On-line Real-time Crowd Behavior Detection in Video Sequences

Andrea Pennisi[a], Domenico D. Bloisi[a,*], Luca Iocchi[a]

[a] *Department of Computer, Control, and Management Engineering*
Sapienza University of Rome
via Ariosto 25, 00185, Rome (Italy)

Abstract

Automatically detecting events in crowded scenes is a challenging task in Computer Vision. A number of off-line approaches have been proposed for solving the problem of crowd behavior detection, however the off-line assumption limits their application in real-world video surveillance systems. In this paper, we propose an on-line and real-time method for detecting events in crowded video sequences. The proposed approach is based on the combination of visual feature extraction and image segmentation and it works without the need of a training phase. A quantitative experimental evaluation has been carried out on multiple publicly available video sequences, containing data from various crowd scenarios and different types of events, to demonstrate the effectiveness of the approach.

Keywords: event detection, crowd analysis, image segmentation, intelligent surveillance

1. Introduction

Event detection in the field of automatic video surveillance has gained a growing interest [1]. The huge amount of data generated by existing surveillance systems in public areas requires the development of intelligent solutions that can avoid information overload for the users [2]. In particular, in the context of a crowd image analysis problem, it is desirable to develop on-line

[*] Corresponding author
Email address: bloisi@dis.uniroma1.it (Domenico D. Bloisi)
URL: http://www.dis.uniroma1.it/~bloisi/ (Domenico D. Bloisi)

algorithms that reliably detect abnormal events in real-time. As an example, the automatic detection of anomalies in crowded scenes can be used to avoid crowd related disasters and ensure public safety [3].

An anomaly can be defined as "*something that deviates from what is standard, normal, or expected*"[1]. This means that *abnormal events* can be identified as irregular situations with respect to usual normal ones. Thus, the abnormal detection becomes the identification of abnormal events given some sample normal events. Zhan et al. [4] point out that conventional Computer Vision can be ineffective when dealing with the analysis of very crowded video sequences. Indeed, in a high-density situation the presence of severe occlusions consistently limits the performance of traditional methods for visual tracking [3]. Additional factors that can limit the effectiveness of existing approaches aiming at detecting abnormal events are: 1) Off-line computation and 2) Need of a training phase. The off-line assumption can limit the application of anomaly detection methods in practice [5]. For instance, it is desirable to detect panic situations as soon as possible in order to avoid damage to people. The methods that rely on the training of a classifier are limited by the possible lack of well-suited training data. Indeed, since it is not easy to find data about real emergency situations in crowded scenes, the resulting classifier could be suitable only for dealing with particular video sequences.

In this paper, we propose an on-line and real-time method for automatic anomaly detection in crowded scenes, which does not need any training stage. The method is inspired by the concept of Shannon entropy [6]. Entropy characterizes the uncertainty about the source of information and it increases for more sources of greater randomness. The idea is that the less likely an event is, the more information it provides when it occurs. Fig. 1 shows an example where the Shannon entropy value increases in the case of an abnormal event.
The two main contributions of the proposed approach are:

1. The use of two different metrics, namely *instant entropy* and *temporal occupancy variation*, to detect on-line abnormal situations in crowded scenes;
2. An unsupervised segmentation algorithm for images containing crowds.

[1] Definition from the Oxford Dictionary.

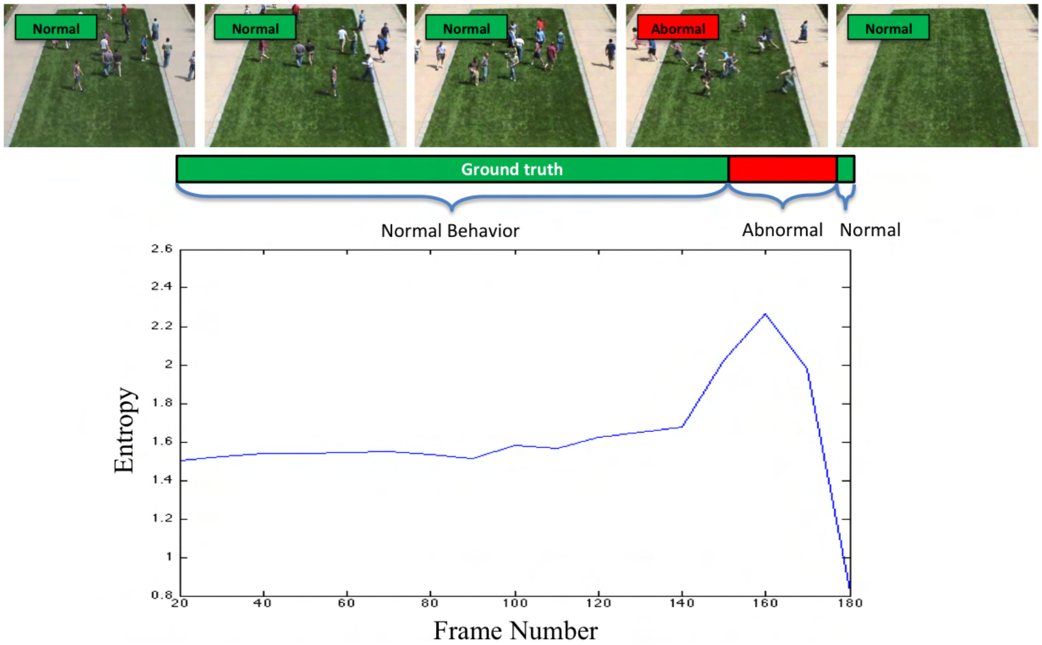


Figure 1: Shannon entropy variation in the sequence Lawn from the UMN data set.

Furthermore, we provide:

- A novel video sequence annotated with ground truth data, containing images of hundreds of runners at the start of a marathon, as an example of crowd video with locally steady optical flow.
- Ground truth annotations for two well-known video sequences containing abnormal events in crowded scenes, namely PETS 2009 [7] and AGORASET [8].
- The C++ source code and all the data used for the experimental evaluation, thus allowing for reproducing the results described in this paper and to compare other similar approaches.

The remainder of the paper is organized as follows. Related work is analyzed in the next Section 2, while our method is presented in Section 3. Section 4 describes the qualitative and quantitative experimental results, providing also a comparison with other on-line approaches in the literature. Finally, conclusions are drawn in Section 5.

2. Related Work

The techniques for crowd behavior analysis are usually grouped into two main categories [3, 9]: 1) object-based and 2) holistic approaches. In the object-based methods the analysis is carried out at an individual level. For example, it can be of interest to detect if a single person is trying to enter a restricted area or if an individual is moving against the dominant flow. On the other hand, holistic techniques treats the crowd as a single entity, trying to extract global information, such as the main flow of the crowd, instead of analyzing single trajectories.

We propose a different classification, based on the nature of the methods used for detecting abnormal situations. According to our classification, existing approaches can be grouped into:

- Statistical analysis;
- Background subtraction;
- Image segmentation;
- Classification.

Statistical analysis. Methods in this category are based on the collection of particular features representing the flow of the crowd. For example, Mehran et al. [9] propose a method for localizing abnormal behaviors by using a social force model. A grid of particles is placed over the image for analyzing the space-time average of optical flow. The moving particles are treated as individuals and the flows in the scene are estimated by employing the social force model. The interaction forces are then mapped into the image plane to obtain a Force Flow for every pixel in the current frame. Spatio-temporal volumes of Force Flow are randomly selected for modeling the normal behavior of the crowd. Then, the normal and abnormal behaviors are classified by exploiting an approach based on a bag of words. The regions of anomalies in the abnormal frames are localized using interaction forces. Social force model algorithms only consider the temporal characteristic, i.e. local velocity, and ignore the spatial information such as local density for detecting the crowd behavior.

Zhu and Saligrama [1] propose a probabilistic framework that takes into consideration local spatio-temporal anomalies to characterize the observed scene by optimal decision rules. If anomalies are local optimal decision, they are local as well, even if the behavior exhibits global spatial and temporal statistical dependencies. This helps to collapse the large ambient data dimension space to detect local anomalies. Consistent data-driven local empirical rules with provable performance can be derived with limited training data. The empirical rules are based on scores functions derived from local nearest neighbor distances. These rules aggregate statistics across spatio-temporal locations and scales and produce a single composite score for video segments. However, this method is scene-specific.

Chang et al. [10] describe a statistical framework able to recognize group-level activity in many scenarios, by combining a soft grouping metric and track-based motion analysis. The approach recognizes group interactions without making hard decisions about the underlying group structure. In particular, a path-based grouping scheme is used to understand if an individual belongs to a group. The method is bottom-up and thus suffers when the tracking output is not reliable.

Zhang et al. [11] describe a social attribute-aware force model for abnormal crowd pattern detection in video sequences. An unsupervised method is used to estimate the scene scale and two attributes, namely social disorder and congestion, are introduced to describe the realistic social behaviors by means of statistical context features. Through the semantic attribute-aware enhancement, it is possible to improve the model on the basis of social forces. Even if the method has good results, it is an off-line method.

Kratz and Nishino [12] describe a statistical framework for modeling the motion pattern behavior of extremely crowded scenes in order to detect unusual events. The authors model the dense activity of the crowd using a 3D Gaussian distribution of spatio-temporal gradients, by capturing the local spatio-temporal motion patterns through a distribution-based Hidden Markov Model. The results demonstrate that the proposed approach provides a suitable representation for analyzing crowded scenes, detecting unusual motion patterns in pedestrian behavior including movement against the normal flow of traffic. The method is appropriate for crowded scenes of a very high density, but it cannot handle videos containing middle to low density crowds, which is often the case in video surveillance systems.

Background Subtraction. Approaches that use background subtraction are commonly based on the creation of a Gaussian Mixture Model (GMM) to extract the foreground objects. For example, Fradi et al. [13] propose a method for people counting that harness the advantage of incorporating a uniform motion model into GMM background subtraction to obtain high accurate foreground segmentation. The counting is based on foreground measurements, where a perspective normalization and a crowd measure-informed corner density are introduced with foreground pixel counts into a single feature. The approach demonstrates the benefits of integrating GMM with motion cue and of normalizing the proposed feature as well. However, the method is not adaptive to varying illumination conditions.

Srivastava et al. [14] describe a method for crowd flow estimation by counting the number of people passing through a designated region in a unit of time. The method analyzes the total number of foreground pixels over a chosen time period, since it is directly proportional to the number of people passing through a predefined area. A scaling factor, which depends on the local texture features, is used to manage the occlusions. This approach have better performance with respect to individual tracking on dense crowd, but large occlusions can limit the foreground pixel-based counting for extracting the exact number of people in the scene.

Li et al. [15] propose a foreground detection approach for crowd motion analysis called optical flow and background model (OFBM). The method relies on Lucas-Kanade optical flow and Gaussian background model methods to eliminate the noise due to brightness changes and occlusions. This approach overcomes the shortages of optical flow and background subtraction, but it is not fast enough to be applied in real-time processing.

Image Segmentation. The methods in this category rely on the identification of the crowd flow by using a grid particles placed in the scene in order to detect the evolution of the crowd flow in the scene. Solmaz et al. [16] propose a framework to identify multiple crowd behaviors through stability analysis for dynamical systems. A scene is overlaid by a grid of particles initializing a dynamical system defined by the optical flow. Time integration of the dynamical system provides particle trajectories that represent the motion in the scene; then, these trajectories are used to locate regions of interest in the scene. Linear approximation of the dynamical system provides behavior classification through the Jacobian matrix. The eigenvalues are only considered in the regions of interest, consistent with the linear approximation and the implicated behaviors. In such a way, the method can identify five types of behaviors. However, the method can be not useful when significant overlap of motion patterns is present in the scene or when a consistent characteristic flow is missing.

Ali and Shah [17] propose a framework in which Lagrangian Particle Dynamics is used for the segmentation of high density crowd flows and detection of flow instabilities. The authors treat a flow field generated by a moving crowd as an aperiodic dynamical system. Therefore, a grid of particles is overlaid on the flow field in order to monitor the evolution of the particles. Then, a Finite Time Lyapunov Exponent (FTLE) field is used to quantify the amount of particles and to reveal the Lagrangian Coherent Structures (LCS) present in the underlying flow. The LCS divides flow into regions (motion patterns) respecting the dynamics of the scene. The changes in the number of flow segments is considered as an instability. It is worth noting that, this method is not able to deal with overlapping motion patterns since only one motion label is assigned to each pixel.

Classification. This category includes approaches that exploit classifiers to recognize the behavior of the observed scene. Greenewald and Hero [18] describe an approach able to learn the normative multi-frame pixel joint distribution and to detect deviations from it. The authors use a likelihood based approach to learn the spatio-temporal covariance in the low-sample regime. The approach estimates the covariance by exploiting parameter reduction and sparse models. The first method considered is the representation of the covariance as a sum of Kronecker products, which is found to be an accurate approximation in this setting. Then, they consider the sparse a multi-resolution model and apply the Kronecker product methods to it for further parameter reduction. Even though such a method achieves good results, the authors state that part of the data set has been used for training the classifier, thus making the method environment-dependent.

Idrees et al. [19] describe an approach to count the number of individuals in extremely dense crowds. The method relies on multiple sources such as low confidence head detections, repetition of texture elements (using SIFT), and frequency-domain analysis to estimate counts, along with confidence associated with observing individuals, in an image region. Then, a global consistency constraint on count using Markov Random Field is employed. Moreover, the approach scales well to different densities, producing constant error rates across images with diverse count. However, this approach does not provide a true estimate of the number of individuals crossing into and out of an area.

To overcome all the above described problems related to the huge amount

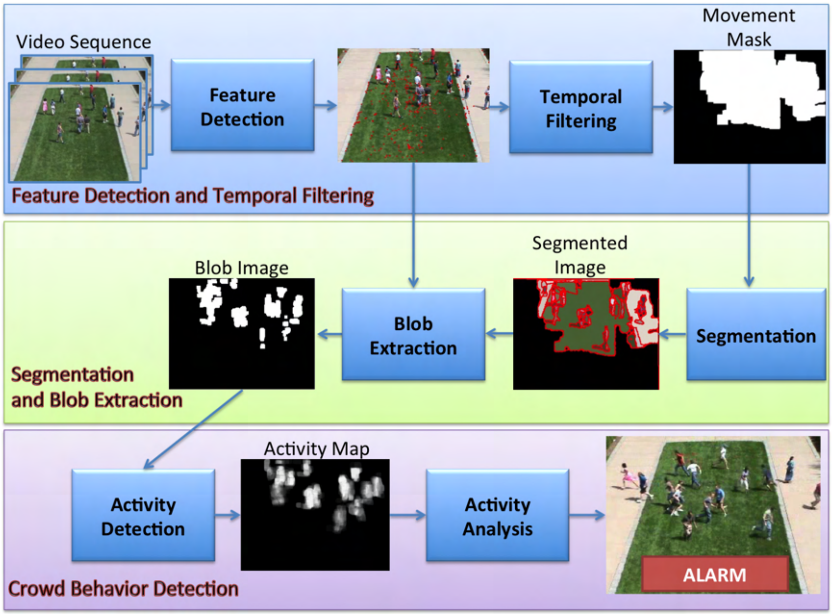
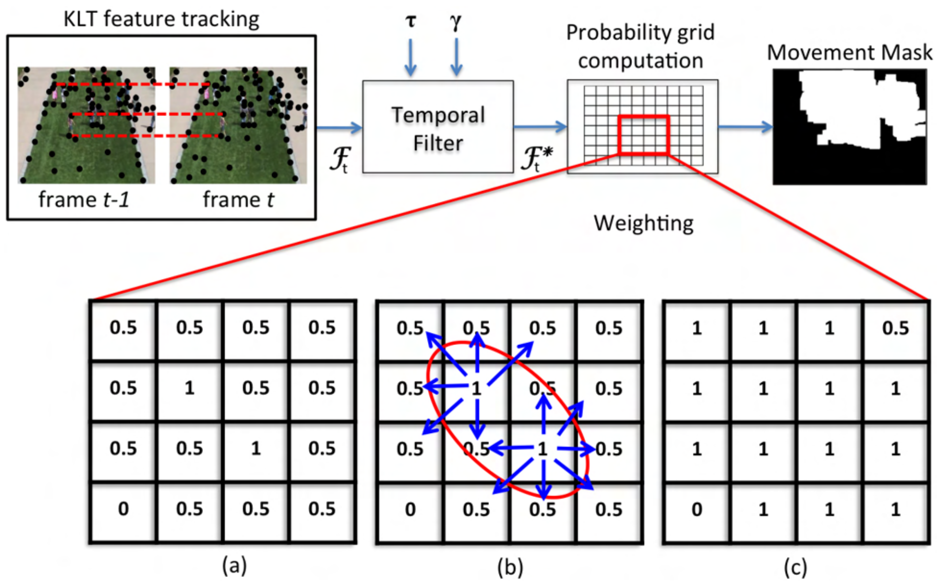


Figure 2: Block diagram of the FSCB method with the three stages.

of data needed for training the classifiers, to the scene-specific descriptors, and moreover, to the non-real time performance, we propose, in this work, a statistical analysis approach that combines feature detection and image segmentation in order to detect abnormal behaviors. The proposed method is on-line and runs in real-time. In particular, two metrics, namely *entropy* and *temporal occupation variation*, are taken into account for detecting abnormal crowd behaviors, without the need of a training phase.

3. Feature Tracking and Image Segmentation for Behavior Detection

In this section, the description of our crowd behavior detection method, called **FSCB**, is provided. FSCB is made of three main steps: 1) **F**eature detection and temporal filtering; 2) image **S**egmentation and blob extraction; 3) **C**rowd **B**ehavior detection. The block diagram of the FSCB method is shown in Fig. 2 and the details of each step are given in the rest of this section. Moreover, the FSCB website, containing the source code of the method and the data used for the experimental evaluation, is located at `http://www.dis.uniroma1.it/~pennisi/eventdetection.html`.



0.5	0.5	0.5	0.5
0.5	1	0.5	0.5
0.5	0.5	1	0.5
0	0.5	0.5	0.5

(a)

0.5	0.5	0.5	0.5
0.5	1	0.5	0.5
0.5	0.5	1	0.5
0	0.5	0.5	0.5

(b)

1	1	1	0.5
1	1	1	1
1	1	1	1
0	1	1	1

(c)

Figure 3: Movement mask is generated trough a probability grid: (a) The value 1 is assigned for each tracked feature point, while the value 0.5 is assigned to its 8-connected neighbors. (b) Once all the features have been considered, if there are adjacent points with value 1, then (c) the probabilities of all their neighbors are set to 1 too.

3.1. Feature Detection and Temporal Filtering

The first step of FSCB aims at finding descriptive visual features of the crowd flow in the observed scene. We assume that the following conditions hold in the scene: i) Brightness constancy, i.e., the projection of the same point looks the same in every frame; ii) Small motion, i.e., points do not move very far; iii) Spatial coherence, i.e., points move like their neighbors. The above conditions are usually satisfied in video sequences containing crowded scenes recorded at 25 frames per second with a fixed camera.

Given the above assumptions, we decided to exploit the Kanade-Lucas-Tomasi (KLT) feature tracker [20] for detecting and tracking local visual features, instead of using other feature descriptors like Harris corners, SIFT or SURF. Indeed, KLT works very well in situations where the distance between images is small, it displays good immunity to tuning parameters, and it has low computational needs [21].

The output of the KLT tracker at time t is a set $\mathcal{F}_t\{\langle f_i^{t-1}, f_i^t\rangle \mid i = 1, \ldots, n\}$, of corresponding feature points in two consecutive frames captured at time $t-1$ and t, respectively (see Fig. 3). Once $\mathcal{F}_t$ has been calculated, a temporal filtering is applied in order to create a binary mask M, containing only the moving points in the scene. To this end, the two thresholds τ and γ are adopted to filter out not moving points: τ is the temporal window size representing the length of an history queue, while γ is the minimum velocity value (in pixel per second) to consider a feature point as a moving one.

For each couple $\langle f_i^{t-1}, f_i^t \rangle \in \mathcal{F}_t$ a vector $\mathcal{V} = \{v_1, ..., v_z\}$, $z \leq \tau$, is maintained in memory, where the element v_j, $1 \leq j \leq z$, represents the velocity, recorded at time $t - z + j$, of the feature point f_i. In particular, the velocity v of a feature point f at time t is calculated as:

$$v_f^t = \frac{\sqrt{\left(f^{t-1}(x) - f^t(x)\right)^2 + \left(f^{t-1}(y) - f^t(y)\right)^2}}{frame\ rate\ in\ seconds} \tag{1}$$

where the couple (x, y) represents the pixel coordinates.

At the arrival of every new frame, a set of filtered features $\mathcal{F}_t^*$ is obtained by discarding from $\mathcal{F}_t$ the features having $v_z \leq \gamma$. Then, a probability grid is used for weighting the motion points $\mathcal{F}_t^*$. The grid has the same size of the input image and it is divided into cells, one for each pixel, and each cell is initialized with the value 0. The values of the cells in the grid are modified as follows: 1) for each feature point belonging to $\mathcal{F}_t^*$, the value 1 is assigned to the corresponding cell, while the value 0.5 is assigned to all its 8-connected neighbor cells in the grid (see Fig. 3a). Then, 2) the grid is further modified in order to cluster adjacent moving points: If a cell of the grid with value 1 has neighbors with value 1 as well (as shown in Fig. 3b), then all the cells in their neighborhood are set to 1 also (see Fig. 3c).

Finally, the binary movement mask M is generated by considering the cells in the grid with value 1 as white points, and the remaining ones as black points. M provides a map of the regions in the image where the moving features have been detected. In all our experiments, we set τ and γ to 10 frames and 2 pixels per second, respectively.

3.2. Image Segmentation and Blob Extraction

The RGB image segmentation is performed by using an approach similar to the one described by Taylor and Cowley in [22]. First, the current RGB frame I in input is filtered by using M as a mask, thus obtaining a new RGB image I^*. Then, I^* is segmented according to two consecutive steps: 1) *Edge Segmentation* and 2) *Delaunay Triangulation*. The former is used

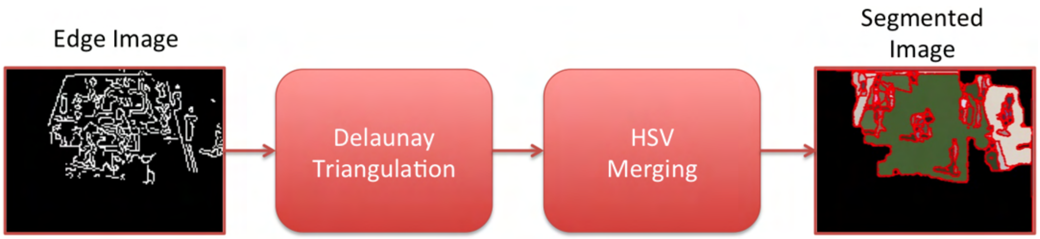


Figure 4: The segmentation step is made of 3 steps: edge detection, Delaunay Triangulation and HSV merging.

for splitting the image into local coherent regions, the latter for aggregating homogeneous regions in a global fashion.

I^* is filtered by applying a Gaussian blur filter with a 3×3 kernel size. Then, it is converted to grayscale, obtaining an image G, and the Edge Segmentation procedure begins with a Canny edge extraction, that leads to the creation of an edge image containing the intensity edges in G (Fig. 4). The two parameters *min* and *max* in the Canny algorithm have been set to the values 0.03 and 2.0, respectively, in order to focus on short edges in G.

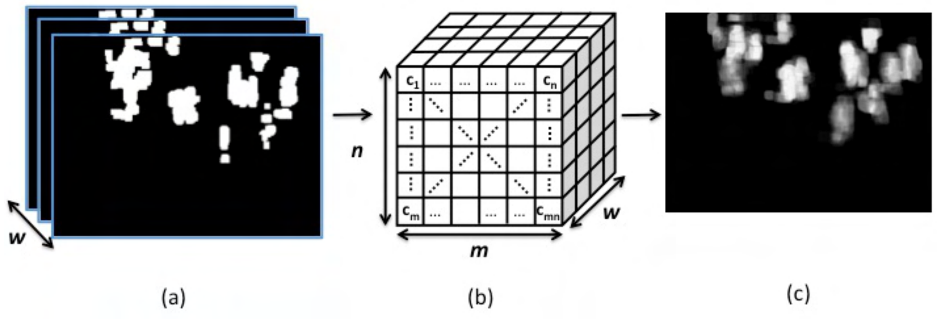
The contents of the edge image are then vectorized into connected line segments and used as input for a Delaunay Triangulation procedure, which computes a triangular tessellation of the image.

The Delaunay Triangulation of a point set $\mathfrak{P}$ is characterized by the empty circumdisk property: No point in $\mathfrak{P}$ lies in the interior of any triangle's circumscribing disk.

Definition *[23].* In the context of the finite point set $\mathfrak{P}$, a triangle is Delaunay if its vertices are in $\mathfrak{P}$ and its open circumdisk is empty (i.e., it contains no point in $\mathfrak{P}$). It is worth noting that, any number of points in $\mathfrak{P}$ can lie on a Delaunay triangle's circumcircle. An edge is *Delaunay* if its vertices are in $\mathfrak{P}$ and it has at least one empty open circumdisk. A Delaunay Triangulation of $\mathfrak{P}$, denoted *Del* $\mathfrak{P}$, is a triangulation of $\mathfrak{P}$ in which every triangle is Delaunay.

Given the connected line segments generated as in [24], the function *Delaunay* from the CGAL[2] library is used to carry out the triangulation. The nodes of the planar triangular graph obtained from the Delaunay Triangulation represent the set of triangles and the edges indicate adjacency relations

[2] `https://www.cgal.org`



(a)

(b)

(c)

Figure 5: Activity Map Computation: a) A set of blob images is collected over a time window w. b) A 3D-Grid of size $m \times n \times w$ is used to record the time persistence of each pixel. c) The activity map is obtained by clustering the data in the 3D-Grid.

between them, i.e., there is an edge between two nearby triangles.

The triangular graph is segmented using a merging procedure that iteratively finds and merges the two regions with the lowest normalized boundary cost. Each one of the triangles in the graph is considered in turn, calculating the average HSV color of all the pixels that lie within its circumcircle. An association threshold ω is used for merging the triangles color similarity: If a pair of triangles have a difference in the normalized HSV values lower than ω, then they are merged into a single triangle. An example of the results produced by the image segmentation task is shown in Fig. 4.

The extraction of the blobs is performed by applying again the KLT feature tracker, this time on the image I^*, in order to find the moving blobs. A set $\mathcal{F}_{blob}$ of couples of corresponding feature points is generated as before. Then, the features are filtered by using Eq. 1, thus obtaining a new set of filtered features $\mathcal{F}^*_{blob}$. The set $\mathcal{F}^*_{blob}$ is re-projected onto the segmented image in order to detect the set S of moving blobs. A blob is considered as a moving one if its area contains at least a feature point $f \in \mathcal{F}^*_{blob}$. In such a way, a binary blob image is obtained.

3.3. Crowd Behavior Detection

The crowd behavior in the observed scene is detected by carrying out a statistical analysis on the data collected over a temporal window w. As shown in Fig. 5, given in input a set of binary blob images (Fig. 5a), a 3D-Grid of size $m \times n \times w$ (Fig. 5b) is used to generate a grayscale activity map (Fig. 5c). The width m and the height n of the grid are the same of the input

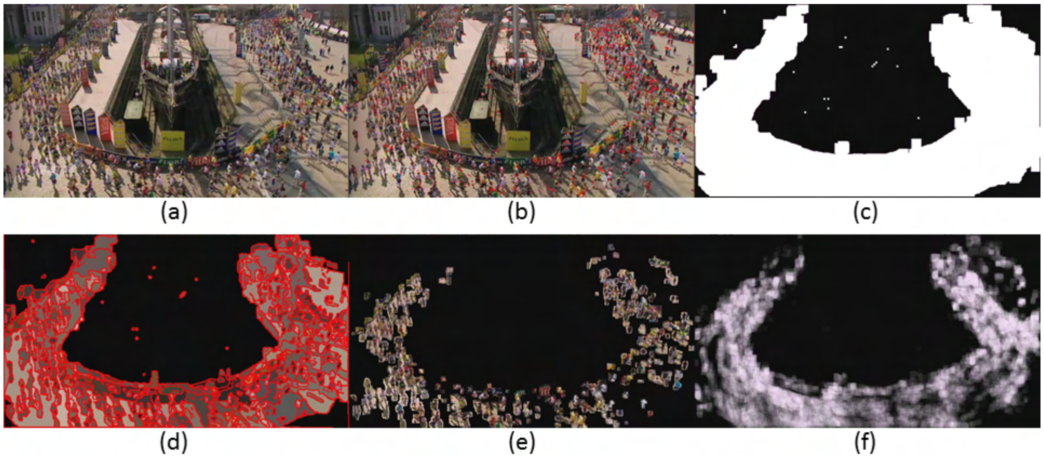


Figure 6: An example of activity map computation. a) The current frame in input (video sequence from [17]). b) Visual feature extraction with KLT. c) Temporal mask. d) Segmented image. e) Moving blob image f) Activity map.

image, while the depth w corresponds to the length of the temporal window. Then, each voxel a in the 3D-Grid is set to the value 1 if the corresponding pixel p in the blob image is white. The depth of the voxel a is represented by a set of 1s, equal to the number of the corresponding white pixels in the blob images, over the time window w. In such a way, the temporal persistence of each point p in the scene is given by the depth of the corresponding voxel a in the 3D-Grid. The gray values in the activity map (Fig. 5c) are strictly related to the persistence of the pixel during the time window w, i.e., a value near 255 in the activity map indicates a point with high activity. In our experiments, the length w of the temporal window is set to the frame rate value of the video sequence at hand.

Fig. 6 shows all the steps that are performed for obtaining the activity map in a high density crowd scenario. It can be noted that only the part of the image containing a real motion is taken into account.

Once the activity map is available, it is possible to analyze the trend of the following two measures:

1. *Image entropy*;
2. *Temporal Occupancy Variation* (TOV).

The image entropy serves for obtaining a measure of the uncertainty in the image values by counting the average amount of information required to

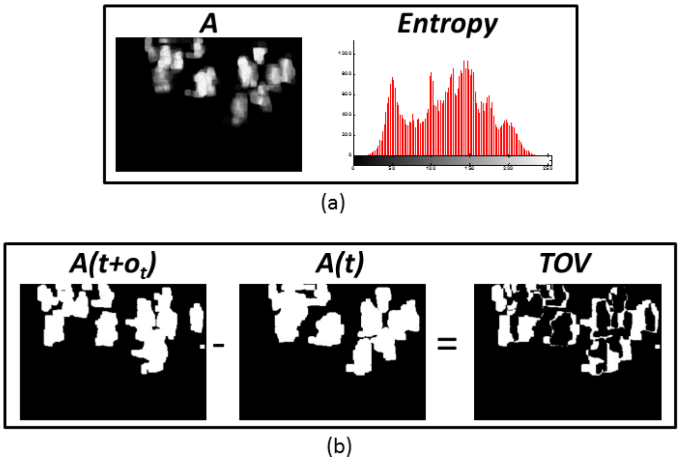


Figure 7: The two measures considered for crowd behavior detection: a) Image entropy. b) Temporal occupancy variation.

encode the image values. The zero order entropy for an image I is defined as:

$$Entropy\,(I) = \sum_{i=1}^{n} p_i \, log_2 \, p_i \tag{2}$$

where n is the number of separate symbols, p_i is the frequency of the i-th pixel in the image, and the result is measured in bits per symbol (pixel value). Then, by assuming that an infrequent event provides more information than a frequent event [25], it is possible to monitor the instant variation of an image I in order to detect sudden changes. A threshold e_v is set as a "sentinel": If $Entropy\,(I(t+1)) - Entropy\,(I(t)) > e_v$ something of anomalous is happening. An example of image entropy calculation on an activity map A is shown in Fig. 7a.

The temporal occupancy variation (TOV) takes into account the space occupied by the detected moving blobs over time. Given a temporal threshold o_t, the TOV is given by: $TOV = A(t+o_t) - A(t)$. The value of TOV represents the percentage of image space occupied during a time interval o_t. If the value of TOV increases, it means that the scene is changing. We assume that in case of a great variation in the TOV value, an abnormal event is happening. An example of TOV calculation is shown in Fig. 7b.

A discussion about the values used for the thresholds e_v and o_t is provided in the next section.

4. Experimental Evaluation

The experimental results described in this section are related to the problem of detecting events of interest in crowded scenes. Multiple publicly available video sequences have been selected for quantitatively evaluating the proposed approach and for comparing it with other recent state-of-the-art on-line approaches.

4.1. Data Sets Description

Four different data sets have been selected for the experiments: UMN [26], PETS 2009 [7], AGORASET [8], and Rome Marathon [27]. Each data set contains one or multiple video sequences and the corresponding ground truth data. Each frame in a video sequence is labeled with a value "normal" or "abnormal", with "abnormal" meaning that an event of interest is in progress. Ground truth data are already available for the UMN data set, while for the other three data sets we have generated the corresponding annotation data, that are available from the FSCB website. Fig. 8 shows two sample frames from each one of the considered sequences.

UMN Data set. UMN data set has been collected by the University of Minnesota, USA, and it consists of eleven videos representing escape events. The videos are captured in three different indoor and outdoor scenes, commonly denoted as Lawn, Indoor, and Plaza. Each video starts with a crowd, of about 20 people, that walks in different directions, then an abnormal event causes people to run away.

AGORASET. The AGORASET data set is composed of synthetic scenes representing various crowd simulations. Seven scenes are represented, corresponding to an evolution of a human flow in different environments, e.g., an environment with obstacles, an evacuation through a door, etc. In our experiments, we focus on the dispersion scenario (see Fig. 8), where a crowd with about 100 people walks around in a close environment and then moves suddenly to the limit of the environment. Moreover, we manually annotated this sequence creating ground truth data.

PETS 2009. This data set has been recorded for the workshop PETS 2009 at Whiteknights Campus, University of Reading, UK. PETS 2009 comprises multi-sensor sequences containing crowd scene scenarios with increasing scene complexity and it is made of three data sets: S1) concerning person count and density estimation; S2) addressing people tracking; S3) involving flow

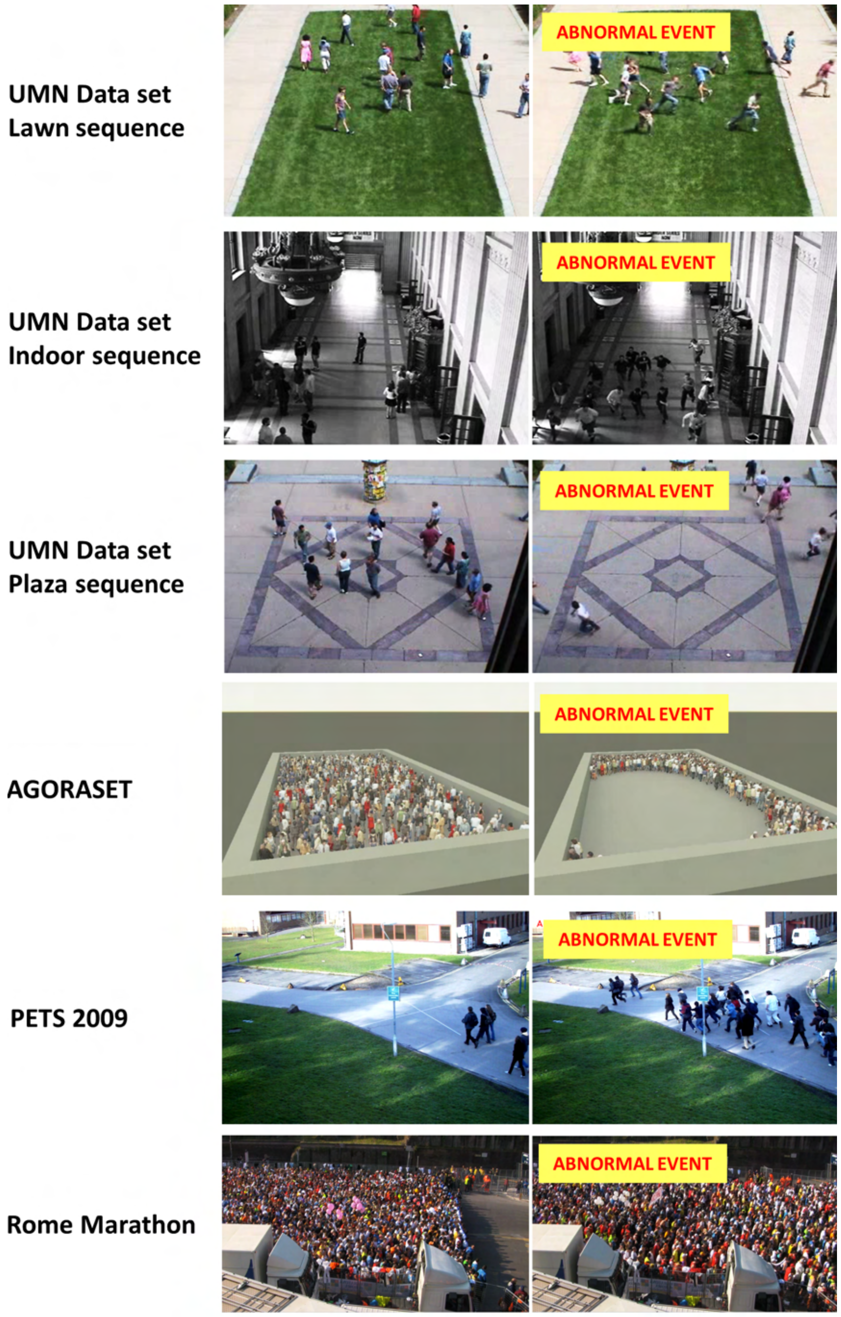


Figure 8: The considered data sets.

Table 1: Data sets characteristics.

Name	**Event Type**	**No. of Events**	**No. of Frames**	**Frame Size**	**Frame Rate**	**Crowd Density Level**	**Scene**	**Behavior**
UMN	escape	11	7710	320×240	30	2	real	artificial
PETS 2009	escape	4	815	768×576	7	1	real	artificial
AGORASET	escape	1	600	640×480	30	3	synthetic	artificial
Rome Marathon	race starting	2	11575	1920×1080	25	3	real	natural

analysis and event recognition. In our experiments, we used the S3 data set (see Fig. 8).

Rome Marathon. Since the scarcity of publicly available data set for crowd behavior understanding is an actual problem for the Computer Vision community, we decided to publish two novel video sequences containing crowded scenes. The data set has been recorded during the 2013 Rome Marathon and it is available for download, together with ground truth data for each video, at the FSCB website. The Rome marathon data set is made of two video sequences representing two different situations: 1) the starting of the marathon and 2) the cleaning of the street. As shown in the samples in Fig. 8, the scenes contain thousands of people participating to the marathon. As a difference with respect to the above described data sets, this scenario also contains natural human behavior (i.e., people were not instructed to act in a particular way for the scope of this acquisition).

Table 1 summarizes the characteristics of the above described data sets. We used a number from 1 to 3 to denote the crowd density level.

4.2. Metrics

In order to obtain quantitative results for our FSCB algorithm, we measured the number of frames in the video sequence at hand that are detected as False Positives (FP), True Positives (TP), False Negatives (FN), and True Negatives (TN) with respect to the ground truth data. True Positive Rate (TPR) and False Positive Rate (FPR) can be computed with the following formulas:

$$TPR = \frac{TP}{TP + FN} \tag{3}$$

$$FPR = \frac{FP}{FP + TN} \tag{4}$$

Table 2: Anomaly detection results on the whole UMN data set. Our approach is compared with other published on-line methods.

Method	Type	Area under ROC curve (AUC)
Optical Flow [9]	on-line	0.84
Neural Network [30]	on-line	0.93
FSCB	on-line	**0.95**

TPR and FPR can be used for generating a Receiver Operating Characteristics (ROC) curve and for computing the relative Area Under Curvature (AUC). The area under the ROC is a convenient way of comparing different classification methods. A random classifier has an area of 0.5, while and ideal one has an area of 1. The quantitative results obtained by FSCB on the four data sets are provided below.

4.3. Quantitative Results of the Entire Pipeline

In order to qualitatively evaluate the performance of our FSCB algorithm, we tested the approach generating the ROC curve for each of the above described data sets. All the used ground truth data are publicly available at the FSCB website.

It is worth noting that, there exists a large variety of off-line crowd behavior detection methods that are able to achieve an AUC value near 1 on the considered sequences (e.g., a value of 0.99 is obtained in [29] on UMN). However, such performance are obtained by analyzing the entire video, i.e., having the possibility of exploiting knowledge about events that will happen in the future. This type of analysis can be useful to obtain a model for different crowd behaviors, but off-line analysis can result ineffective for practical use. For such a reason, we compare our FSCB method only with on-line state-of-the-art methods.

For the UMN data set a double comparison has been carried out. In the first set of experiments, in order to carry out a fair comparison with published results, the entire data set is considered as a whole video sequence.

The ROC curve generated on the entire UMN sequence (11 videos treated as a single one) is shown in Fig. 9. In particular, the value of e_v has been varied in the range $0.1 \leq e_v \leq 0.2$, while the value of o_t in the range $30 \leq o_t \leq 35$.

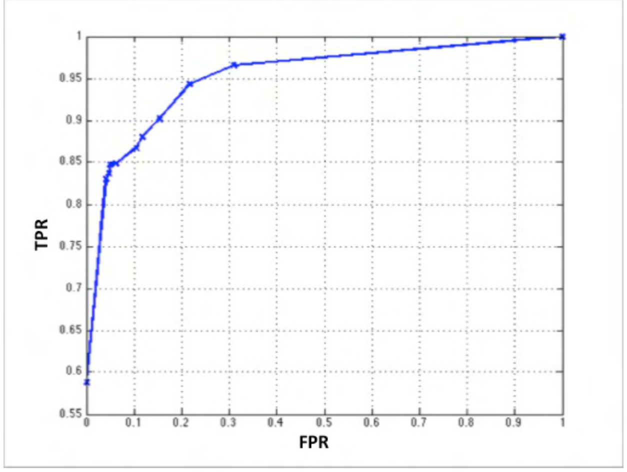


Figure 9: The ROC curve for FSCB on the whole UMN data set. The results are obtained by varying the thresholds τ and o_t.

Table 2 shows that FSCB achieves better results than the methods relying on pure Optical Flow (results from [9]) and on a Neural Network (results from [30]). FSCB outperforms such methods because it operates with sets of blobs instead of feature sets, which make the approach more robust to sudden changes in the scene. Moreover, the FSCB method is not based on a classifier for identifying abnormal events, thus making the approach more flexible and feasible to several environments. The false positives for FSCB are generated by the anticipated detection of the crowd event with respect to the ground truth data.

The second set of experiments on the UMN data set has been carried out by considering the sequences as divided according to the three different scenarios: Lawn, Indoor and Plaza. Our method has been compared with other two recent on-line crowd behavior detection methods [31, 32]. Results are shown in Table 3. FSCB performs slightly better than the other two methods on all the three considered sequences.

As already mentioned, three additional video sequences have been considered along with the UMN data set for quantitative evaluation. The results are shown in Table 4. For all the three considered data sets, FSCB is able to achive good results with an AUC value over 0.90. FSCB obtains good detection results on different video sequences, without the need of using a classifier for detecting the crowd behavior in the observed scene.

Table 3: Anomaly detection results on single sequences of UMN data set. FSCB method is compared with other published methods.

Method	**Type**	**Area under ROC curve (AUC)**		
		Lawn	Indoor	Plaza
STCOG [31]	on-line	0.9362	0.7759	0.9661
COV [32]	on-line	0.9605	0.8628	0.9746
FSCB	on-line	**0.9641**	**0.8764**	**0.9750**

Table 4: Anomaly detection results of the FSCB method on PETS 2009, AGORASET, and Rome Marathon data sets.

Data set	**Area under ROC curve (AUC) for FSCB**
PETS 2009 [7]	0.93
AGORASET [28]	0.94
Rome Marathon [27]	0.96

In order to compare the results of our method with related state-of-the-art approaches, the above reported comparison has been carried out by analyzing all the frames in the sequences, without considering the actual computational speed. A discussion about the computational speed and the results that can be obtained with an on-line quantitative evaluation are given below.

Computational Speed. We tested the computational speed of FSCB in terms of frames per second (FPS). To the best of our knowledge, the computational load for similar approaches in the literature has not been published. The tests have been made by using a commercial notebook with an Intel Core i7 CPU 2.4 GHz 8 GB RAM and a single-threaded C++ implementation of the FSCB algorithm. The results are shown in the last column of Fig. 10. From the obtained results it can be noted that, for 320×240 images, FSCB runs in real-time. When the frame size increases the computational speed for FSCB decreases arriving at 5 fps for full high definition (FHD) images.

On-line Quantitative Evaluation. FSCB approach is designed to be completely on-line and it does not need any training phase. We performed another set of experiments by considering the actual frame rate of the sequences at hand, i.e., down-sampling the sequences in input to simulate an

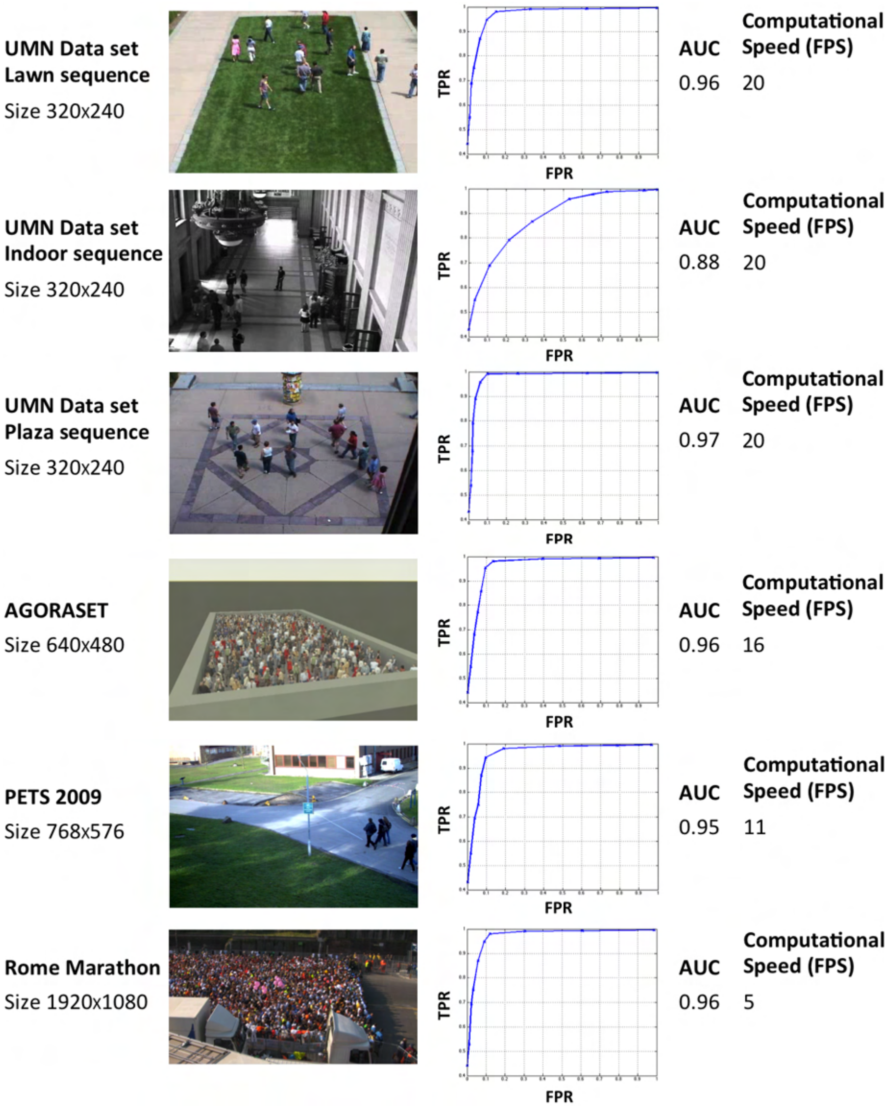


Figure 10: FSCB on-line simulation results.

on-line data stream. In particular, we calculated the results obtained in a realistic real-time setting, where the acquisition frame rate corresponds to the processing frame rate. The results and the ROC curves for the on-line simulation on all the considered data sets are shown in Fig. 10.

Additional analysis has been carried out on the Rome Marathon data set, by varying the frame size for evaluating the relationship between the

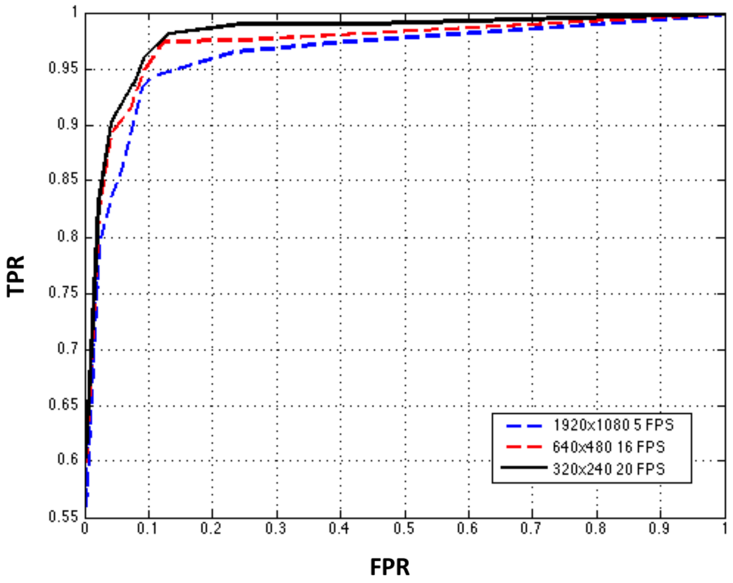


Figure 11: Receiver Operating curves related to the Rome Marathon data set, obtained by varying the frame size.

Table 5: Evaluation of the FSCB on the Rome Marathon data set by varying the frame size.

Frame Size	**FPS**	**AUC**
1980×1080	5	0.9610
640×480	16	0.9727
320×240	20	0.9785

detection results and the frame rate of the input data. Results shown in Fig. 11 and in Table 5 demonstrate that the accuracy of the results increases with a higher frame rate, which directly influences the feature tracking step, allowing to achieve better results.

4.4. Analysis of the Single Pipeline Stages

To demonstrate experimentally the impotance of each step in the FSCB pipeline, we replaced two stages of the pipeline shown in Fig. 2 with similar state-of-the-art techniques. In particular, two replacements have been carried out: 1) The feature detection, temporal filtering, and segmentation steps have been replaced by the MOG2 background subtraction method [33], which

Table 6: Evaluation of the Pipeline Stages.

Data set	**AUC for FSCB**	**AUC for BGSUB**	**AUC for SEGM**
UMN [26]	**0.95**	0.90	0.86
PETS 2009 [7]	**0.93**	0.87	0.78
AGORASET [28]	**0.94**	0.88	0.80
Rome Marathon [27]	**0.96**	0.91	0.75

provides as output a binary foreground mask; 2) The temporal mask and the segmentation stages have been replaced by the Watershed segmentation method [34]. We used the MOG2 and Watershed implementations provided by the OpenCV library[3].

The two obtained modified pipelines have been tested on the same data sets used for the original FSCB method. The results of the comparison are shown in Table 6. The proposed pipeline performs better than the other two modified pipelines on all the considered video sequences. Indeed, the use of background subtraction makes the algorithm less sensitive to rapid changes, due to the time interval needed for updating the background model, that is not well suited for rapid changes.

The use of a pure segmentation method like Watershed can provoke false positives in areas of the image not actually involved motion (e.g., due to reflections). Furthermore, the use of the information about the velocity of the blobs performed by FSCB helps in selecting only the blobs that are related to the crowd flow, while such a selection is not possible when using a pure segmentation method.

5. Conclusions

In this paper, a real-time and on-line crowd behavior detection algorithm for video sequences is described. The algorithm, called FSCB, is based on a pipeline made of the following stages: 1) stable features are tracked between frames of the sequence; 2) a temporal mask is extracted; 3) moving blobs are found using segmentation; 4) anomalous events are detected using two measures, i.e., instant entropy and temporal occupancy variation.

[3] http://opencv.org

Quantitative experiments have been conducted on different publicly available data sets: UMN [26], PETS 2009 [7], AGORASET [8]. For PETS 2009 and AGORASET, ground truth data have been produced and made available at the FSCB website. Furthermore, a novel annotated data set, Rome Marathon [27], containing crowded scenes from the start of a marathon, has been created.

FSCB has been quantitatively compared with other state-of-the-art methods for on-line crowd event detection. The results of the comparison demonstrate the effectiveness of the proposed approach, that works without the need of a training stage and obtain real-time performance on 320×240 images.

References

[1] V. Saligrama, Z. Chen, Video anomaly detection based on local statistical aggregates, in: Computer Vision and Pattern Recognition (CVPR), (2012), pp. 2112–2119.

[2] C. Brax, L. Niklasson, M. Smedberg, Finding behavioural anomalies in public areas using video surveillance data, in: 11th International Conference on Information Fusion (2008), pp. 1–8.

[3] J. Junior, S. Mussef, C. Jung, Crowd analysis using computer vision techniques, IEEE Signal Processing Magazine (2010).

[4] B. Zhan, D. N. Monekosso, P. Remagnino, S. Velastin, L.-Q. Xu, Crowd analysis: a survey, Machine Vision and Applications 19 (2008) 345–357.

[5] M. Rodriguez, J. Sivic, I. Laptev, J.-Y. Audibert, Data-driven crowd analysis in videos, in: IEEE International Conference on Computer Vision (ICCV), 2011, pp. 1235–1242.

[6] C. E. Shannon, A mathematical theory of communication, The Bell System Technical Journal 27 (1948) 379–423, 623–656.

[7] J. Ferryman, A. Shahrokni, Pets 2009 benchmark data, `http://cs.binghamton.edu/~mrldata/pets2009.html`, 2009.

[8] P. Allain, N. Courty, T. Corpetti, C. Creusot, Agoraset: a dataset for crowd video analysis, `http://www.sites.univ-rennes2.fr/costel/corpetti/agoraset/Site/AGORASET.html`, 2012.

[9] R. Mehran, A. Oyama, M. Shah, Abnormal crowd behavior detection using social force model, in: Computer Vision and Pattern Recognition (CVPR), 2009, pp. 935–942.

[10] M.-C. Chang, N. Krahnstoever, W. Ge, Probabilistic group-level motion analysis and scenario recognition., in: International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 747–754.

[11] Y. Zhang, L. Qin, H. Yao, Q. Huang, Abnormal crowd behavior detection based on social attribute-aware force model, in: International Conference on Image Processing (ICIP), 2012, pp. 2689–2692.

[12] L. Kratz, K. Nishino, Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models, 2013 IEEE Conference on Computer Vision and Pattern Recognition (2009).

[13] H. Fradi, J. Dugelay, Low level crowd analysis using frame-wise normalized feature for people counting, in: Information Forensics and Security (WIFS), 2012 IEEE International Workshop on, pp. 246–251.

[14] S. Srivastava, K. K. Ng, E. J. Delp, Crowd flow estimation using multiple visual features for scenes with changing crowd densities., in: International Conference on Advanced Video and Signal-Based Surveillance (AVSS), IEEE Computer Society, 2011, pp. 60–65.

[15] L. Wei, W. Xiaojuan, M. Koichi, Z. Hua-An, Foreground detection based on optical flow and background subtract, in: International Conference on Communications, Circuits and Systems, pp. 359–362.

[16] B. Solmaz, B. E. Moore, M. Shah, Identifying behaviors in crowd scenes using stability analysis for dynamical systems, IEEE Trans. Pattern Anal. Mach. Intell. (2012).

[17] S. Ali, M. Shah, A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis, in: Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–6.

[18] K. Greenewald, A. Hero, Detection of Anomalous Crowd Behavior Using Spatio-Temporal Multiresolution Model and Kronecker Sum Decompositions, Technical Report, AFRL ATR Center, 2014.

[19] H. Idrees, I. Saleemi, C. Seibert, M. Shah, Multi-source multi-scale counting in extremely dense crowd images., in: Computer Vision and Pattern Recognition (CVPR), 2013, IEEE, 2013, pp. 2547–2554.

[20] J. Shi, C. Tomasi, Good features to track, in: Computer Vision and Pattern Recognition (CVPR), 1994, pp. 593–600.

[21] J. Klippenstein, H. Zhang, Quantitative evaluation of feature extractors for visual slam, in: Fourth Canadian Conference on Computer and Robot Vision (CRV), pp. 157–164.

[22] C. Taylor, A. Cowley, Parsing indoor scenes using rgb-d imagery, in: Proceedings of Robotics: Science and Systems.

[23] S.-W. Cheng, T. K. Dey, J. Shewchuk, Delaunay mesh generation, CRC Press, 2012.

[24] Q. Wu, Y.Yu, Two-level image segmentation based on region and edge integration, in: Digital Image Computing: Techniques and Applications, pp. 957–966.

[25] T. Gevers, A. Gijsenij, J. van de Weijer, J. M. Geusebroek, Color in Computer Vision : Fundamentals and Applications, Series in Imaging Science and Technology, The Wiley-IS&T, 2012.

[26] University of Minnesota, Unusual crowd activity data set, `http://mha.cs.umn.edu/Movies/Crowd-Activity-All.avi`, 2006.

[27] Ro.Co.Co. Lab, Rome marathon data set, `http://www.dis.unrioma1.it/~pennisi/download/romemarathon.zip`, 2014.

[28] P. Allain, N. Courty, T. Corpetti, AGORASET: a dataset for crowd video analysis, in: 1st ICPR International Workshop on Pattern Recognition and Crowd Analysis, Tsukuba, Japan, pp. 26–31.

[29] W. Shandong, M. Brian E., S. Mubarak, Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes, Conference on Computer Vision and Pattern Recognition (2013) 2054–2060.

[30] Y. Cong, J. Yuan, J. Liu, Sparse reconstruction cost for abnormal event detection., in: CVPR, pp. 3449–3456.

[31] Y. Shi, Y. Gao, W. Ruili, Real-time abnormal event detection in complicated scenes, in: 20th International Conference on Pattern Recognition (ICPR), pp. 3653–3656.

[32] T. Wang, J. Chen, H. Snoussi, Online detection of abnormal events in video streams, Journal of Electrical and Computer Engineering (2013) 1–12.

[33] Z. Zivkovic, Improved adaptive gaussian mixture model for background subtraction, in: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2 - Volume 02, pp. 28–31.

[34] F. Meyer, Topographic distance and watershed lines, Signal Process. 38 (1994) 113–125.