Video Anomaly Detection With Compact Feature Sets for Online Performance

Roberto Leyva, Victor Sanchez, *Member, IEEE*, and Chang-Tsun Li, *Senior Member, IEEE*

Abstract—Over the past decade, video anomaly detection has been explored with remarkable results. However, research on methodologies suitable for online performance is still very limited. In this paper, we present an online framework for video anomaly detection. The key aspect of our framework is a compact set of highly descriptive features, which is extracted from a novel cell structure that helps to define support regions in a coarse-to-fine fashion. Based on the scene's activity, only a limited number of support regions are processed, thus limiting the size of the feature set. Specifically, we use foreground occupancy and optical flow features. The framework uses an inference mechanism that evaluates the compact feature set via Gaussian Mixture Models, Markov Chains, and Bag-of-Words in order to detect abnormal events. Our framework also considers the joint response of the models in the local spatio-temporal neighborhood to increase detection accuracy. We test our framework on popular existing data sets and on a new data set comprising a wide variety of realistic videos captured by surveillance cameras. This particular data set includes surveillance videos depicting criminal activities, car accidents, and other dangerous situations. Evaluation results show that our framework outperforms other online methods and attains a very competitive detection performance compared with state-of-the-art non-online methods.

Index Terms—Video anomaly detection, online processing, video surveillance.

I. Introduction

AUTOMATIC video surveillance is one of the most active areas in computer vision. At the core of automatic video surveillance are anomaly detection methods, which have been shown to be highly effective to detect unusual events without *a priori* knowledge about these events [1], [2]. Despite important advances in video anomaly detection over the past decade, there is a lack of methods specifically designed for online processing, which deters its applicability in practical scenarios. Within this context, online processing refers to attaining a frame processing time that is shorter than the time

it takes to process a new frame according to the sequence's frame rate [3], [4]. Another important factor that also deters their applicability in practical scenarios is the fact that research on realistic surveillance videos is still very limited. State-of-the-art methods have been mainly designed and tested using datasets that poorly represent realistic abnormal events. These datasets usually contain simulated scenes with actors behaving abnormally, e.g., [4]–[7]; or more realistic scenes but with a very limited number of abnormal events, e.g., [8]–[10].

In this paper, we propose a new video anomaly detection framework suitable for online processing. Our framework employs a novel cell structure that helps to extract the scene's motion information based on local activity. This significantly reduces the number of features to be processed during the training and inference stages, which consequently reduces computational times. The main characteristics of our framework are:

- The extracted compact set of features comprises features from foreground occupancy and optical flow. Features from foreground occupancy help to efficiently capture events associated with weak motion, such as loitering or the abnormal presence of subjects; while features from optical flow are useful to detect events associated with sudden motion, such as panic or fights.
- Multiple inference models are employed to accurately describe the activity of challenging scenes, where anomalous events can be due to sudden motion, weak motion, or both. This is particularly useful to attain a good performance on scenes depicting realistic events; e.g., robberies, car accidents and other dangerous situations.

We test our framework on popular existing datasets and on a new rich collection of real sequences captured by surveillance cameras and depicting realistic events. Evaluation results show that our framework achieves competitive results compared to non-online methods, and outperforms online methods. In both cases, our framework attains frame processing times that are suitable for online processing of sequences with frames rates of up to 30 frames per second (FPS).

The rest of the paper is organized as follows. In the next section we discuss the related work and the contributions of our framework. In Section III, we describe in detail our proposed framework. In Section IV, we briefly describe the datasets used for evaluation and present the performance evaluation results. We conclude this paper in Section V.

II. Related Work

Anomaly detection methods can be classified into two main categories: *accuracy-oriented* methods, which are mainly concerned with improving the detection accuracy, and *processing-*

time-oriented methods, which are mainly concerned with reducing frame processing times. The latter category aims at attaining online performance.

The class of *accuracy-oriented* methods has seen important contributions over the past decades. However, the good performance of these methods is usually attained at the expense of increasing frame processing times. These methods are characterized by employing various techniques to first select the spatio-temporal regions of the scene to be modeled and analyzed. Such techniques include dense scanning [3], [11], multi-scale scanning [12]–[14] and convolution-based Spatio-Temporal Interest Point (STIP) detection [15], [16]. These techniques usually provide sufficient data to capture the scene's dynamics and spatio-temporal compositions; however, the number of spatio-termporal regions selected for analysis may result in a large number of features to be processed [3], [11]–[13], [15]–[17]. Although important efforts have been made to reduce the complexity associated with the definition of a scene's spatio-temporal compositions [3], [15], many of the proposed improvements may still require considerably long computations [3], [11], [15]. Another important characteristic of these *accuracy-oriented* methods is their highly descriptive features used to improve performance. Among these, optical flow features have been shown to increase detection accuracy [18], [19]. For example, in [19] the authors propose a fully unsupervised non-negative sparse coding based approach that employs histograms of optical flow (HOFs) to detect abnormalities in crowded scenes with promising performances. In [12], the authors adopt Multi-scale HOFs (MHOFs), which preserve temporal contextual information, to detect anomalies in crowded scenes as a matching problem. Computing such descriptive features, however, may require long processing times [3], [11], [14], [19], [20]. For example, local descriptors computed using dense scanning techniques have been shown to improve performance, but at the expense of multiple repeated computations [3], [13].

Processing-time-oriented methods have recently gained interest within the area of video anomaly detection [3], [21], [22]. These methods usually reduce computational times by reducing the number of features to be processed per frame [21]–[23] or by employing local low-complexity descriptors [3], [15], [21], [22]. For example, the work of Lu's *et al.* [21] and that of Biswas and Babu [22] manage to model a small number of features even though they employ multi-scale scanning techniques. *Processing-time-oriented* methods may also employ features that are fast to compute, but not highly descriptive. For example, in [22] the authors employ the motion vectors of a video sequence as features in a histogram-binning scheme. In [21], the authors employ local temporal gradients extracted in a multi-scale fashion as the main feature. Another common approach to reduce processing times is by employing cell-based methods to extract features from fixed spatio-temporal regions [12], [13], [20], [23]. Cell-based methods therefore do not require STIPs or other saliency detection techniques; moreover, they can be used to limit the number of extracted features [13].

It is evident that there is a trade-off between detection accuracy and processing times in video anomaly detection methods. The challenge is then to appropriately balance this trade-off by employing few, but highly descriptive, features in order to attain online performance with a competitive accuracy. Our proposed framework balances this trade-off by employing a compact set of optical flow and foreground occupancy features that are highly descriptive. This is achieved by defining a novel cell structure on the scene, from which features are extracted only from those cells that are deemed to be relevant to the analysis. The main differences between existing approaches and our framework are as follows:

1) Although other cell structures have been previously proposed, e.g., [13], [23], [24], our framework employs a novel fine-to-coarse cell structure that is computationally efficient and uses cells of multiple sizes. The latter helps to take into account the intrinsic camera-object distance in the analysis.

2) Instead of employing common convolution-based STIPs, e.g. [15], [16], our framework employs their binary counterpart FAST (Fast Accelerated Segmentation Test) [25], which is a binary-based technique that detects interest points by comparing the intensity of a particular pixel with that of its neighbors. FAST STIPs therefore also contribute to attain online performance.

3) In order to extract highly descriptive features, our framework employs two sources of motion information with online performance for the first time. Specifically, it employs optical flow and background subtraction information, both of which have been successfully used separately in the past [3], [23].

4) Our framework successfully employs optical flow features, which are known to be highly computationally complex [3], [11], with online performance. This is attained by extracting these features only from those cells that are relevant to the analysis.

III. PROPOSED FRAMEWORK

In the same spirit of local-region based approaches proposed in [12], [14], [17], [26], and [27], we propose to analyze the motion in local areas of the scene and build a probabilistic inference model considering local spatio-temporal information. We are interested in addressing the high computational complexity and long processing times usually associated with state-of-the-art anomaly detection methods in order to attain online performance. The main core of the proposed framework is then to efficiently describe the events in the scene without computing a significant number of features. Hence, we focus on an efficient scene representation using a compact set of features. Our framework is graphically summarized in Fig. 1. In the Training Stage, we first construct a cell structure for the whole scene to define the spatio-termporal regions, or video volumes, to be analyzed (Section III-A). A compact set of features is then extracted from a limited number of video volumes. These features are based on foreground occupancy and optical flow information (Section III-B). The compact set of features is analyzed to construct various models. In the Detection Stage, after extracting a compact set
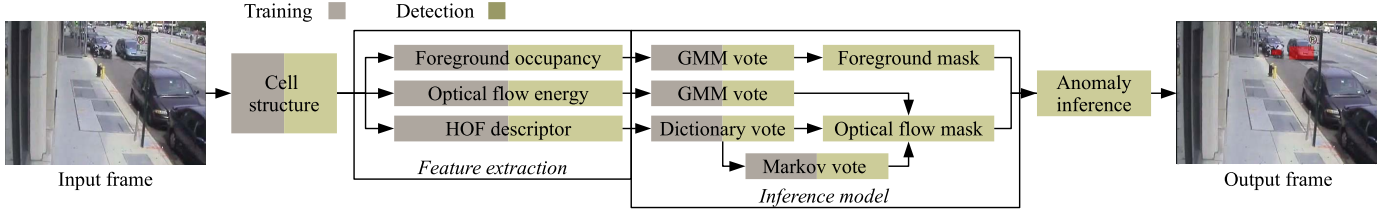
Fig. 1. Proposed framework. In the Training Stage, we build a cell structure to define the support regions to be analyzed. Features from foreground occupancy and optical flow are extracted to construct four models. In the Detection Stage, extracted features are evaluated using the models constructed in the previous stage. The output of these models are then used to create two inference likelihood masks that lead to the detection of anomalous events.

of features, the models are used to detect anomalous video volumes (Section III-C). Finally, an inference mechanism that considers the local spatio-temporal neighborhood of cells is used to detect abnormal events (Section III-D.)

A. Cell Structure

Regions of the scene that are relatively close to the camera provide more descriptive information than those located far from it. Therefore, taking into account the camera's position in the scene to extract features can significantly enhance performance [23], [28], [29]. To this end, we extract information from the scene using a cell structure where cells are defined using a non-overlapping grid over the spatial domain. An equal-sized cell structure is intuitively not appropriate to deal with the camera's position and the associated scene's perspective, as it results in extracting features equally from all regions of the scene regardless of their position relative to the camera. A common solution to compensate for the scene's perspective is to track objects as they enter and exit the scene. However, tracking objects is particularly challenging and computationally complex in crowded scenes [30]–[32]. In this work, we assume that the camera acquiring an unobstructed view of a scene is installed in a high position looking downwards. This is a valid assumption for the majority of video surveillance cameras. When the camera looks downwards from a high position to acquire an unobstructed scene, the lower region of the scene then tends to be the one closest to the camera. Based on this assumption, we propose to create a grid with variable-sized cells where the largest cells are located at the lower region of the scene (i.e., regions closest to the camera), and the smallest cells are located at the upper region of the scene (i.e., regions farthest from the camera). Large cells then provide more information to compute features.

We create the cell structure and define the size of the constituent cells according to the frame size. Starting from the frame's top border, let y_k be the vertical dimension of the kth cell as associated with its vertically adjacent cell, i.e., the $(k+1)$th cell, by:

$$y_{k+1} = \alpha y_k, \tag{1}$$

where $\alpha > 1$ is a *growing rate* that makes the $(k+1)$th cell larger than the kth cell. Thus, the vertical size of the frame, denoted by Y, can be expressed in terms of the recursive vertical dimension of each cell as follows:
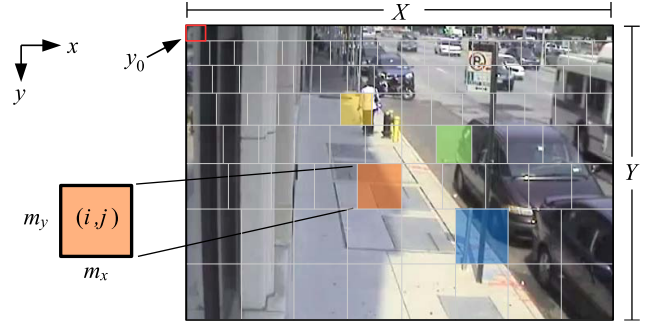
$$Y = \sum_{k=0}^{n} \alpha^k y_0, \tag{2}$$

Fig. 2. Example cell structure for a scene. Cells of different size are highlighted in different colors for illustration purposes. Largest cells depict regions closest to the camera. The cell c at position (i, j) has spatial dimensions of $m_x \times m_y$.

where n is the number of cells along the vertical dimension and y_0 is the vertical dimension of the smallest cell. To find n, we initially set y_0 to an initial value. Eq. 2 can be easily transformed into its geometric series form:

$$Y/y_0 = \frac{\alpha^{n+1} - 1}{\alpha - 1}. \tag{3}$$

The value of n can then be calculated as follows:

$$n = \lfloor \log_\alpha \left(Y/y_0(\alpha - 1) + 1 \right) - 1 \rceil. \tag{4}$$

We use Eq. 4 to adjust the vertical dimension of the smallest cell. This adjusted dimension is denoted as $\hat{y}_0$:

$$\hat{y}_0 = \left\lfloor \frac{\alpha - 1}{\alpha^{n+1} - 1} Y \right\rfloor. \tag{5}$$

We follow a similar procedure to determine the size of the horizontal dimension of the cells. Let X denote the horizontal dimension of the frame. Starting at the top border of the frame, at position $X/2$, i.e., the mid-section of the frame, we use the same growing rate α to increase the horizontal dimension of cells. Specifically, we modify this dimension in a symmetrical manner from position $X/2$. We, however, do not adjust the initial horizontal dimension, as we expect to find most of the changes in objects' size along the vertical dimension. An example of our proposed cell structure is shown in Fig. 2. Appendix describes in more detail the construction of our proposed cell structure.
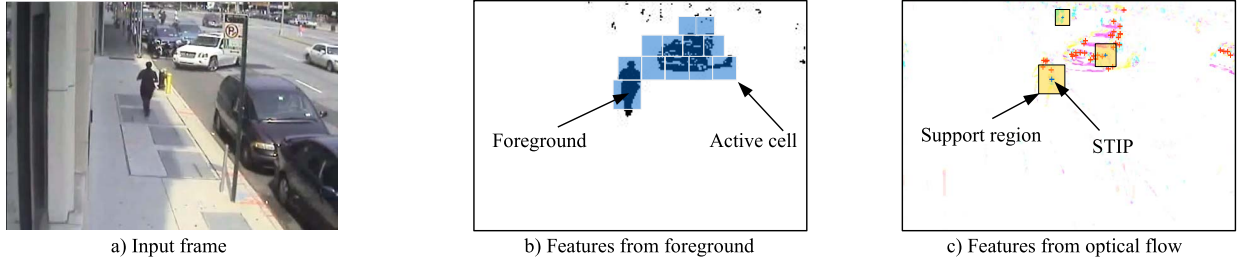
Fig. 3. Features extraction. (a) Input frame. (b) From the foreground mask B_t, we extract features from active cells . (c) From absolute frames differences, we detect FAST STIPs and the corresponding support regions are encoded using optical flow energy and HOF descriptors. In (c), three example support regions of different size are shown for illustration.

B. Feature Extraction

Motion and change detection has an important impact on the features extracted from the video sequences. Many motion and change detection algorithms have been developed that perform well in some types of videos, but most are sensitive to sudden illumination changes, environmental conditions, background/camera motion, and shadows. To this end, important efforts have been made by the CDNET initiative to provide a benchmark for testing and ranking existing and new algorithms for change and motion detection [33]–[36]. In this work, we employ motion and change detection algorithms that allows us to extract a compact set of very descriptive features. Specifically, two sources of motion information are used, namely, background subtraction and optical flow information, in order to extract foreground occupancy and optical flow features, respectively.

1) Foreground Occupancy Features: Foreground information is very useful to determine occupancy and long term events [23]. Foreground occupancy refers to the information that captures the size of objects in the scene and their corresponding duration in that scene [23]. To this end, we first employ background subtraction to detect the objects in the scene [37]. Applying background subtraction results in a collection of binary masks, one for each frame, where true logical values represent the foreground. We denote each of these binary masks as B_t for frame t. For each cell c located at position (i, j) in the cell structure proposed in Section III-A, we define a video volume $u \in \mathbb{R}^3$ on B_t. Video volume u has dimensions $m_x \times m_y \times m_t$, where dimensions m_x and m_y are determined by the horizontal and vertical dimensions of the cell, respectively, and m_t denotes the number of frames, which is fixed for all video volumes. The size of the detected objects and their duration in the scene can then be easily computed by counting the number of foreground pixels in each video volume u. For each video volume u associated with cell at postion (i, j), we then compute feature $F(i, j) \in \mathbb{R}$, which represents the foreground occupancy, as follows:

$$F(i, j) = \frac{1}{N} \sum_{n=1}^{N} u^{(n)}, \qquad (6)$$

where N is the number of pixels in u.

Only those cells whose associated video volumes that have a foreground occupancy $F(i, j)$ above a threshold are considered as active. Specifically, a cell is considered as active if at least 10% of the pixels in the associated video volume

belongs to the foreground (see Fig. 3b). Only video volumes associated with active cells are further analyzed. This helps to avoid analyzing regions that mostly depict background, thus reducing frame processing times and false alarms.

2) Optical Flow Features: To extract features from optical flow, we first detect STIPs using the FAST detector [25]. FAST is a binary-based technique that detects interest points by comparing the intensity of a particular pixel p with that of its neighbors. If all neighbouring pixel intensities are greater or less than the pixel intensity of p, then the pixel is considered to be an interest point. This particular binary-based detector has significant advantages in terms of speed over convolution-based detectors [15], [38]. Note that although 3D spatio-temporal detectors have been widely used for the same purpose, their computational times are considerably long for online processing. In order to discard background information, we apply the FAST detector on the absolute temporal frame differences (see Fig. 3c). For each space location (x_p, y_p) detected by FAST at the frame difference t_p, we generate a video volume $v \in \mathbb{R}^3$ of size $m_x \times m_y \times m_t$ centered at (x_p, y_p, t_p). Sizes m_x and m_y are determined by the size of cells in the structure proposed in Section III-A and size m_t is fixed for all video volumes. Specifically, m_x and m_y are equal to the horizontal and vertical size, respectively, of the cell in which the space location (x_p, y_p) detected by FAST falls into. We compute optical flow energy and a HOF descriptor [39] to generate the feature pair $\{O_p(x_p, y_p, t_p), w_p(x_p, y_p, t_p)\}$ for each STIP detected by FAST, where

- $O_p(x_p, y_p, t_p)$ is the optical flow energy computed as:

$$O_p(x_p, y_p, t_p) = \frac{1}{N} \sum_{n=1}^{N} \left\| \left[v_x^{(n)}, v_y^{(n)} \right] \right\|_2, \qquad (7)$$

  where v_x and v_y correspond to the horizontal and vertical optical flow components, respectively, for the N pixels in video volume v; and
- $w_p(x_p, y_p, t_p)$ is a HOF descriptor; a 5-bin optical flow histogram calculated in the range $[0, 4/5\pi]$ (see Fig. 4). The histograms are normalized using ℓ_1 normalization.

C. Inference Model

We build an inference model to detect anomalous video volumes. The model, as depicted in Fig. 1, is composed of four sub-models. Foreground occupancy features and optical flow energy features are analyzed separately by two distinct
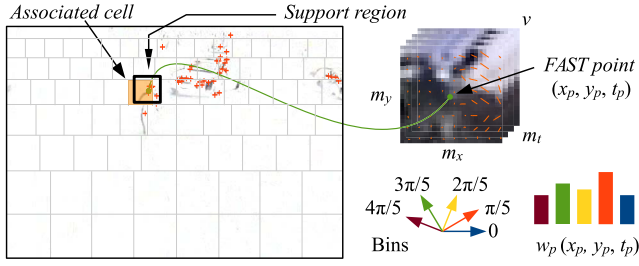
Fig. 4. HOF descriptors are extracted from video volume v, which is the support region for the FAST STIP at position (x_p, y_p, t_p). The size of v is determined according to the cell in which the space location (x_p, y_p) falls into.

Gaussian Mixture Models (GMMs). The HOF descriptors are analyzed by a Dictionary Model and a Markov Model.

1) GMM for Foreground Occupancy: to capture variable-sized objects and long-term activity, we use the foreground occupancy information of the scene. This information allows us to deal efficiently with objects that appear for different periods of time in the scene. Foreground occupancy also provides information about the size of objects, which is captured by the number of active cells, as described in Section III-B.1. The foreground occupancy of each cell (see Eq. 6) is analyzed by a GMM (see Eq. 8) with parameters $\theta^F = \{\pi_k^F, \mu_k^F, \sigma_k^F\}$, representing the weight, mean and standard deviation, respectively, of the kth component of the GMM. The model's elements are determined exhaustively by iterating the Akaike Information Criterion (AIC) over the model:

$$p_{FG}(F(i,j) \mid \theta^F) = \sum_k \pi_k^F \mathcal{N}(F(i,j) \mid \mu_k^F, \sigma_k^F), \quad (8)$$

where $\mathcal{N}$ is a normal distribution. For the GMM model of Eq. 8, the AIC compares models in the light of information entropy as a measure of Kullback-Leibler divergence. The AIC for the given model is:

$$AIC(k, F) \triangleq log\left(p_{FG}(F \mid \theta_{MLE}^F)\right) - dof(k), \quad (9)$$

where F represents the values to be modeled, whose likelihood is to be maximized by the corresponding distribution of parameters; and θ_{MLE}^F is the corresponding set of parameters that results in the maximum likelihood estimation (MLE). Experimentally, we observe that more than 10 degrees of freedom (dof) usually do not provide relevant information. Thus, we limit the number of iterations to $k = 10$.

To evaluate the likelihood of the current cell's foreground occupancy, $F(i,j)$, we also consider the likelihood of the immediate neighboring cells, as follows:

$$p_{FGL}(F(i,j)) = \prod_{x=i-1}^{i+1} \prod_{y=j-1}^{j+1} \delta_{x,y} p_{FG}(F(x,y) \mid \theta^F), \quad (10a)$$

$$\delta_{a,b} = \begin{cases} 1, & a = 0, b = 0 \\ 0.2, & otherwise, \end{cases} \quad (10b)$$

where δ is an exception-modified Kronecker delta function.

2) GMM for Optical Flow Energy: events like panic and other sudden variations in the scene might not be properly described by the number of objects in the scene, their size or long-term activity, but rather by the speed of their motion. In order to capture sudden variations in the scene, we use a GMM of optical flow energy with parameters $\theta^O = \{\pi_k^O, \mu_k^O, \sigma_k^O\}$, representing the weight, mean and standard deviation, respectively, of the kth component of the GMM, as follows:

$$p_{OF}(O_p(x_p, y_p, t_p) \mid \theta^O)$$
$$= \sum_k \pi_k^O \mathcal{N}(O_p(x_p, y_p, t_p) \mid \mu_k^O, \sigma_k^O). \quad (11)$$

The model in Eq. 11 is also estimated by recursively minimizing the AIC metric, as done for the GMM of foreground occupancy. The Akaike criterion for model p_{OF} is then:

$$AIC(k, O_p) \triangleq log\left(p_{OF}(O_p \mid \theta_{MLE}^O)\right) - dof(k), \quad (12)$$

where O_p represents the values to be modeled and θ_{MLE}^O is the corresponding set of parameters that results in the maximum likelihood estimation.

3) Dictionary Model for HOF Descriptors: we are interested in capturing the intrinsic activity information of the scene by taking into account the fact that the activity may vary within the scene. For instance, in a scene depicting a traffic intersection, the activity in the sidewalk may differ a lot from the activity in the road. However, anomalous events may occur in both, the road and the sidewalk. Based on this, we propose to create individual dictionaries, one per cell, instead of creating a global dictionary as in [3], [15], [23], and [40] . It has been previously shown that individual Bag-of-Words can significantly enhance performance in action recognition [41]. Each cell defined as described in Section III-A is assigned a dictionary that is generated from the set S of HOF descriptors within the cell. To this end, we use k-means to define the cluster centroid $z_i \in \mathbb{R}^5$ in a dictionary, as follows:

$$z_i : \arg\min_S \sum_{i=1}^k \sum_{w_p \in S} \left\| w_p - z_i \right\|_2^2, \quad (13)$$

where $w_p \in \mathbb{R}^5$ is a HOF descriptor as defined in Section III-B.2. The dictionary generated is associated with a probabilistic vote according to the ℓ_2 distance. The distance d to those *seen* words is expected to be $\ell_2 \simeq 0$ if the word is present in the dictionary and $\ell_2 \gg 0$ otherwise. We calculate the posterior likelihood of the distance of the observed words as a normal distribution with parameters $\theta^{DIC} = \{\mu^{DIC}, \sigma^{DIC}\}$, representing the mean and standard deviation of the distribution, respectively:

$$p_{DIC}(d_p \mid \theta^{DIC}) = \mathcal{N}(d_p \mid \mu^{DIC}, \sigma^{DIC}), \quad (14)$$

where d_p is the ℓ_2 distance of the word $w_p \in S$ to the cluster centroid z_i.

4) Markov Model for HOF Descriptors: in order to capture unusual word ensembles [3], we employ a Finite-State Markov Chain (FSMC). Markov Models have been successfully used to detect anomalous events in the context of long-term
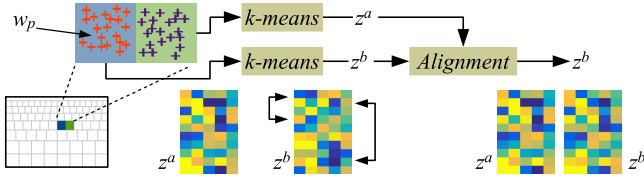
Fig. 5. Individual dictionaries are built from the set S of HOF descriptors within each cell. The dictionaries are aligned to ensure the correct FSMC transitions.

activities [42], [43]. However, these models are usually designed to detect anomalous events in a global context. A global context may be difficult to address if the activity in the scene varies significantly across different regions, e.g., the activity in a sidewalk and the road in a scene depicting a traffic intersection. Thus, we use a local model to detect anomalous events by considering the Markov Model of different regions. Let us consider the current state X_l given by the matching label l of the local dictionary, the FSMC probability is then:

$$p_{MRV}(X_{1:L}) = p(X_1) \prod_{l=2}^{L} p(X_L \mid X_{l-1}), \quad (15)$$

where L is the number of transitions defined by the total number of labels L in the local dictionary. The matching label index, l, is defined as:

$$l : \arg\min_{l} \left\| w_p - z_l \right\|_2^2, \quad (16)$$

and the associated transition matrix, A, is defined as:

$$A_{ij} = p(X_l = j \mid X_{l-1} = i), \quad (17a)$$

$$\sum_j A_{ij} = 1. \quad (17b)$$

We are interested in knowing how likely it is that words i and j co-occur. The probability of observing the two words $\{i, j\}$ is given by the occurrence n of the words, as follows:

$$A_{ij}(n) = p(X_{l+n} = j \mid X_l = i). \quad (18)$$

The order of occurrence of words is not important if the number of analyzed frames is limited, as it is in this case. Therefore, we can discard their order of occurrence making matrix A symmetrical.

Since we use a number of different isolated dictionaries, i.e., one per cell, the FSMC requires that the transition states between neighboring regions correspond to the same matching labels. For example, in the case of two neighboring cells, a and b, with different labels associated to the same word (see Fig. 5). We therefore align a pair of neighboring dictionaries, $z^a \in \mathbb{R}^{k \times 5}$ and $z^b \in \mathbb{R}^{k \times 5}$, computed using k-means, as follows:

$$i : \arg\min_{i} \left\| z_j^a - z_i^b \right\|_2^2, \quad (19a)$$

$$z_j^c = z_i^b, \quad (19b)$$

where $z_j^a \in \mathbb{R}^5$ and $z_i^b \in \mathbb{R}^5$ are the words j and i associated to the dictionaries for cells a and b, respectively; and $z^c \in \mathbb{R}^{k \times 5}$ is an empty auxiliary dictionary. By setting z^b equal to z^c,

the dictionary alignment is performed. After this alignment procedure, the matching labels l_p^a and l_p^b for the word w_p in dictionary a and b, respectively, are the same, i.e., $l_p^a = l_p^b$. This ensures that the FSMC transition is the same in a local spatial region of the scene.

D. Anomaly Inference

The anomaly inference mechanism works in two joint phases. In the first phase, the models are analyzed for potential anomalous events to generate two likelihood binary masks. In the second phase, the posterior vote of these two binary masks is jointly analyzed to determine anomalous events.

1) First Phase - Mask Generation: two binary masks are generated in this phase. The first mask is generated by thresholding the posterior likelihood of the foreground occupancy model. Fig. 6b shows a sample foreground occupancy likelihood map before thresholding. The second mask is generated by thresholding the likelihood of the optical flow energy and HOF descriptor model. Fig. 6c shows a sample optical flow likelihood map before thresholding. The foreground occupancy binary mask, $MASK_{FG}$, is then generated by the posterior likelihood of each active cell given by the pdf of the model in Eq. 6, as follows:

$$\gamma_{FG} = -\log\left(p_{FGL}\right), \quad (20a)$$

$$MASK_{FG} = \begin{cases} anomalous, & \gamma_{FG} > \epsilon_{FG} \\ normal, & \gamma_{FG} \leqslant \epsilon_{FG}, \end{cases} \quad (20b)$$

where ϵ_{FG} is a threshold used to determine if the foreground model vote is normal.

Similarly, we capture the posterior likelihood of the optical flow energy and HOF descriptors into binary mask $MASK_{OF}$, as follows:

$$\gamma_{OF} = -\log\left(\prod\{p_{OF}, p_{DIC}, p_{MRV}\}\right), \quad (21a)$$

$$MASK_{OF} = \begin{cases} anomalous, & \gamma_{OF} > \epsilon_{OF} \\ normal, & \gamma_{OF} \leqslant \epsilon_{OF}, \end{cases} \quad (21b)$$

where ϵ_{OF} is a threshold used to determine if the optical flow model vote is normal.

2) Second Phase - Mask Joint Analysis: in the second phase, we evaluate $MASK_{FG}$ and $MASK_{OF}$ using a joint criterion. Specifically, if a cell is deemed anomalous in any of the individual masks, then the corresponding frame at time t is marked as anomalous in that region using $MASK_t$, as follows:

$$MASK_t = MASK_{FG,t} \vee MASK_{OF,t}. \quad (22)$$

In order to make the inference mechanism more resilient to noise, we use the two consecutive frames at times $\{t, t+1\}$ to determine the abnormality of frame at time t, as follows [23]:

$$M\hat{A}SK_t = MASK_t \wedge MASK_{t+1}. \quad (23)$$

The binary mask $M\hat{A}SK_t$ then represents the abnormal regions in the current frame as the combination of the models of foreground occupancy and optical flow (see Fig. 6). Note that one of the advantages of using variable-sized cells is that the anomaly inference mechanism can locate abnormal

a) Input frame **b)** Foreground occupancy likelihood map **c)** Optical flow likelihood map **d)** Labeled frame
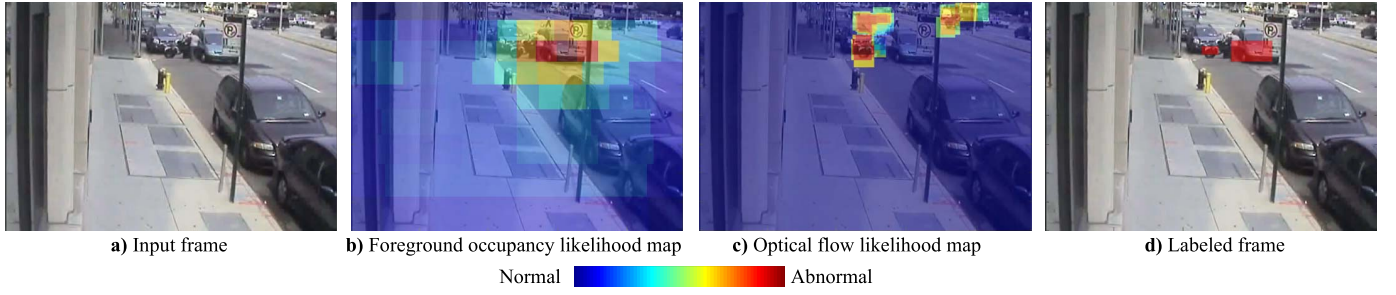
Normal ▬▬▬▬ Abnormal

Fig. 6. Anomaly inference mechanism. From (a) incoming frames, the posterior likelihood models are evaluated by thresholding the (b) foreground occupancy likelihood map and the (c) optical flow likelihood map. (d) The frame is labelled by evaluating consecutive frames.

Fig. 7. Example frames of the UMN dataset (first column), the UCSD dataset (columns 2 and 3) and the Subway dataset (fourth column). Abnormal regions are highlighted by boxes.

regions at different levels of spatial granularity according to their position relative to the camera. In the example shown in Fig. 6, the exact anomalous regions depicting a robbery is detected. The region occupied by the anomalous vehicle is much smaller than those regions occupied by other vehicles closer to the camera.

We finish this section with discussions about the suitability of our framework for online processing. Our framework generates a limited number of features, thus reducing computational times. Specifically, it generates foreground occupancy features (see Eq. 6) only for those video volumes whose associated cells are considered as active, thus considerably reducing the number of encoded features of this type. The number of encoded foreground occupancy features may be as low as zero if no activity is detected in the scene. This is an important advantage compared to other methods, e.g., [3], [24], that densely extract features from the entire sequence regardless of the activity in the scene. Similarly, the number of generated optical flow features is limited by the number of strongest STIPs detected by FAST. In this work, we limit the number of FAST STIPs to the 40 strongest detections. This also considerably reduces the overall number of encoded features of this type. Moreover, we create dictionaries for only those cells where HOF descriptors are found, thus limiting the number of dictionaries to be processed. In Section IV-E, we show that extracting features is one of the most time consuming steps. When classifying the current frame (see Eqs. 20-21), the posterior function is evaluated

considerably fast due to the linear complexity of the associated normal distribution, $\mathcal{N}$. The FSMC model (see Eq. 18) is a simple memory access procedure where the label given by dictionary matching gives the matrix index (i, j) (see Eq. 17), thus the associated computational complexity is very low. The generation of the final mask (see Eqs. 22-23) only involves evaluating two binary masks representing the vote given by the models. Both masks can be evaluated remarkably fast due to their binary nature and the fact that only AND/OR operations are required. Overall, our framework is designed to detect anomalous events in the shortest time possible, making it suitable for online processing. This will be further confirmed in the next Section.

IV. PERFORMANCE EVALUATION

A. Datasets

The UMN [5], USCD [8] and Subway [4] datasets are used for performance evaluations. In order to evaluate our framework on realistic video surveillance data, we also use a new collection of realistic videos captured by surveillance cameras with challenging events to be detected. This dataset is hereinafter referred to as the Live Videos dataset (LV dataset). We summarize these datasets next.

1) UMN Dataset: 11 video sequences depicting people walking in random directions and suddenly simulating panic (see Fig. 7). Videos are captured in three different scenarios with no camera motion and insignificant illumination

Fig. 8. Example frames of the LV dataset. Abnormal events are highlighted by boxes. a) Wrong Way Sequence: traffic goes in a single direction; suddenly men come out of three cars and people start running; one of them starts shooting (top right corner of frame) and traffic slows down; motorcycles start circulating in the wrong way. b) Robbery Sequence: a security guard is at the exit of a supermarket; costumers exit the scene after payment; three armed men suddenly brake into the supermarket; they force some costumers to lie on the floor and beat one of the cashiers. c) Panic Sequence: video surveillance of costumers in a convenience store; suddenly (captured by another surveillance camera) four armed men enter the store and costumers start running; some costumers try to hide first and later escape through the exit. d) Traffic Accident Sequence: a two-way road with sidewalks, where pedestrians are walking; a truck crashes into a house hitting a car and a light pole.

changes. Specifically, it comprises two outdoor scenarios with good illumination and one indoor scenario with poor illumination. The videos are captured at 30 FPS. Ground truth of the instants of time when the abnormal events occur is provided; however, no ground truth of the specific abnormal regions is provided.

2) UCSD Dataset: 96 sequences with different crowd densities where the abnormal events correspond to the presence of non-pedestrians entities on a sidewalk (see Fig. 7). Videos are captured with no changes in illumination or camera motion from two different perspectives overlooking two different sidewalks, resulting in two different scenes: the Peds1 and Peds2 scenes. The ground truth provided allows evaluation at

the frame and pixel-levels. For scene Peds1, we use 36 videos for testing and 34 videos for training. For scene Peds2, we use 16 videos for testing and 12 videos for training.

3) Subway Dataset: two scenes from the entrance and exit of a subway station. Three actors perform unusual activities which include entering without payment, wrong-way direction, loitering and irregular interactions. (see Fig. 7.)

4) LV Dataset: 28 realistic sequences of different frame sizes captured at different frame rates in indoors and outdoors scenarios with several illumination changes and some camera motion. All sequences are captured by surveillance cameras (see Fig. 8 for some examples). The videos depict various crowd densities, from empty scenes to the presence of

TABLE I
MAIN CHARACTERISTICS OF THE NEW LV DATASET

Duration	3.73 hours
Frame rate	7.5 - 30 fps
Resolution	minimum: QCIF (176×144) maximum: HDTV 720 (1280×720)
Format	Video MP4 in H.264
No. of videos	28
Dataset and ROI-level ground truth URL	https://cvrleyva.wordpress.com/
Anomalous frames	69996
Events of interest	32
Scenarios	Outdoor and indoor, uncontrolled environments, streets, highways, traffic intersections and public areas.
Synthetic sequences	None
Crowd density	Scenes with no subjects to very crowded scenes

TABLE II
MOST IMPORTANT PARAMETERS OF THE PROPOSED FRAMEWORK

Parameter	Description	Recommended values
α	Cell size growing rate	$1.06 - 1.2$
ϵ_{FG}	Anomaly inference foreground occupancy model threshold	$80 - 125$
ϵ_{OF}	Anomaly inference optical flow model threshold	$5.5 - 8.5$

thousands of people. Anomalous events last from a couple of frames to thousands of frames. All videos comprise a number of test and training frames. Ground-truth at the region of interest (ROI)-level is provided as a separate sequence of binary masks. No pixel-level ground truth is provided as this type of ground truth is usually very challenging to determine in realistic videos if the abnormal regions contain both foreground and background pixels. As discussed in [8], a method might correctly classify a whole frame as abnormal by incorrectly detecting any region where no abnormal event actually happens. In this case, the system is just *lucky* as the frame is classified as abnormal without correctly detecting the abnormal event. Evaluations at the ROI-level, using the appropriate ground truth, can therefore avoid this situation. The main characteristics of the LV dataset are summarized in Table I.

B. Experimental Setup

We have evaluated our framework against a number of state-of-art video anomaly detection methods [3], [4], [12]–[16], [19], [21]–[23], [43]. In this work, a method is considered to be suitable for online processing if frames are processed within the FPS rate. For example, for a 30 FPS sequence, frame processing times should be shorter than 33 ms; i.e., (1/30s). Based on this criterion, a very limited number of methods can be considered as being able to attain online performance. Among these, the method proposed by Lu *et al.* in [21] and that proposed by Biswas and Babu in [22] are among the state-of-the-art.

To extract the foreground of the sequences, we use the implementation of Alekhin [44]. For the UMN and UCSD datasets, we learn the background from 200 frames using a learning rate of 10^{-2}. During testing, the learning rate is set to 10^{-3}. For the LV dataset, we use the same learning rate parameter of 10^{-2} considering 300 frames. To evaluate the proposed method of Lu *et al.* [21], we use the code available in [45] with the parameters suggested in the demo code section. To evaluate the method of Biswas and Babu [22],

we use the code available in [46] to estimate the interpolation steps; a maximum of five GMM model components are used. The other parameters are kept as proposed by the authors.

For our framework, we empirically determine the value of parameters α, which is the cell growing rate in the proposed cell structure, and ϵ_{FG} and ϵ_{OF}, which are the thresholds in Eq. 20-21. To this end, we evaluate the effect of these parameters on our frameworks' performance, at the pixel-level, using the Peds1 scene of the UCSD dataset. This particular scene contains relatively small frames with challenging abnormal events to be detected. Therefore, this scene can be used to determine values for α, ϵ_{FG} and ϵ_{OF} that are appropriate for the other tested datasets. To determine the value of α, the pixel-level Receiver Operating Characteristics (ROC) curve is computed using different values of α (see Fig. 9a). The ROC curve corresponds to the Equal Error Rate (EER) over the Area Under the Curve (AUC) of the evaluated method, i.e., EER/AUC, and denotes its precision in terms of its sensitivity. As $\alpha \to 1$, all cells tend to have the same initial size, y_0. If $\alpha \gg 1$, cells tend to increase in size starting from position ($x = X/2, y = 0$) in the frame, towards $x = 0$, $x = X$ and $y = Y$, where X and Y denote the vertical and horizontal dimensions of the frame, respectively. Details about the construction of the cell structure may be found in the Appendix. Results in Fig. 9a show that for values of α close to one, i.e., $\alpha = 1.02$, the AUC is reduced. This trend is also evident for large values of α, i.e., $\alpha = 1.3$. For this scene, we can observe that $\alpha = 1.06$ provides the largest AUC. We thus use $\alpha \in [1.06, 1.2]$ in our experiments.

From Fig. 9b, we observe that tuning ϵ_{OF} has a more profound impact on pixel-level EER than tuning ϵ_{FG}. Note that ϵ_{FG} values above 80 in conjunction with ϵ_{OF} values above 6.5 attain the lowest pixel-level EERs. We have thus set $\epsilon_{FG} = 80$ and $\epsilon_{OF} = 6.5$ in our experiments to accommodate for the different characteristics of the evaluated datasets. Table II summarizes the most important parameters of our framework and the recommended range of values.

Some of the sequences in the LV dataset contain very large frames, e.g. HD 1200×720. In order to reduce complexity, we scale the frames by sub-sampling to a fixed size of 160×240.

Our framework is initialized as follows:

1) The cell structure is built according to α and an initial size y_0 (see Appendix). This structure is used to determine the spatio-temporal regions from which features are extracted.
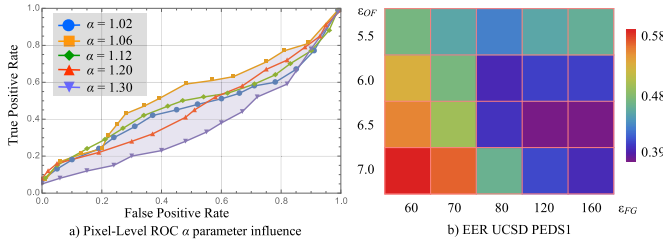
Fig. 9. Pixel-level Equal Error Rate (EER) for scene Peds1 of the UCSD dataset using different values for parameters a) α and b) ϵ_{FG} and ϵ_{OF} .

TABLE III
AUC VALUES FOR THE UMN DATASET

Authors	AUC	Frame Processing Time	On-line Performance
Hu *et al.* [14] †	0.977	200 ms	
Li *et al.* [43]	0.996	1100 ms	
Cong *et al.* [50]	0.973	3800 ms	
Zhu *et al.* [19]	0.997	4600 ms	
Lu [21]	0.701	**6 ms**	✓
Biswas and Babu [22]	0.736	14 ms	✓
Ours	**0.883**	31 ms	✓

† After optical flow and histogram calculations.

2) Features are extracted from the training frames and stored.
3) Extracted features are processed to generate the GMMs, dictionaries, and the FSMC.
4) After training, the framework has all models required to start inferring abnormal events.

C. Performance Metrics

For the UMN, results are reported in terms of the AUC. For this dataset, we use the provided top corner labels as ground truth. Since this dataset provides no pixel-level ground truth, for our framework we classify the whole frame as abnormal if at least one region is classified as abnormal. This is the same criterion used in the other evaluated methods. It is important to note that the AUC and EER are similar metrics of a methods's performance. Specifically, $AUC \rightarrow 1$ when $EER \rightarrow 0$.

For the UCSD dataset, results are reported in terms of the EER at the frame and pixel-levels. For this dataset, a frame is deemed to be correctly classified if at least 40% of the pixels are correctly classified [8], [47]. This is the criterion used in all evaluated methods. The masks provided in [48], [49] provide the pixel-level ground truth. For the Subway dataset, we rank a method by counting the number of events that are detected in each scene. For the LV dataset, results are reported in terms of the ROC curve. Here, a frame is deemed to be correctly detected as abnormal when at least 20% of the ROI is detected. The corresponding event is thus classified as a true positive.

D. Results

Table III tabulates average AUC values for the UMN dataset. Results for the compared methods are tabulated as reported in the corresponding referred publication. Note that

TABLE IV
EER FOR THE PEDS1 SCENE OF THE UCSD DATASET

Authors	EER Frame Level	EER Pixel Level	Frame Processing Time	On-line Performance
Javan and Levine [3]	15	27	190 ms	
Hu *et al.* [14] †	18	36	200 ms	
Cheng *et al.* [15]	19.9	38.8	1100 ms	
Cong *et al.* [50]	23	51.2	3800 ms	
Zhu *et al.* [19]	15	–	4600 ms	
Lu *et al.* [21]	**15**	59.1	**6 ms**	✓
Biswas and Babu [22]	24.66	50.95	14 ms	✓
Ours	21.15	**39.7**	31 ms	✓

† After optical flow and histogram calculations.

TABLE V
EER FOR THE PEDS2 SCENE OF THE UCSD DATASET

Authors	EER Frame Level	EER Pixel Level	Frame Processing Time	On-line Performance
Javan and Levine [3]	13	26	220 ms	
Hu *et al.* [14] †	15	–	200 ms	
Li *et al.* [43]	18.5	–	1100 ms	
Lu *et al.* [21]	22.3	49.8	**6.1 ms**	✓
Biswas and Babu [22]	29.6	42.3	12.5 ms	✓
Ours	**19.2**	**36.6**	31 ms	✓

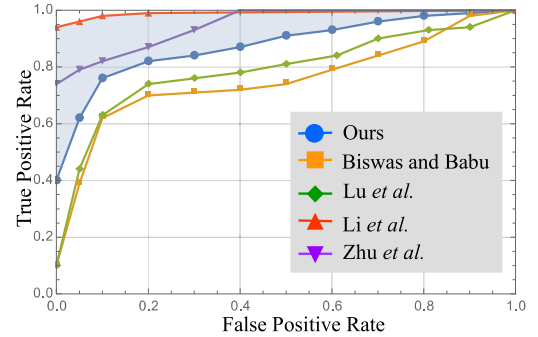† After optical flow and histogram calculations.

Fig. 10. ROC curve for the UMN dataset.

our framework attains very competitive results compared to non-online methods, which are expected to outperform online methods. Compared to online methods, our framework attains the highest AUC values. We also report the frame-level ROC curve for this dataset. These results are shown in Fig. 10. It can be observed that our framework outperforms the online method proposed by Biswas and Babu in [22] and that proposed by Lu *et al.* in [21]. Our framework also attains very competitive results compared to other non-online methods that have been designed for performance, and not processing times, like the one proposed by Zhu *et al.* [19].

Results for scenes Peds1 and Peds2 of the UCSD dataset are tabulated in Tables IV and V, respectively. Results for the other compared methods are tabulated as reported in the corresponding referred publication. As expected, non-online methods tend to attain the lowest EER values at both the frame and pixel-levels. However, their frame processing times are considerably long. For example, the best reported non-online method,
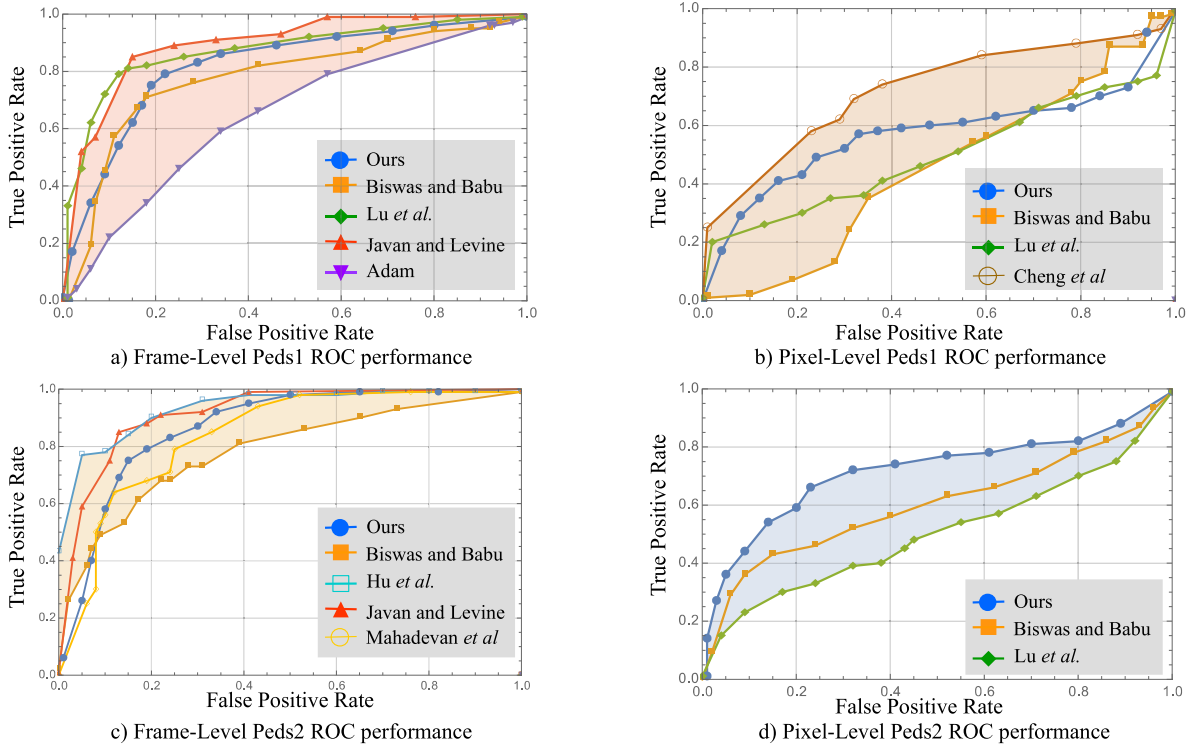
Fig. 11. ROC Curves for the UCSD dataset at the frame- and pixel-level. (a)-(b) Results for Peds1 scene. (c)-(d) Results for Peds2 scene.

i.e., the method in [3], attains a frame processing time six times longer than that attained by our framework. This very long frame processing time is mainly due to the dense multi-scale sampling used, which is known to be computationally complex. Our framework attains a pixel-level EER about 11% lower than that attained by the online method of Biswas and Babu in [22]. For scene Peds2, our proposed framework achieves a better performance at the frame-level than that attained by the online method in [22]. We also report the ROC curves for the UCSD datasets in Fig. 11. From this Fig., one can observe that our framework achieves a very competitive performance compared to non-online methods. Specifically, at the frame-level, our framework attains results very similar to many of the best performing non-online methods (see Fig. 11 (a) and (c)). Our framework's results are comparable with methods 10 to 20 times slower, e.g., Cheng *et al.*'s method in [15], which have been designed for performance and not for processing times. At the pixel-level, our framework achieves also a competitive performance compared to non-online methods, and significantly outperforms online methods (see Fig. 11 (b) and (d)). Overall, our framework achieves a very competitive performance on the UCSD dataset compared to non-online methods, while outperforming some of the online methods.

Results for the Subway dataset are tabulated in Table VI. Results in this Table are reported following the convention for this dataset, i.e., we report the number of detected events by a method for the Entrance/Exit scenes, for each type of anomalous event. The first row indicates the number events to be detected (ground truth of the dataset). For example, for the type of anomalous events *Wrong Direction* (WD), the ground

TABLE VI

NUMBER OF EVENTS DETECTED FOR THE ENTRANCE/EXIT SCENE OF THE SUBWAY DATASET FOR DIFFERENT TYPES OF ANOMALOUS EVENTS: WRONG DIRECTION (WD), NO PAYMENT (NP), LOITERING (LT), IRREGULAR INTERACTION (II), MISCELLANEOUS (MISC) AND FALSE ALARM (FA)

Authors	WD	NP	LT	II	MISC	FA	On-line Performance
Ground Truth	26/9	13/0	14/3	4/0	9/7	0/0	
Hu *et al.* [14] †	**26/9**	6/0	14/3	4/0	8/7	6/2	
Zhao *et al.* [26]	25/9	9/0	14/3	4/0	**9/7**	5/2	
Biswas and Babu [22]	24/8	5/0	6/2	2/0	5/3	14/10	✓
Lu *et al.* [21]	25/9	7/0	13/3	4/0	8/7	**4/2**	✓
Ours	21/6	**9/0**	8/3	2/0	4/2	12/7	✓

† After optical flow and histogram calculations.

truth indicates that they are 26 of such anomalous events in the Entrance scene and 9 in the Exit scene. This is indicated as 26/9.

From Table VI, one can observe that our framework achieves a competitive accuracy compared to other online methods. It particularly outperforms other online methods for the No Payment (NP) type of events, i.e., our framework is able to detect 9 out of the 13 events. It is important to note that these NP events are the most important ones in this dataset, and correctly detecting them is one of the main motivations behind this dataset. For the Wrong Direction (WD) type of events, our framework also attains a competitive accuracy, very close to the best performing non-online methods, which have been designed specifically to attain a high detection accuracy.

Results for the LV dataset are plotted in Fig. 12 for our framework, the online method of Biswas and Babu [22] and that of Lu *et al.* [21]. We can see that our framework is
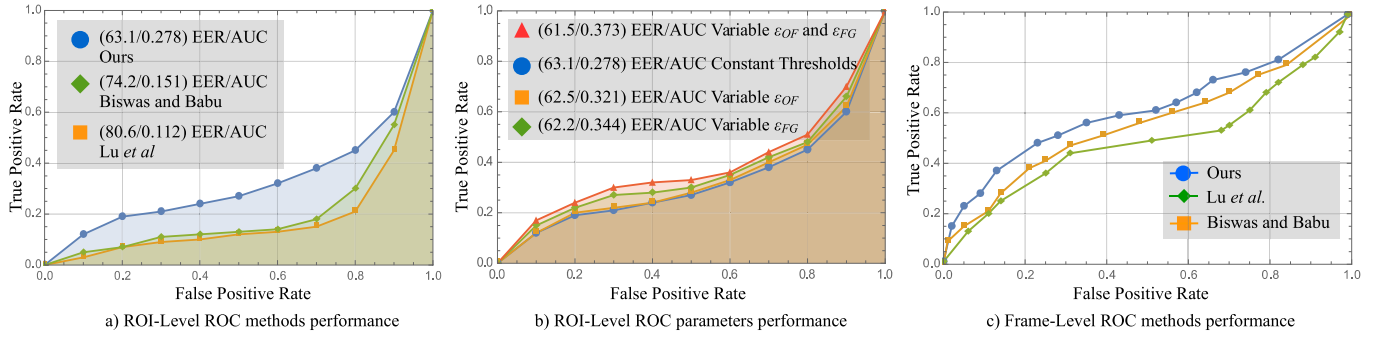
a) ROI-Level ROC methods performance   b) ROI-Level ROC parameters performance   c) Frame-Level ROC methods performance

Fig. 12.   ROC curves of compared online methods for the LV dataset. a) Our framework is evaluated with constant threshold values ϵ_{FG} and ϵ_{OF}. b) ROC curves of the proposed framework for the LV dataset when ϵ_{FG} and ϵ_{OF} values are modified. c) ROC curve LV dataset using a frame-level criterion.
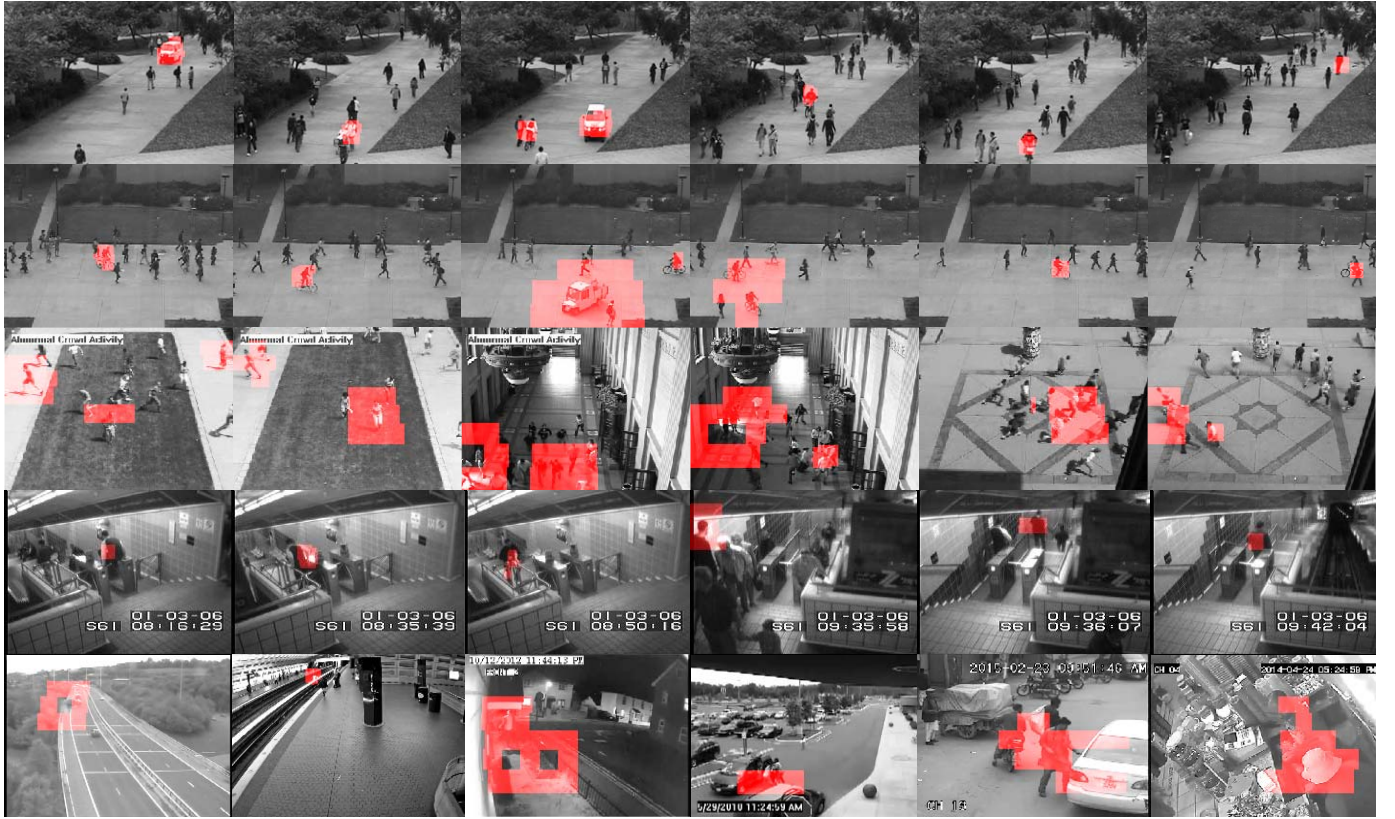

Fig. 13.   Example frames showing the anomalous events detected by our framework. **1st Row.** UCSD Peds1: a man with a trolley, cyclists, small cars and skaters. **2nd Row.** UCSD Peds2: small cars and cyclists. **3th Row.** UMN: people in panic at the moment when they start to run. **4th Row.** Subway: (left to right) Entrance scene showing people entering without payment and walking in the wrong direction. **5th Row.** LV: (from left to right) a lorry hitting a car and capsizing in a highway; a man in a wheelchair falling into the subway tracks; a man destroying private property; a woman being kidnapped outside a shopping mall; an armed robbery; a cashier being beaten by burglars.

significantly better than the evaluated online methods. Specifically, the attained EER is nearly 10%-18% lower than that attained by the online methods in [21] and [22]. It is important to note that the ROC curves in Fig. 12a) are below the $y = x$ straight line. This is because we are counting as true positives only those cases when a method successfully detects the ROI depicting the abnormal event within a frame. If the method fails to detect this ROI and detects other region, we count the detection as a false negative. Consequently, this criterion allows us to determine if a method is capable of detecting exactly the region of the scene where abnormal events happen.

Alternatively, we can label the whole frame as abnormal whenever any region is detected as abnormal in an abnormal frame, i.e., following a frame-level criterion. This evidently increases AUC values, but prevents measuring if the method is capable of detecting the exact regions that generate the anomaly. In order to have the most complete set of results, we also plot in Fig. 12c) ROC curves using such a frame-level criterion. It can be seen that the ROC curves now approach the $y = x$ straight line, as expected. Our proposed framework also attains the best performance based on this frame-level criterion.

TABLE VII
FRAME PROCESSING TIMES AND AUC VALUES OF
ONLINE METHODS FOR THE LV DATASET

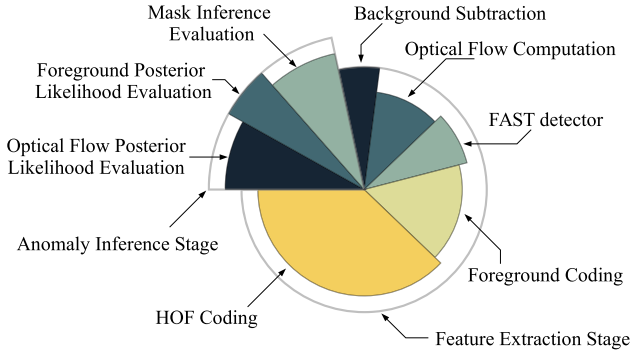| Authors	AUC	Frame Processing Time
Lu *et al.* [21]	0.112	**6.8 ms**
Biswas and Babu [22]	0.151	13.2 ms
Ours	**0.278**	32.5 ms

Fig. 14. Required time by various processes of our proposed framework during the Detection Stage for a single frame.

Table VII tabulates frame processing times and AUC values for the LV dataset. From this Table, one can observe that our framework attains the highest AUC values and meets online performance for 30 FPS videos, which is the highest frame rate in the LV dataset. Although the other tested online methods are capable of attaining shorter frame processing times for this dataset, it is important to note that their AUC values are close to 50% lower than that attained by our framework. The shorter frame processing times attained by Lu *et al.*'s and Biswas and Babu's methods are mainly due to the fact that these methods do not employ optical flow nor background subtraction to collect motion features. They instead use simple temporal gradients and the motion vectors associated with the compressed video sequences. This inevitably decreases frame processing times, but sacrifices detection performance.

We have also evaluated the effect on the ROC curve of the LV dataset when ϵ_{FG} and ϵ_{OF} are varied. Specifically, we have modified the optical flow model threshold (ϵ_{OF}), while keeping the foreground occupancy model threshold fixed ($\epsilon_{FG} = 6.5$). We have also evaluated the case of modifying ϵ_{FG}, while keeping $\epsilon_{OF} = 80$ fixed, and the case of modifying both thresholds. These thresholds are modified using the range of values plotted in Fig. 9b). The values that provide the highest detection accuracy for each video sequence is selected. Results of this evaluation are shown in Fig. 12b). As expected, tailoring both thresholds for each sequence provides the best performance (see red curve in Fig. 12b)). It is interesting to note that tailoring ϵ_{FG} while keeping ϵ_{OF} fixed provides a better performance than tailoring ϵ_{OF} while keeping ϵ_{FG} fixed (see green curve vs. yellow curve in Fig. 12b)). This is mainly because illumination changes are more drastic than camera motion for the tested dataset. Thus adjusting the ϵ_{FG} threshold has a more direct impact on the framework's performance. Adjusting thresholds in our framework is a way of specifying how much the models are to be trusted to

efficiently describe a particular event. Specifically, thresholds ϵ_{FG} and ϵ_{OF} represent trust levels that indicate how much one can trust the model associated with foreground occupancy and optical flow features, respectively. If one wishes to minimize the effect associated with a particular model's inference, the corresponding threshold should be set to a high value. In this case, that particular model is not *trusted*, and the overall inference mechanism mostly depends on the other *trusted* model's inference. Therefore, our framework is flexible in this regard, as it can be adapted according to scene characteristics, if these are known *a priori*. Example frames showing the anomalous events detected by our framework are depicted in Fig. 13.

E. Discussions

1) Accuracy: Detecting abnormal events in realistic scenes is challenging. However, our proposed framework is capable of detecting challenging abnormal events outperforming other online methods. For example, let us take the frame in row 5, column 1 of Fig. 13, which depicts a car accident. In this sequence, our framework is able to detect most of the frames depicting the accident. The evaluated online methods in [21], [22] are not able to detect this event. The main reason for the poor performance of these two other online methods on this sequence is the fact that moving objects tend to slow down when the abnormal event occurs. Consequently, the frame differences, which are the core of both methods, cannot provide features from the region where the accident takes place. This sequence is also a good example to showcase the advantages of the variable-sized cell structure, where small cells are defined in the region depicting the abnormal event. Therefore, our framework can accurately detect the ROI depicting the car accident. Another example that demonstrates the advantages of our proposed cell structure is the frame in row 5, column 2 of Fig. 13, which depicts a man in a wheelchair falling into the subway tracks. In this case, the region where this abnormal event takes place is very far from the camera, thus the ROI to be detected is very small. The proposed coarse-to-fine cells help to accurately detect this event as the region is described by enough features at the correct size. The two other evaluated online methods also fail to detect this particular event. Let us take now the frame in row 5, column 3 of Fig. 13. In this case the abnormal event corresponds to a man breaking into private property and causing some damage to it. The scene is poorly illuminated and consequently features based on STIPs are expected to perform poorly. In this case, our framework profits from the fact that two sources of features are available; those from foreground occupancy information and those from optical flow. Even if the features from optical flow are not descriptive enough, those from foreground occupancy help our framework to correctly detect this event. For this particular scene, the other evaluated online methods fail to detect this ROI, and instead, they incorrectly detect other regions as abnormal.

2) Time Performance: Our proposed framework is implemented in MATLAB and tested on a 2.7GHz CPU with 8GB of RAM. Our full end-to-end MATLAB implementation is
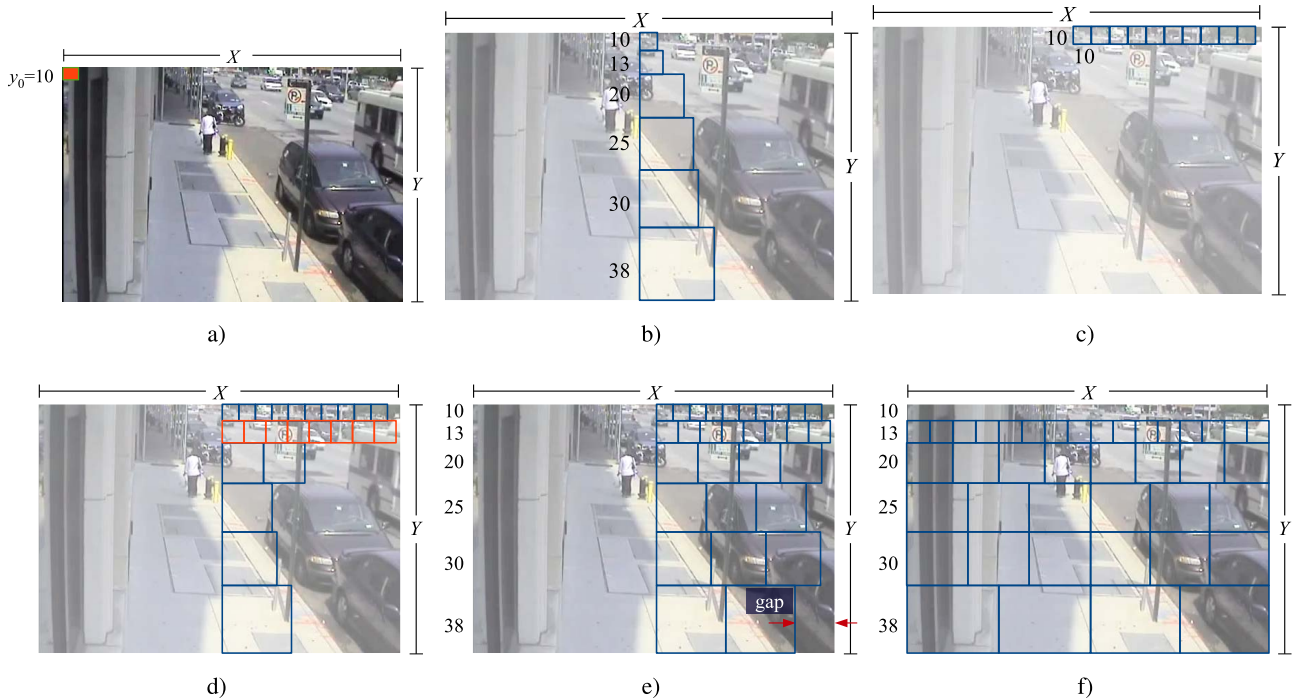
Fig. 15. Example of cell structure generation. (a) Define the size of the smallest square cell in the structure. (b) Populate cells in the vertical direction. (c) Populate cells in the horizontal direction, starting from $X/2$. (d) Repeat horizontal population for $\{13, 20, \ldots\}$. (e) Square cells after full population. (f) Adjusted cells to completely fit the horizontal dimension.

available in https://cvrleyva.wordpress.com/. The code is not parallelized and no GPU arrays are employed to speed up the computations. Fig. 14 shows the proportion of time required by various processes of our framework during the Detection Stage for a single frame. It can be seen that encoding HOF descriptors is the most expensive step. This is mainly because the framework has to calculate every orientation of each pixel in the spatio-temporal support regions defined for the FAST STIPs. Note that the processing times of the likelihood modeling are much lower than those of the feature extraction process. This is mainly because our framework only extracts features for a limited number of support regions and strongest detected FAST STIPs. This significantly reduces the total number of features to be encoded and processed. This is the main aspect of our framework that helps to reduce overall computational times.

During the evaluation, we observed that there usually is a trade-off between computational times and detection accuracy. However, our framework manages to attain high detection accuracies while achieving online performance thanks to the compact set of features and the models used to efficiently process them. Specifically, our framework outperforms online methods, while being very competitive among non-online methods.

As part of the evaluation, we also showed that our framework is flexible to be tailored to the characteristics of the sequences, if these are known *a priori*, in order to improve performance. Our future work is aimed at further enhancing our framework's detection accuracy by exploiting this flexibility; specifically, by considering the optimization of our framework's parameters given a particular set of environmental conditions used to capture a sequence.

V. CONCLUSIONS

In this paper, we proposed an online framework for video anomaly detection. Our framework extracts a compact set of features based on foreground occupancy and optical flow information. The framework employs a novel variable-sized cell structure which allows extracting features from a limited number of different support regions in a fine-to-coarse fashion. This helps to process a significantly smaller number of features than those processed by dense-scanning based methods. We evaluated our framework on the popular UMN, UCSD and Subway datasets, as well on the LV dataset, which is a new collection of realistic sequences captured by surveillance cameras under challenging environmental conditions.

APPENDIX
CONSTRUCTION OF THE CELL STRUCTURE

1) Define $y_0 > 0$ (i.e., size of the smallest square cell) and $\alpha > 1$ (i.e., growing rate of the cell size). See Fig. 15a).

2) Adjust y_0 to $\hat{y}_0$ in order to fit an integer n number of square cells across the vertical dimension Y of the frame:

$$n = \lfloor \log_\alpha \left(Y/y_0 (\alpha - 1) + 1 \right) - 1 \rfloor, \qquad (24)$$

and

$$\hat{y}_0 = \left\lfloor \frac{\alpha - 1}{\alpha^{n+1} - 1} Y \right\rfloor, \qquad (25)$$

3) Calculate the size of the n square cells to be created across the vertical dimension Y using the recursive

equation $y_{k+1} = \alpha y_k$. For instance, for the set of parameters $\{\hat{y}_0 = 10, \alpha = 1.25\}$, and a vertical dimension $Y = 160$, this recursive equation generates $n = 6$ cells of increasing sizes $\{10, 13, 20, 25, 30, 38\}$ (see Fig. 15b)).

4) Starting at $X/2$, i.e., the mid pint of the frame along the horizontal dimension X, populate an integer number of square cells across the X dimension, as illustrated in Fig. 15c). Repeat the same process for the remaining sizes computed in step 3) (see Fig. 15d)). In our example these sizes are $\{13, 20, 25, 30, 38\}$.

5) Fill in any horizontal gaps in order to completely cover the frame in the horizontal dimension from $X/2$ to X. This is done by adding one pixel to the horizontal dimension of the cells populated in step 4) until the cells completely cover the frame from $X/2$ to X (see Fig. 15e)). Note that due to this adjustment in the horizontal size of the cells, the final cells may not be square.

6) Cover the other half of the frame using the cell sizes computed in step 5) (see 15f)).

7) The first row of cells comprises the smallest cells. Our experiments show that false alarms are often triggered in this first row of cells. Based on this observation, we discard the first row from the structure (see Fig. 15f)).

REFERENCES

[1] C. Piciarelli and G. Foresti, "Surveillance-oriented event detection in video streams," *IEEE Intell. Syst.*, vol. 26, no. 3, pp. 32–41, May 2011.

[2] A. A. Sodemann, M. P. Ross, and B. J. Borghetti, "A review of anomaly detection in automated surveillance," *IEEE Trans. Syst., Man, Cybern., C (Appl. Rev.)*, vol. 42, no. 6, pp. 1257–1272, Nov. 2012.

[3] M. J. Roshtkhari and M. D. Levine, "An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions," *Comput. Vis. Image Understand.*, vol. 117, no. 10, pp. 1436–1452, 2013.

[4] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 555–560, Mar. 2008.

[5] *Univeristy of Minnesota Dataset*, accessed on Feb. 1, 2016. [Online]. Available: http://mha.cs.umn.edu/proj_events.shtml

[6] O. Boiman and M. Irani, "Detecting irregularities in images and in video," *Int. J. Comput. Vis.*, vol. 74, pp. 17–31, Aug. 2007.

[7] A. Zaharescu and R. Wildes, "Anomalous behaviour detection using spatiotemporal oriented energies, subset inclusion histogram comparison and event-driven processing," in *Computer Vision—ECCV 2010* (Lecture Notes in Computer Science), vol. 6311, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Germany: Springer, 2010, pp. 563–576.

[8] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1975–1981.

[9] S. Ali and M. Shah, "A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Minneapolis, MN, USA, Jun. 2007, pp. 1–6.

[10] C. C. Loy, T. Xiang, and S. Gong, "Modelling multi-object activity by Gaussian processes," in *Proc. BMVC*, 2009, pp. 1–11.

[11] M. J. Roshtkhari and M. D. Levine, "Online dominant and anomalous behavior detection in videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2611–2618.

[12] Y. Cong, J. Yuan, and Y. Tang, "Video anomaly search in crowded scenes via spatio-temporal motion context," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 10, pp. 1590–1599, Oct. 2013.

[13] M. Bertini, A. D. Bimbo, and L. Seidenari, "Multi-scale and real-time non-parametric approach for anomaly detection and localization," *Comput. Vis. Image Understand.*, vol. 116, no. 3, pp. 320–329, 2012.

[14] Y. Hu, Y. Zhang, and L. S. Davis, "Unsupervised abnormal crowd activity detection using semiparametric scan statistic," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2013, pp. 767–774.

[15] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, "Gaussian process regression-based video anomaly detection and localization with hierarchical feature representation," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5288–5301, Dec. 2015.

[16] K.-W. Cheng, Y.-T. Chen, and W.-H. Fang, "Video anomaly detection and localization using hierarchical feature representation and Gaussian process regression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2909–2917.

[17] V. Saligrama and Z. Chen, "Video anomaly detection based on local statistical aggregates," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2112–2119.

[18] Z. Zhu, J. Wang, and N. Yu, "Anomaly detection via 3D-HOF and fast double sparse representation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 286–290.

[19] X. Zhu, J. Liu, J. Wang, C. Li, and H. Lu, "Sparse representation for robust abnormality detection in crowded scenes," *Pattern Recognit.*, vol. 47, no. 5, pp. 1791–1799, 2014.

[20] H. Mousavi, S. Mohammadi, A. Perina, R. Chellali, and V. Murino, "Analyzing tracklets for the detection of abnormal crowd behavior," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2015, pp. 148–155.

[21] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 FPS in MATLAB," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2720–2727.

[22] S. Biswas and R. V. Babu, "Real time anomaly detection in H.264 compressed videos," in *Proc. 4th Nat. Conf. Comput. Vis. Pattern Recognit. Image Process. Graph. (NCVPRIPG)*, Dec. 2013, pp. 1–4.

[23] V. Reddy, C. Sanderson, and B. C. Lovell, "Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2011, pp. 55–61.

[24] M. Bertini, A. D. Bimbo, and L. Seidenari, "Scene and crowd behaviour analysis with local space-time descriptors," in *Proc. 5th Int. Symp. Commun. Control Signal Process. (ISCCSP)*, May 2012, pp. 1–6.

[25] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2. Oct. 2005, pp. 1508–1515.

[26] B. Zhao, L. Fei-Fei, and E. P. Xing, "Online detection of unusual events in videos via dynamic sparse coding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 3313–3320.

[27] T. Xiao, C. Zhang, H. Zha, and F. Wei, "Anomaly detection via local coordinate factorization and spatio-temporal pyramid," in *Computer Vision—ACCV 2014* (Lecture Notes in Computer Science), vol. 9007, D. Cremers, I. Reid, H. Saito, and M.-H. Yang, Eds. Cham, Switzerland: Springer, 2015, pp. 66–82.

[28] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–7.

[29] R. Leyva, V. Sanchez, and C.-T. Li, "Video anomaly detection based on wake motion descriptors and perspective grids," in *Proc. IEEE Int. Workshop Inf. Forensics Security (WIFS)*, Dec. 2014, pp. 209–214.

[30] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, "Recent advances and trends in visual tracking: A review," *Neurocomputing*, vol. 74, no. 18, pp. 3823–3831, Nov. 2011.

[31] H. Fradi and J.-L. Dugelay, "Towards crowd density-aware video surveillance applications," *Inf. Fusion*, vol. 24, pp. 3–15, Jul. 2015.

[32] H. Guo, X. Wu, N. Li, R. Fu, G. Liang, and W. Feng, "Anomaly detection and localization in crowded scenes using short-term trajectories," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2013, pp. 245–249.

[33] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 393–400.

[34] P. L. St-Charles, G. A. Bilodeau, and R. Bergevin, "SuBSENSE: A universal change detection method with local adaptive sensitivity," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 359–373, Jan. 2015.

[35] S. E. Ebadi, V. G. Ones, and E. Izquierdo, "Efficient background subtraction with low-rank and sparse matrix decomposition," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 4863–4867.

[36] H. Ramadan and H. Tairi, "Automatic human segmentation in video using convex active contours," in *Proc. 13th Int. Conf. Comput. Graph. Imag. Vis. (CGiV)*, Mar. 2016, pp. 184–189.

[37] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-Based Surveillance Systems*. New York, NY, USA: Springer, 2002, pp. 135–144.

[38] I. Laptev and T. Lindeberg, "Space-time interest points," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 1. Oct. 2003, pp. 432–439.

[39] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Computer Vision—ECCV 2006* (Lecture Notes in Computer Science), vol. 3952, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Germany: Springer, 2006, pp. 428–441.

[40] L. Kratz and K. Nishino, "Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 1446–1453.

[41] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Sep. 2006, pp. 2169–2178.

[42] Y. Benezeth, P.-M. Jodoin, V. Saligrama, and C. Rosenberger, "Abnormal events detection based on spatio-temporal co-occurences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 2458–2465.

[43] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 18–32, Jan. 2014.

[44] *Opencv MATLAB Code Generator*, accessed on Feb. 1, 2016. [Online]. Available: https://github.com/Itseez/opencv_contrib/tree/master/modules/matlab

[45] *Jianping Shi Site*, accessed on Feb. 1, 2016. [Online]. Available: http://shijianping.me/

[46] *Kernel Density Estimator*, accessed on Feb. 1, 2016. [Online]. Available: http://uk.mathworks.com/matlabcentral/fileexchange/14034-kernel-density%-estimator

[47] *Ucsd Anomaly Detection Dataset*, accessed on Feb. 1, 2016. [Online]. Available: http://www.svcl.ucsd.edu/projects/anomaly/dataset.htm

[48] B. Antic and B. Ommer, "Video parsing for abnormality detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2415–2422.

[49] *Video Parsing for Abnormality Detection*, accessed on Feb. 1, 2016. [Online]. Available: http://hciweb.iwr.uni-heidelberg.de/compvis/research/parsing

[50] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 3449–3456.

Victor Sanchez received the M.Sc. degree from the University of Alberta, Canada, in 2003, and the Ph.D. degree from The University of British Columbia, Canada, in 2010. From 2011 to 2012, he was with the Video and Image Processing Laboratory, University of California at Berkeley, Berkeley, as a Post-Doctoral Researcher. In 2012, he was a Visiting Lecturer with the Group on Interactive Coding of Images, Universitat Autnoma de Barcelona. He is currently an Associate Professor with the Department of Computer Science, University of Warwick, U.K. His current research interests are in the area of signal and information processing with applications to multimedia analysis, security, image, and video coding and communications. He has authored several technical papers in these areas and co-authored a book (Springer, 2012). His research has been funded by Consejo Nacional de Ciencia y Tecnologia, Mexico, the Natural Sciences and Engineering Research Council of Canada, the Canadian Institutes of Health Research, the FP7 and H2020 programs of the European Union, and the Engineering and Physical Sciences Research Council, U.K.

Roberto Leyva received the M.Sc. degree from the National Polytechnic Institute in 2013. He is currently pursuing the Ph.D. degree with the Department of Computer Science, University of Warwick, U.K. His current research interests are in the area of computer vision and video understanding. His studies are funded by the Mexican Ministries of Education SEP-CONACYT and by the European Union EU H2020 project IDENTITY. He has published several technical papers in these areas.

Chang-Tsun Li received the B.Eng. degree in electrical engineering from National Defence University (NDU), Taiwan, in 1987, the M.Sc. degree in computer science from the U.S. Naval Postgraduate School, USA, in 1992, and the Ph.D. degree in computer science from the University of Warwick, U.K., in 1998. He was an Associate Professor of the Department of Electrical Engineering, NDU, from 1998 to 2002, and a Visiting Professor of the Department of Computer Science with the U.S. Naval Postgraduate School in 2001. He was a professor of the Department of Computer Science, University of Warwick, U.K., until 2016. He is currently a Professor of the School of Computing and Mathematics, Charles Sturt University, Australia. The outcomes of his multimedia forensics research have been translated into award-winning commercial products protected by a series of international patents and have been used by a number of police forces and courts of law around the world. His current research interests include multimedia forensics and security, biometrics, data mining, machine learning, data analytics, computer vision, image processing, pattern recognition, bioinformatics, and content-based image retrieval. He was involved in the organisation of many international conferences and workshops and also served as member of the international program committees for several international conferences. He is currently Associate Editor of the *EURASIP Journal of Image and Video Processing* and an Associate of Editor of *IET Biometrics*. He is also actively contributing keynote speeches and talks at various international events.