2008**, *30*, 555–560.
9. Boiman, O.; Irani, M. Detecting irregularities in images and in video. *Int. J. Comput. Vis.* **2007**, *74*, 17–31.
10. Piciarelli, C.; Micheloni, C.; Foresti, G.L. Trajectory-based anomalous event detection. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, *18*, 1544–1554.
11. Piciarelli, C.; Foresti, G.L. On-line trajectory clustering for anomalous events detection. *Pattern Recognit. Lett.* **2006**, *27*, 1835–1842.
12. Calavia, L.; Baladrón, C.; Aguiar, J.M.; Carro, B.; Sánchez-Esguevillas, A. A semantic autonomous video surveillance system for dense camera networks in smart cities. *Sensors* **2012**, *12*, 10407–10429.
13. Lee, Y.S.; Chung, W.Y. Visual sensor based abnormal event detection with moving shadow removal in home healthcare applications. *Sensors* **2012**, *12*, 573–584.
14. Shilton, A.; Palaniswami, M.; Ralph, D.; Tsoi, A.C. Incremental training of support vector machines. *IEEE Trans. Neural Netw.* **2005**, *16*, 114–131.
15. Lau, K.; Wu, Q. Online training of support vector classifier. *Pattern Recognit.* **2003**, *36*, 1913–1920.
16. Cauwenberghs, G.; Poggio, T. Incremental and decremental support vector machine learning. *Adv. Neural Inf. Process. Syst.* **2000**, *1*, 409–415.
17. Diehl, C.P.; Cauwenberghs, G. SVM Incremental Learning, Adaptation and Optimization. In Proceedings of International Joint Conference on Neural Networks (IJCNN), Portland, OR, USA, 20–24 July 2003; pp. 2685–2690.
18. Laskov, P.; Gehl, C.; Krüger, S.; Müller, K.R. Incremental support vector learning: Analysis, implementation and applications. *J. Mach. Learn. Res.* **2006**, *7*, 1909–1936.
19. Desobry, F.; Davy, M.; Doncarli, C. An online kernel change detection algorithm. *IEEE Trans. Signal Process.* **2005**, *53*, 2961–2974.

20. Zhang, Y.; Meratnia, N.; Havinga, P. Adaptive and Online One-Class Support Vector Machine-Based Outlier Detection Techniques for Wireless Sensor Networks. In Proceedings of International Conference on Advanced Information Networking and Applications Workshops (WAINA), Bradford, UK, 26–29 May 2009; pp. 990–995.
21. Gómez-Verdejo, V.; Arenas-García, J.; Lazaro-Gredilla, M.; Navia-Vazquez, A. Adaptive one-class support vector machine. *IEEE Trans. Signal Process.* **2011**, *59*, 2975–2981.
22. Kivinen, J.; Smola, A.J.; Williamson, R.C. Online learning with kernels. *IEEE Trans. Signal Process.* **2004**, *52*, 2165–2176.
23. Tax, D. One-Class Classification. Ph.D. Thesis, Delft University of Technology, Delft, The Netherlands, 2001.
24. Tax, D.M.; Duin, R.P. Support vector data description. *Mach. Learn.* **2004**, *54*, 45–66.
25. Noumir, Z.; Honeine, P.; Richard, C. Online One-Class Machines Based on the Coherence Criterion. In Proceedings of the 20th European Signal Processing Conference (EUSIPCO), Bucharest, Romania, 27–31 August 2012; pp. 664–668.
26. Kim, P.J.; Chang, H.J.; Choi, J.Y. Fast Incremental Learning for One-Class Support Vector Classifier Using Sample Margin Information. In Proceedings of the 19th International Conference on Pattern Recognition (ICPR), Tampa, FL, USA, 8–11 December 2008; pp. 1–4.
27. Schölkopf, B.; Platt, J.C.; Shawe-Taylor, J.; Smola, A.J.; Williamson, R.C. Estimating the support of a high-dimensional distribution. *Neural Comput.* **2001**, *13*, 1443–1471.
28. Suykens, J.; Lukas, L.; van Dooren, P.; de Moor, B.; Vandewalle, J. Least Squares Support Vector Machine Classifiers: A Large Scale Algorithm. In Proceedings of European Conference on Circuit Theory and Design (ECCTD), Stresa, Italy, 29 August–2 September 1999; pp. 839–842.
29. Choi, Y.S. Least squares one-class support vector machine. *Pattern Recognit. Lett.* **2009**, *30*, 1236–1240.
30. Vapnik, V.N. *Statistical Learning Theory*; Wiley: New York, NY, USA, 1998.
31. Vapnik, V.N.; Lerner, A. Pattern recognition using generalized portrait method. *Autom. Remote Control* **1963**, *24*, 774–780.
32. Boser, B.E.; Guyon, I.M.; Vapnik, V.N. A Training Algorithm for Optimal Margin Classifiers. In Proceedings of ACM the 5th Annual Workshop on Computational Learning Theory (COLT), Pittsburgh, PA, USA, 27–29 July 1992; pp. 144–152.
33. Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines and other Kernel-Based Learning Methods*; Cambridge University Press: Chambridge, UK, 2000.
34. Canu, S.; Grandvalet, Y.; Guigue, V.; Rakotomamonjy, A. SVM and Kernel Methods Matlab Toolbox, updated on 20 February 2008; A SVM Toolbox fully written in Matlab, even the QP solver; Perception Systèmes et Information, INSA de Rouen, Rouen, France, 2008.
35. Schölkopf, B.; Smola, A.J.; Williamson, R.C.; Bartlett, P.L. New support vector algorithms. *Neural Comput.* **2000**, *12*, 1207–1245.
36. Ben-Hur, A.; Horn, D.; Siegelmann, H.T.; Vapnik, V. Support vector clustering. *J. Mach. Learn. Res.* **2002**, *2*, 125–137.
37. Suykens, J.A.; Vandewalle, J. Least squares support vector machine classifiers. *Neural Process. Lett.* **1999**, *9*, 293–300.

38. Suykens, J.; Gestel, T.V.; Brabanter, J.D.; Moor, B.D.; Vandewalle, J. *Least Squares Support Vector Machines*; World Scientific: Singapore, 2002.
39. Honeine, P. Online kernel principal component analysis: A reduced-order model. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1814–1826.
40. Tropp, J.A. Greed is good: Algorithmic results for sparse approximation. *IEEE Trans. Inf. Theory* **2004**, *50*, 2231–2242.
41. Richard, C.; Bermudez, J.C.M.; Honeine, P. Online prediction of time series data with kernels. *IEEE Trans. Signal Process.* **2009**, *57*, 1058–1067.
42. Horn, B.K.; Schunck, B.G. Determining optical flow. *Artif. Intell.* **1981**, *17*, 185–203.
43. Tuzel, O.; Porikli, F.; Meer, P. Region covariance: A fast descriptor for detection and classification. *Lect. Notes Comput. Sci.* **2006**, *3952*, 589–600.
44. Schölkopf, B.; Smola, A.J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*; MIT Press: Cambridge, MA, USA, 2002.
45. Hall, B. *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction*; Springer: Berlin, Germany, 2003.
46. Cong, G.; Fanzhang, L.; Cheng, S. Research on Lie Group kernel learning algorithm. *J. Front. Comput. Sci. Technol.* **2012**, *6*, 1026–1038.
47. Detection of Events—Detection of Unusual Crowd Activity. Available online: <http://mha.cs.umn.edu/Movies/Crowd-Activity-All.avi> (accessed on 12 December 2013).
48. Hoffmann, H. Kernel PCA for novelty detection. *Pattern Recognit.* **2007**, *40*, 863–874.
49. Cong, Y.; Yuan, J.; Liu, J. Sparse Reconstruction Cost for Abnormal Event Detection. In Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 20–25 June 2011; pp. 3449–3456.
50. Mehran, R.; Oyama, A.; Shah, M. Abnormal Crowd Behavior Detection Using Social Force Model. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 935–942.
51. Shi, Y.; Gao, Y.; Wang, R. Real-Time Abnormal Event Detection in Complicated Scenes. In Proceedings of the 20th International Conference on Pattern Recognition (ICPR), Istanbul, Turkey, 23–26 August 2010; pp. 3653–3656.