Swarm Intelligence for Detecting Interesting Events in Crowded Environments

Vagia Kaltsa [1,2], *Student Member, IEEE,* Alexia Briassouli [2], *Member, IEEE,* Ioannis Kompatsiaris [2], *Senior Member, IEEE,* Leontios J. Hadjileontiadis [1], *Senior Member, IEEE,* and Michael G. Strintzis [1], *Life Member, IEEE*

***Abstract*—This work focuses on detecting and localizing anomalous events in videos of crowded scenes, i.e. divergences from a dominant pattern. Both motion and appearance information are considered, so as to robustly distinguish different kinds of anomalies, for a wide range of scenarios. A newly introduced concept based on swarm theory, Histograms of Oriented Swarms (HOS), is applied to capture the dynamics of crowded environments. HOS, together with the well known Histograms of Oriented Gradients (HOG), are combined to build a descriptor that effectively characterizes each scene. These appearance and motion features are only extracted within spatiotemporal volumes of moving pixels to ensure robustness to local noise, increase accuracy in the detection of local, non-dominant anomalies, and achieve a lower computational cost. Experiments on benchmark datasets containing various situations with human crowds, as well as on traffic data, led to results that surpassed the current state of the art, confirming the method's efficacy and generality. Finally, the experiments show that our approach achieves significantly higher accuracy, especially for pixel-level event detection compared to State of the Art (SoA) methods, at a low computational cost.**

***Index Terms*—swarm intelligence, crowd, anomaly, traffic.**

I. Introduction

THE widespread use of surveillance systems in roads, stations, airports or malls has led to a huge amount of data that needs to be analyzed for safety, retrieval or even commercial reasons. The task of automatically detecting frames with anomalous or interesting events from long duration video sequences has concerned the research community in the last decade. Event, and especially anomaly detection in crowded scenes is very important, e.g. for security applications, where it is difficult even for trained personnel to reliably monitor scenes with dense crowds or videos of long duration. Numerous methods have been proposed to assist in this direction.

The analysis of motions and behaviors in crowded scenes constitutes a challenging task for traditional computer vision methods, as barriers like occlusions, varying crowd densities and the complex stochastic nature of their motions are difficult to overcome. Computational cost is one more complicating factor, as it has to be kept within reasonable limits. In many practical situations, it is crucial to analyze crowded scenes in real time, or at least as fast as possible, considering the fact that security personnel should act quickly if something seems to be "not as usual". Furthermore, the ambiguity of the term "anomaly" sets its own limitations in our effort to identify it, as there is no commonly accepted definition, and it varies significantly depending on the given scenario. This means that an "anomaly" pattern in one video sequence may often be part of the "normal" pattern of another. In order to address these issues, we define as "anomalies" the events that display a low probability of occurring based on earlier observations.

We deal with the challenging problem of detecting abnormal patterns in videos of crowded scenes that emerge as spatiotemporal changes, both in motion and appearance. An appearance-related anomaly would be, e.g. a bicycle passing through a crowd. Moreover, sudden changes in velocity, like an abrupt increase of its magnitude and the dispersion of individuals in the crowd are detected, indicating that something unusual and potentially dangerous may have occurred.

In this work we propose a novel method for anomaly detection and localization that incorporates both motion and appearance information. We introduce a descriptor created from Histograms of Oriented Gradients (HOG) to capture appearance, and the newly introduced Histograms of Oriented Swarms (HOS), to capture frame dynamics. Swarm intelligence has been used in the past only in the framework of Particle Swarm Optimization (PSO) in [1], where PSO optimizes a fitness function minimizing the interaction force derived from the Social Force Model (SFM). However, in our work, swarms are used in a very different way: the core idea is to construct a prey based on optical flow values over a specific time window and deploy a compact swarm flying over it to acquire accurate and discriminative information of the underlying motion. The agents' motion is determined by forces acting on the swarm (Sec. IV), which, unlike [1], do not correspond to the SFM, but are used to determine the swarm motion and location.

Thus, this work introduces an innovative deployment of swarm intelligence, which, together with the HOG descriptor, forms a new feature capable of successfully determining a region's "normality" in an SVM framework. In order to capture "anomalies" appearing in a small part of the frame, our algorithm is applied only on regions of interest, and temporal information is incorporated to improve accuracy. Even though benchmark datasets of human crowds were mainly used for the algorithm's validation, results on other kinds of videos of crowded scenes, e.g. traffic, reveal that the proposed method can be extended and generalized to different scenarios. The experimental section shows that our algorithm outperforms state of the art (SoA) algorithms in accuracy and at a low computational cost. Our contribution can be summarized as follows:

1) Swarms are used in an original way, via Histograms of

[1]Aristotle University of Thessaloniki,[2] Information Technologies Institute, Multimedia Knowledge and Social Media Analytics Laboratory, CERTH

Oriented Swarms (HOS) that are introduced to characterize crowd motion for anomaly detection. They lead to credibly filtered flow in videos of crowds, resulting to very few noisy flow values. Thus, swarm intelligence captures the motion of crowded scenes in an efficient way that can be extended to other types of videos.

2) The method can be efficiently applied even when the motion in the crowded scene is non-uniform in space and time, and "anomalies" appear locally in a changing context. This is shown in the experiments of Sec. VI on the complete UCSD dataset, where our method's accuracy for pixel level anomaly detection surpasses the SoA.

II. State of the art

Even though significant research has taken place on event and anomaly detection from static cameras [2], [3] the majority of these works address non-crowded scenes, where detailed visual information can be exploited for each individual. However, real-world surveillance scenarios often involve crowds of people or dense traffic, where such information cannot be easily extracted with traditionally used methods. Therefore, a number of different approaches have been proposed to handle these situations. Several interesting works [4], [5], [6], [7] introduce tracking methods, nevertheless, they seem to be effective only in videos with crowds of low density, as tracking is otherwise hindered, due to the high degree of occlusions, while their computational cost is also greatly increased. As a result, the current SoA focuses mostly on analyzing entire frames spatially, temporally or both. Existing methods can be classified in two main categories: those that use only motion information to detect an abnormality in the scene, and those that use both appearance and motion information to describe the scene dynamics.

In the first category, Wu et al. [8] use chaotic dynamics in particles' representative trajectories as a means to build a model capable of locating an outlier that moves with a different pattern. Even though this method works for very dense videos where a global motion pattern exists, it is unable to detect local abnormalities that take place in a small region in the frame, or in the absence of a global pattern. Activity recognition based exclusively on trajectories is also proposed by [9]. However, this method is only based on motion information, completely ignoring the existence of "interesting" activities that exhibit a typical motion pattern. In the same category, Mehran et al. [10] use the Social Force Model (SFM) to describe a crowd's normal behaviour based on motion characteristics, while Cui et al. [11] make use of interaction energy potentials derived from the interest points' position and velocity. In [12], the min cut/max flow algorithm is used to define each block's dominant direction and crowd motion segmentation is performed by training the algorithm separately, for each spatial location. That method also only relies on motion patterns to detect an anomaly, completely ignoring appearance information. Another interesting work in the same domain, is that of Cong et al. [13], who introduce a sparse reconstruction cost to measure the normality of the testing sample, considering dictionary learning methods. Saligrama et al. [14] extract local low-level motion descriptors and utilize score functions for anomaly detection, derived from local nearest neighbour distances. A different approach is used by Adam et al. [15] who use fixed monitors to extract local low level features and determine a preset threshold for each monitor in order to declare an alert, while Kim et al. [16] propose a Markov Random Field model in a Bayesian framework for the final inference. A Gaussian mixture model is used by Ryan [17] for anomaly detection in crowded scenes based on textures of optical flow in 3D volumes. 3D Gaussian distributions that characterize the underlying motion patterns of spatiotemporal cuboids are used in [18]. In this work, KL divergence is used as a distance measure to identify similar cuboids in the same location and new prototypes are created accordingly. Observations are then only evaluated according to distributions occurring in the same spatial location by creating a single HMM for each location. As a result the method proposed leads to many false positives in sparse videos, as the number of frames needed to work properly is huge, and it has to cover every region separately in order to train each HMM efficiently. Thus, that method is appropriate for crowded scenes of a very high density, but cannot handle videos of crowds with middle to low density, which are often captured by surveillance cameras. The same 3D gradient features are used by Lu et al. [19] in a different framework: they propose the use of sparse coefficients to fit new data to a previously learned dictionary. Sparse combination learning is introduced instead of searching the whole search space as classic sparsity-based methods do, thus greatly reducing the computational cost. The method exhibits remarkably high speed performance but at the expense of its accuracy, as shown in the experiments. Finally, an almost real-time algorithm is suggested in [20] for event detection, based on the clustering statistics derived from moving particles.

A common problem that is encountered by all the methods mentioned earlier, is their inability to successfully detect anomalies that move similarly to the "normal" motion pattern, as they rely solely on motion characteristics. A second category of methods tackles this issue by incorporating appearance information as well. One work that stands out in this category is that of [21], that uses mixtures of dynamic textures to describe each 3D cuboid extracted from video sequence and detect temporal and spatial abnormalities. However, the computational cost of that algorithm, around 25 sec per frame, makes it prohibitive for many applications. An improved version of this method, with a lower computational cost, that is similar to ours, is found in [22]: that method's accuracy is also improved, but it still remains lower than ours as the experiments in Sec. VI show. The joint modelling of appearance and dynamics is also proposed by Ito et al. [23] for detecting interesting events via density estimation ratio to classify frames in two classes, normal or abnormal. Despite the applicability of that method to many scenarios, it is only suited for detecting events that occur over the entire frame (e.g. global changes in motions, scene changes etc.) and it misses local abnormalities. Another work that uses features based on both motion and texture is that of [24]. In that work, the input image is split into nonoverlapping cells and features based on motion, size and texture are extracted and are fed into two classifiers. The main drawback of the method is that

the classification of each cell is determined by a pre-defined threshold, which makes the method sensitive to input video. Another interesting work is that of [25] which proposes a method of detecting abnormalities indirectly after establishing a complete interpretation of the foreground, by using a set of hypotheses. Afterwards, anomalies are defined as those hypotheses that are required to explain the foreground but which themselves cannot be explained by normal training samples. That method works efficiently on the UCSD dataset, however, the need for interpretation of all foreground objects may arise difficulties in more dense crowd datasets. Results on the UCSD dataset provided in Sec.VI, also show that our method provides better accuracy. Finally, the work of [26] uses densely sampled spatiotemporal video volumes at each pixel location to construct a low level codebook and bag of video words is used to detect anomalous events. However, that method only uses the HOG descriptor to capture both motion and appearance characteristics omitting essential information that could led to better results. As the experiments show, our approach outperforms all the methods described above, in the important pixel level criterion on the UCSD dataset, which is used by most SoA works, making it more suitable for spatiotemporally local anomaly detection.

The deployment of swarms involves the calculation of internal and external interaction forces, characterized by a number of parameters. In this work, an analytical description of the method's robustness to various parameter values is presented in Sec.VI-A. Currently, new methods are being developed for the evaluation of parameter sensitivity in [27], which may be taken into account in future extensions of this work.

This paper is organised as follows: Sec. III describes the problem formulation, Sec. IV extensively presents the mathematical background of the new descriptor, while anomaly detection and localization is described in Sec. V. Finally, a detailed experimental evaluation is discussed in Sec. VI and conclusions are summarized in Sec. VII.

III. Problem Formulation

In this work, we address the problem of detecting dynamically changing anomalies in both space and time in videos with crowds of varying densities. In order to effectively capture these anomalies for a wide range of situations, we incorporate both motion and appearance features. Our algorithm uses data derived from automatically extracted regions of interest (ROIs) instead of entire video frames, so as to only process pixels containing information relevant to the event taking place, while at the same time achieving a lower computational cost, fewer false alarms, greater precision and successful spatiotemporal localization of anomalies, both on a global and local scale.

In order to extract the ROIs, we apply background subtraction using weighted moving mean [28], as it has been shown to be robust and reliable, however other SoA background subtraction methods like Gaussian Mixture Models (GMMs) could also be used, leading to equivalent results. We define interest points on a dense grid in the resulting foreground and ROIs are described as rectangular areas of fixed size around each interest point. The size of the ROIs is determined at the beginning of each set of experiments, and depends on the camera viewpoint for each dataset. Due to the static nature of surveillance cameras, the block size needs to be set only once for each camera, or in our case for each dataset, and thus does not affect our algorithm's generality. For the UCSD dataset, a ROI of 20×20 pixels is used, as it is large enough to capture activity/appearance related details, but is not too large, so as to include noisy information in the descriptor.

Once ROIs are extracted, the interest points in them are tracked until the next frames using the KLT tracker, while the foreground grid is continuously updated, with new interest points defined in each new frame's foreground area. The resulting ROIs and the interest points in them are considered informative and are retained if at least 60% of that ROI contains motion, otherwise that interest point and its ROI are considered to be noisy and are ignored. The ROI needs to contain at least 60% moving pixels in order to be as informative as possible; if a ROI contains fewer moving pixels, noisy (motionless) data will also be taken into account, while if it is required to contain more moving pixels, potentially informative interest points may be ignored.

Spatiotemporal feature extraction from ROIs follows for a particular time window, to acquire descriptors that effectively describe the video dynamics, and help identify both local and global abnormalities. We consider both motion and appearance features, as their combined use allows the detection of anomalies, i.e. deviations of motion and/or appearance from usual patterns, leading to a generally applicable method. An overview of the procedure for extracting the descriptor is depicted in Fig.1(a). The stages for modelling appearance and motion are discussed in more detail in the sequel.

A. Appearance modelling

In order to extract the appearance characteristics of a video sequence, the Histograms of Oriented Gradients (HOG) proposed in [29] are used, as the HOG descriptor has several advantages over other appearance features: it is color invariant as it uses gray scale images, and is also invariant to illumination and local geometric transformations as a result of the normalization that takes place. At the same time, it effectively captures the local edge and gradient structure, so it can distinguish variations in appearance even in small areas of the image. The implementation of HOG that is adopted is that of [30], as it creates direction invariant HOGs by following a mirroring technique, where mirrored shapes are mapped into the same bin. Direction invariant appearance features (HOGs) decrease intra-class variation, e.g. for walking, which is the predominant activity in human crowds, resulting in similar appearance descriptors for motions in opposite directions. This leads to more robust appearance descriptors that are suitable for the needs of anomaly detection in crowded videos, which can describe, for example, the density or sparseness of a crowd more effectively by ignoring directionality (which is not relevant for appearance).

The HOG descriptor is applied in ROI blocks that are tracked over time and are extracted as described in the previous section, so the final HOG descriptor for each block also incorporates temporal information. The procedure for this computation is as follows: each block k is first divided into 2×2 cells as suggested in [29] for a more detailed description

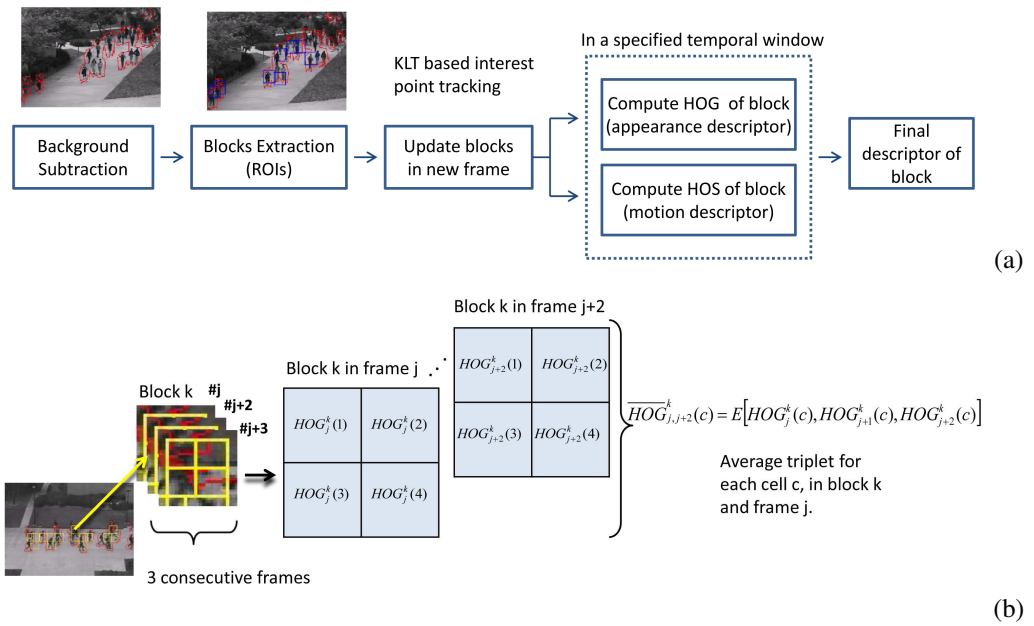


Fig. 1. Problem formulation. (a) Overview of final motion-appearance descriptor calculation. (b) Extraction of appearance descriptor (HOG). Each block is divided into 4 cells and HOG histograms are calculated for each of them. The block is tracked over time and the final HOG descriptor results from the average of consecutive triplets after a normalization step. The $HOG_j^k(c)$ symbol represents the HOG histogram, calculated from the c^{th} cell of block k, at frame j.

that also takes spatial location information into account and mitigates the effects of local noise. For example, if occlusions are present in a ROI, its division into 2×2 cells may limit their presence to only one of the cells, instead of the whole area, leading to a less noisy appearance descriptor. The division of a block into 2×2 cells was chosen, as it was found by Dalal et al. [29] to retain a sufficient level of detail for describing appearance. A weighted histogram of gradients is then created for each cell using 9 bins, corresponding to the gradients' orientation. The HOG of the c^{th} cell ($1 \leq c \leq 4$) in block k of frame j is thus represented by $HOG_j^k(c)$, of dimension 1×9. Each histogram is normalized and the 4 resulting cell histograms are concatenated, forming a 1×36 block descriptor, which is also normalized for noise elimination. Once HOGs for each block are calculated for all frames in the temporal window under examination, they are averaged over 3 consecutive frames so as to include richer temporal information and at the same time achieve temporally local noise reduction. The final appearance descriptor is thus a concatenation of a 3 frame average for each cell c in block k:

$$\overline{HOG}^k_{j,j+2}(c) = E[HOG^k_j(c), HOG^k_{j+1}(c), HOG^k_{j+2}(c)] \quad (1)$$

This means that a 15 frame time window will result in 5 concatenated triplets of 1×36 descriptors, resulting in a 1×180 final spatiotemporal appearance descriptor. The entire process for extracting HOG descriptor is depicted in Fig.1(b). For simplicity of notation, in the sequel, the HOG descriptor of Eq. (1) for block k, averaged over frames j to $j+2$ will be represented as $\overline{HOG}_{j,j+2}$ including the average over all 4 cells.

B. Motion modelling using HOS descriptor

This work introduces a novel method for capturing crowd dynamics based on the application of swarm intelligence, which is used to build a novel motion descriptor. Swarm intelligence in computer science is inspired from the behaviour and characteristics of real swarms encountered in nature. Swarms are comprised of individuals, which act autonomously, while following the specific rules of a swarm and interacting with each other. Although the decisions of a swarm's individuals take place locally, their aggregated behaviour can match events in crowded environments, which makes them relevant in many applications, as shown in Sec. VI.

Swarm based methods have been used in the literature for image filtering and noise reduction [31], but their incorporation for the analysis of motion in videos is an original concept first presented in [32]. The core idea is the monitoring of movements in crowded scenes by a swarm of agents "flying" over them, to capture their dynamics in a collective way while also taking motion history into account. Swarms are thus deployed and the agents' positions are extracted from their accelerated motion, derived from the forces acting on the swarm as described in Sec. IV. They are then used to form Histograms of Oriented Swarms (HOS), which are used to capture the ROIs' underlying motion and detect anomalous events in them. The main concepts of our swarm descriptor are presented in the following section.

IV. Swarm Modelling for Crowds Dynamics

In our implementation, we adopt physics-based modelling of crowded scenes, as their properties are highly correlated with those of a swarm in nature. The swarm model that is used is based on the general theory described in [31] and on the behavior of natural swarms, consisting of predators, which "fly" over the "prey", following its dynamics. In [31], swarm modelling is used to filter noise in images, whereas in this work it is deployed to better characterize the highly complex and stochastic motion information from videos of crowds. In our implementation, swarms comprise of agents and a prey:

the agents "track" the prey, but also interact with each other, as they would in nature. Hence, agents ("predators") are subject to three types of forces: "physical" forces, like inertia and friction, interaction forces between them, and external forces dependent on the prey. Interaction forces ensure the cohesion of the swarm of agents, friction forces maintain elementary memory of the agents' velocity, while external forces depend on the characteristics of the prey being tracked.

Consequently, in this approach, swarm intelligence maps the motion information into a more informative space by efficiently tracking the motion represented by the prey. Agents filter the prey motion, avoiding false alarms and local noise caused e.g. by occlusions or outlier optical flow values. The prey corresponds to the values of the variable that we want to leverage in the discriminative process. In our case, we are interested in the extraction of motion features via the swarm modelling, so optical flow (OF) values are used as a prey, as detailed in the next section. Thus, the use of swarms is expected to lead to better results than when using OF information alone, as they can capture the most important aspects of crowd behaviour while circumventing the effects of local noise, occlusions and the overall complexity of motion in crowded scenes.

A. Prey Generation

The prey that is tracked by the swarm comprises of OF magnitude values of pixels lying inside ROIs, instead of their luminance, which is the case in [31]. Hence, the number of prey in each frame varies, as it is equal to the number of ROIs in the frame. In this section we describe how prey data is extracted, namely how it is mapped to be tracked by agents. As mentioned previously, ROIs correspond to rectangular areas around each interest point containing a fixed number of n pixels. In order to form the prey for a ROI in a temporal window of m frames, we consider the pixels of each ROI sequentially over time. Each pixel at position i in a particular ROI of frame j has OF magnitude equal to O_{ij}, where $1 \leq i \leq n$ and $1 \leq j \leq m$. For the prey construction, we consider the i^{th} pixel's OF sequentially over time. The OF magnitude is used to determine the prey's position x_p as follows:

$$x_p(t) = O_{ij} \tag{2}$$

where t is a spatiotemporal index that spans all n ROI pixels over m frames, so that $1 \leq t \leq n \cdot m$. The selection of the sequence of pixels for prey construction is very important for capturing meaningful temporal information. As a result of the above processing, the final prey position data that the swarm will track for all pixels $1, \ldots, nm$ in each ROI cuboid is defined as:

$$[x_p(1), \ldots, x_p(nm)] = [O_{11}, \ldots, O_{1m}, \ldots, O_{n1}, \ldots, O_{nm}], \tag{3}$$

where the O_{ij} represent the OF magnitude. This process of prey construction is illustrated in Fig.2.

After each prey is extracted, a swarm of agents is generated to characterize its motion, leading to more accurate analysis of its behavior, as also shown in Sec. VI, where using swarms leads to better classification than when only using the OF.

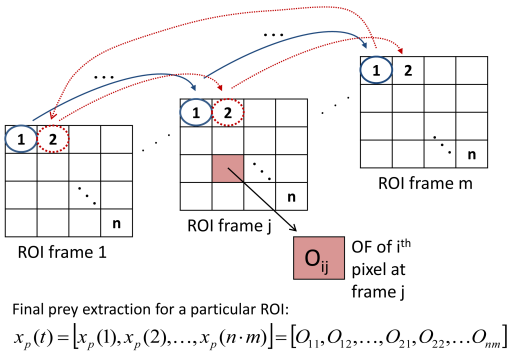


Fig. 2. Prey extraction in a m frame window occurs sequentially in a cuboid of m frames. First, m "OF values" of the 1^{st} pixel are taken into account, then m instances of the 2^{nd} pixel and so on, until m instances of the n^{th} pixel, where n is the number of pixels in each ROI.

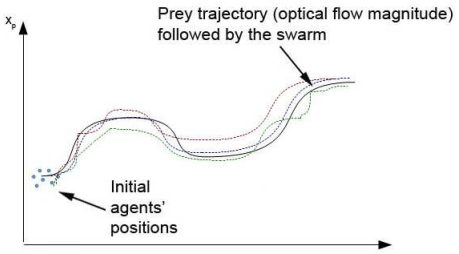


Fig. 3. A swarm following prey: dashed lines show agents' trajectories while the continuous line depicts prey trajectory.

The orientation of each pixel's OF is also taken into account for the construction of swarm histograms: the correlation of swarm behavior with OF orientations is high, as swarm behavior (agents' positions and accelerations) for each t is determined by the OF magnitude and orientation of the corresponding pixel. In the following section, we describe agents' dynamics which are then used to create HOS.

B. Extraction of Forces

In this section we present the manner in which the agents operate, i.e. the way they "fly over" the prey and track it. Agents are groups that we define to track the prey and characterize its state: they are initially located in random positions, which change over time according to agent-prey forces, agent-to-agent forces and friction forces presented here. The result of these forces' interactions is the accelerated motion of the agents, which is affected and formed according to prey behaviour. These forces are inspired by crowd psychology and the analysis of movements of individuals in crowds [33], matching real world behaviors of people (or other entities, like cars or animals) in crowded situations: for example, when agents are too close to each other, repulsive forces develop between them, while the opposite occurs (attraction forces develop) when they are at a large distance, ensuring the cohesion of the swarm of agents. An illustrative example of the way the swarm follows the prey is given in Fig.3.

The interaction force F_{neigh} is the force between agent i and all other agents of the swarm found in the neighbourhood of i, at a distance smaller than ρ. It can be attractive or repulsive depending on the agents' distances in the swarm, and its role

is to prevent collisions of the agents and to ensure swarm cohesion. It is determined by the following equation:

$$F_{neigh}(i,t) = \sum_{j \in V_i} F_{int}(i,j,t), \tag{4}$$

where F_{int} is the interaction force between each agent i and all other agents j of the swarm in its vicinity V_i, determined as the agents whose distance from i is smaller than ρ. We define each agent i's position at t as $x_i(t)$, so $F_{int}(i,j,t)$ is:

$$F_{int}(i,j,t) = \frac{\beta \cdot (x_i(t-1) - x_j(t-1))}{d(i,j)^2} \tag{5}$$

when $|x_i(t-1) - x_j(t-1)| \leq d_{min}$ and

$$F_{int}(i,j,t) = \frac{-\alpha \cdot (x_i(t-1) - x_j(t-1))}{d(i,j)^2} \tag{6}$$

when $d_{min} < |x_i(t-1) - x_j(t-1)| \leq \rho$.

Here $d(i,j)$ denotes the distance between agents i and j, $x_i(t-1)$ is the previous position of agent i and α, β are weighting parameters set equal to 1, as agent-prey distances are found to be a sufficient measure of internal force strength, and do not need to be amplified or compressed. Nevertheless, the effect of these parameters on the anomaly detection accuracy is investigated in detail in Sec. VI-A, where experiments are run for a very wide range of values of α, β, showing that their values indeed do not greatly affect the outcomes of our method. The value of d_{min} ($0 \leq d_{min} \leq \rho$) sets the boundary that determines if the interaction force is attractive or repulsive.

The second force is the velocity dependent friction force F_{fric} that acts on each agent i, offering to the swarm a type of elementary memory. It depends on the velocity the agent formerly had, corresponding to the previous prey location $t-1$:

$$F_{fric}(i,t) = -\mu \cdot \dot{x_i}(t-1) \tag{7}$$

where $0 \leq \mu \leq 1$ is the friction coefficient and $\dot{x_i}(t-1)$ the former velocity of agent i. After experimentation, $\mu = 0.4$ is found to provide the best tradeoff between tracking speed, smoothness and accuracy as shown analytically in Sec. VI-A.

Finally, the swarm is driven across the frame mainly by the external force F_{ext} given by Eq. (8) below, which is an elastic force that makes the swarm follow the trajectory of the prey, as every agent is attracted to it. It is an agent-prey force that guides the swarm "over" the prey, so it moves in parallel with it, in our case with the optical flow magnitude:

$$F_{ext}(i,p,t) = \lambda \cdot (x_p(t-1) - x_i(t-1)). \tag{8}$$

It is clear that the external force between the swarm agents and the prey pixels is directly dependent on their relative position values (in practice the OF magnitude), with the force becoming weaker as the swarm agent diverges from its prey. This force is similar to the restoration force of a harmonic oscillator, so λ represents the positive spring constant, whose value is equal to $\lambda = 1$ in the experiments.

C. HOS Descriptor

In order to form the HOS descriptor, we examine the evolution of the agents' positions, determined by prey motion patterns and the forces affecting the agents. We modify Newton's second law of motion by inserting an elementary parameter γ that takes into account the previous velocity values, as in Eq. (9) shown below. Then, the acceleration $\ddot{x_i}(t)$ of each agent i at position $x_i(t)$ is given by the vector sum of all forces acting on it, considering the fact that an agent's mass equals 1, along with the γ-weighted velocity of the previous time instant. Thus, the acceleration of each agent is given by:

$$\begin{aligned} \ddot{x_i}(t) &= (\gamma - 1)\dot{x_i}(t-1) + F_{neigh}(i,t) \\ &+ F_{fric}(i,t) + F_{ext}(i,p,t), \end{aligned} \tag{9}$$

where γ is a memory parameter, relating past values of the velocity with the current acceleration. When pixel flow undergoes a sudden change, it will be captured by the forces acting on it in Eq. (9), so the influence of its previous value will be mitigated. As a result of the forces, the swarm follows accelerated motion and the velocity of agent i at location x_i is:

$$\dot{x_i}(t) = \gamma \cdot \dot{x_i}(t-1) + \delta \cdot \ddot{x_i}(t), \tag{10}$$

where δ constitutes a timestep parameter, essentially forming an autoregressive process with flow values changing slowly over space and time. Therefore, the positions of agents are continuously updated and their new values are given for each spatiotemporal location t by the following equation:

$$x_i(t) = x_i(t-1) + \delta \cdot \dot{x_i}(t-1) + \frac{1}{2}\ddot{x_i}(t)\delta^2. \tag{11}$$

Swarm agents' positions are randomly generated for the first prey position $t = 0$, and their speeds and accelerations are initially set to zero. Their values change over time depending on prey locations, as described above, and the forces affecting the agents. During training, ROIs are extracted and the pixel OF in them is examined and tracked by the agents. We then compute the average of swarm agents' positions of Eq. (11) for each t, and follow a process similar to the HOG extraction of Sec. III-A to extract weighted histograms of agents' positions (HOS), according to the corresponding OF orientation. As in Sec. III-A, each ROI (block) around each interest point, is partitioned into 2×2 cells, and the positions of the swarm agents that follow this particular block establish a weighted histogram of 18 bins according to the OF orientation in each cell. Subsequently, these 4 histograms are concatenated to form the block's HOS. In order to include temporal information, the final motion descriptor contains histograms of subsequent frames, averaged in triplets over each time window.

V. Anomaly Detection and Localization

Appearance and motion descriptors are combined to form the final descriptor for anomaly detection. In a time window of m frames, average triplets of HOG and HOS are consecutively concatenated, resulting in the feature vector of Eq. (12):

$$f = \{\overline{HOG}_{1,3}, \overline{HOS}_{1,3}, \ldots, \overline{HOG}_{m-2,m}, \overline{HOS}_{m-2,m}\} \tag{12}$$

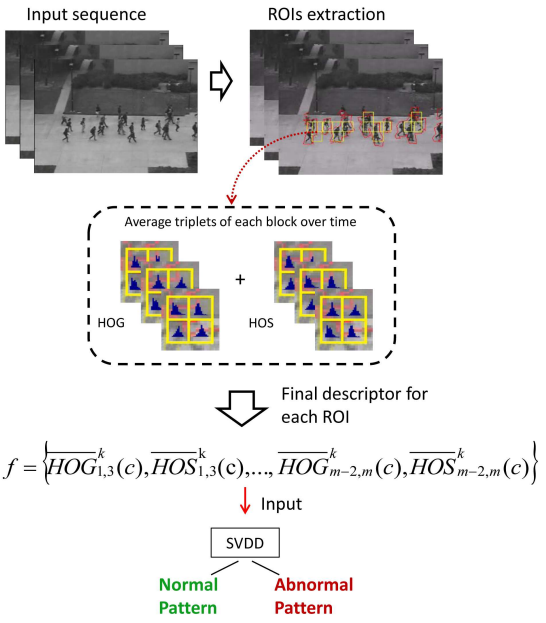


Fig. 4. Overview of method proposed in a time window of m frames.

Here, $\overline{HOG}_{m-2,m}$ is the average of HOG histograms corresponding to a block for frames $m-2$ to m and $\overline{HOS}_{m-2,m}$ is the average of the corresponding HOS histograms taken from Eq. (1). The overall process takes place in each ROI and it is depicted in Fig.4. A normalization step takes place to form the final descriptor so as to achieve scale invariance.

Afterwards, a Support Vector Machine (SVM) is used to determine each region's normality. SVMs are used, as they generally exhibit good performance relatively to other machine learning methods and are also fast to run, for reliable real time detection. Furthermore, they are able to handle large data sets, which generally appear in real life situations. Because of the infinite number of "anomalies" that can be derived in each case, it is impossible to provide examples of all possible anomaly classes, so a one class classifier is chosen. This way, we provide our system exclusively with normal situations, aiming to identify any irregularities deviating from the normal pattern. This leads to a more accurate and general classifier capable of detecting different kinds of anomalies, even when appearing for the first time in the dataset.

The Support Vector Data Description (SVDD) method of [34] was chosen, as it is known to be best suited for outlier detection. According to this approach, spherical boundaries are used instead of planar ones around the provided data of the training set. The goal is to enclose nearly all n training examples in a hypersphere with center o and the smallest possible radius R, with the outliers lying outside this sphere. Thus, its purpose is to minimize the function:

$$min_{R,o}\left(R^2 + C\sum_{i=1}^{n}\dot{\xi}_i\right) \tag{13}$$

subject to:

$$\|\nu_i - o\|^2 \leq R^2 + \dot{\xi}_i, \quad \dot{\xi}_i \geq 0 \quad \forall i \tag{14}$$

In order to create a soft margin and allow for outliers in the training set, slack variables $\dot{\xi}_i$ and a penalty parameter C describe the hypersphere. By using Lagrange multipliers to solve Eq.(13), subject to Eq.(14), with a Gaussian kernel, we conclude that a new "test object" z is accepted when:

$$\|z - o\|^2 = \sum_{i=1}^{n}\lambda_i \exp\left(\frac{\|z-\nu_i\|^2}{\sigma^2}\right) \geq -\frac{R^2}{2} + C_R \tag{15}$$

otherwise z is an outlier. C_R is a constant, dependent only on support vectors ν_i, while λ_i are the Lagrange multipliers and σ represents the standard deviation of the Gaussian kernel.

After training, localization is straightforward, as descriptors are estimated spatially in specific ROIs around interest points. Our algorithm checks each frame's ROI independently, infers about its normality and then notifies the system. Hence, our method is capable of dealing with non-uniformly moving and evolving crowds, as the descriptors are examined and characterized separately in each ROI. It can accurately localize different anomalies in a wide range of videos, from human crowds to traffic, as the experiments that follow demonstrate.

VI. Experiments

In order to evaluate the effectiveness of our method, we applied it on four benchmark datasets of surveillance where different kinds of anomalies were detected. Our algorithm's speed and accuracy on a frame and pixel level were calculated and compared with the SoA, demonstating its effectiveness. An extensive sensitivity analysis has also taken place to examine the effect of varying all parameter values, showing that they do not significantly affect the accuracy of the results.

As mentioned in Sec. III, temporal information is exploited by extracting features over a specific window in time. The length of the window should be large enough to contain sufficient information and, at the same time, as small as possible to avoid undesirable delays during the detection process. Hence, we use a temporal window length that depends on the frame rate and the underlying dominant motion, which in our case is the mean walking frequency of a pedestrian. As an example, Fig.5 depicts the optical flow values of a pedestrian for the "ped2" dataset: it can be seen that the entire motion displays periodicity over time, and therefore a temporal window of 15 frames sufficiently captures the entire cycle for this case. Averaging of the extracted features over time to form triplets follows, as detailed in Sec. III-A, to mitigate the effects of local noise and include richer temporal information in the resulting descriptor.

The size of the extracted blocks is also scene-related and dependent on the camera view, so as to contain adequate

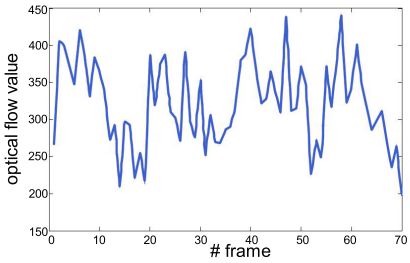


Fig. 5. Walking frequency of a pedestrian in ped2. A time window of 15 frames can capture the whole period of the motion.

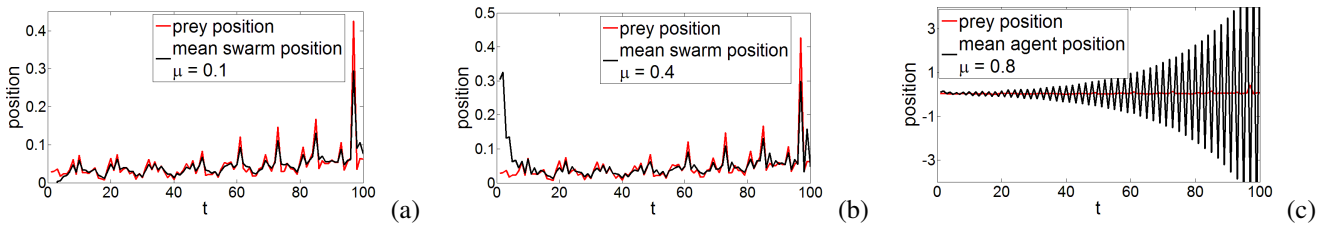


Fig. 6. Effect of different μ values on agents tracking the prey for the first 100 timesteps, with the other swarm parameters kept stable: (a) $\mu = 0.1$, (b) $\mu = 0.4$, (c) $\mu = 0.8$. For $\mu = 0.8$ the system is unstable and uncontrollable oscillations appear, making values of $\mu \geq 0.8$ unsuitable for our system.

information. As explained in Sec.III, its value is determined in the beginning for each dataset and is fixed for all the shots made by the same camera. In this work, we use a ROI of 20×20 for the UCSD dataset as it is small enough to capture anomaly localization, but at the same time large enough to capture useful information about the entities moving in the scene (people, bicylces etc in the UCSD data). The ROI size was determined experimentally, however small variations in its size do not significantly affect the algorithm's accuracy.

Because of the static nature of surveillance cameras, the size of the temporal window and ROI blocks are only set once for each camera, or in our case once for each dataset, and thus do not compromise the generality of our algorithm. The parameters defining the forces affecting the motion of the swarm agents also remain the same in all experiments. In the next section, the method's sensitivity to different parameters affecting the swarm generation is analyzed in detail.

A. Sensitivity Analysis

The parameters α, β of Eqs. (5) and (6) are positive constants representing attraction/repulsion internal forces. In our experiments we use a common value for them, as we want the impact of repulsive and attractive forces to be the same ($\alpha = \beta$). Eqs. (5) and (6) show that increasing their values leads to larger internal forces, which move agents by larger distances, either taking them further away or closer to each other. However, in our method, the center of mass of agents' positions is used to determine the swarm's histogram HOS, which depends on the relative position of each group of agents, rather than the actual values of these internal forces. For lower values of α, β, the individual agents move less with respect to each other, and for larger α, β they move more, but on average the swarm's position stays the same. This is also shown in Fig.7, where the Equal Error Rate (EER) is shown for the UCSD dataset for an extremely wide range of values of $\beta \in [0, 10000]$: the EER fluctuates very little, even for these very large differentiations in β, proving that algorithm's performance is barely affected by these constants. Subsequently, we set $\alpha = \beta = 1$ for all experiments.

Fig.6 shows the effect of changing the friction force parameter μ of Eq. (7). The parameter μ is essentially the equivalent of the spring constant in the definition of a force, with a higher value indicating the presence of more friction. For a lower μ ($\mu = 0.1$), it can be seen that less friction leads to faster but less smooth tracking, as the effect of previous values, is given less weight. Lowering the value of μ can make the friction force more prone to errors, derived from OF values, while increasing its value can result in unstable oscillations, with

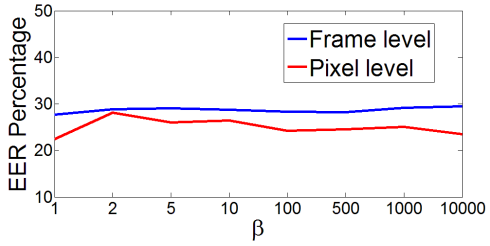


Fig. 7. Sensitivity analysis for β. The EER for UCSD is depicted for different values of β: the EER fluctuates very little, even for very large differentiations in β, proving that algorithm's performance is not significantly affected by β.

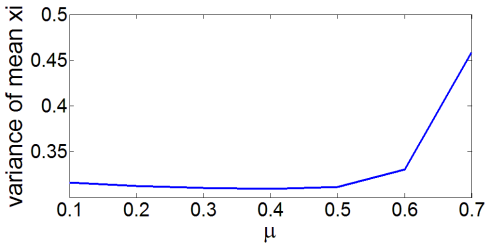


Fig. 8. Variance of the mean position of agents tracking the prey for μ.

F_{fric} reflecting temporally local noise, as Fig.6(c) depicts. In order to demonstrate the smoothness of the resulting tracking as a function of the parameter μ, we plot the variance of the agents' positions $x_i(t)$ for $0.1 \leq \mu \leq 0.7$ in Fig.8, for a video sample from the UCSD dataset. It can be seen that, for smaller values of μ, the variance does not show significant changes, while for $\mu \geq 0.7$ it rises dramatically. For these reasons, and based on the experimental data shown in Fig.6, we use $\mu = 0.4$, even though values of $\mu \leq 0.6$ do not really greatly affect the accuracy of our system.

The parameter γ constitutes an elementary memory parameter that determines the effect of the previous agent's velocity on the current agent's acceleration. In Fig.9(a), the swarm's position in relation to the prey is depicted for different values of γ: for larger values of the parameter, the swarm moves more quickly, but less smoothly. For values greater or equal to 0.8, the system starts to oscillate, so we choose values for γ in the range $[0, 0.7]$. In Fig.9(b), the EER error for the UCSD dataset is depicted for values of γ in this range, where it can be seen that the EER at the frame level is barely affected, while the pixel level EER varies by about 10%, especially for $\gamma > 0.4$. Hence, in our experiments we use $\gamma = 0.4$, as it leads to better results, even though other values of $\gamma \in [0, 0.7]$ do not cause significant fluctuations in the method's performance.

One last parameter that needs to be set for the swarm formulation is δ. This is equivalent to a timestep parameter,

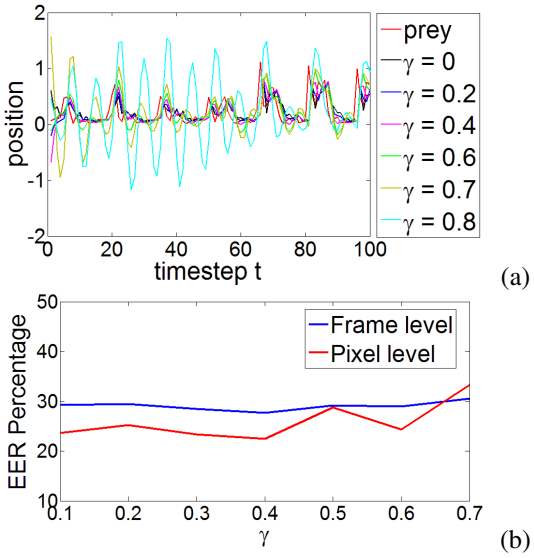


Fig. 9. Sensitivity analysis for γ. (a) Swarm position in relation to prey (red line) for different values of γ, (b) EER for UCSD "ped1" dataset for $\gamma \in [0, 0.7]$. Both frame level and pixel level EERs are shown.

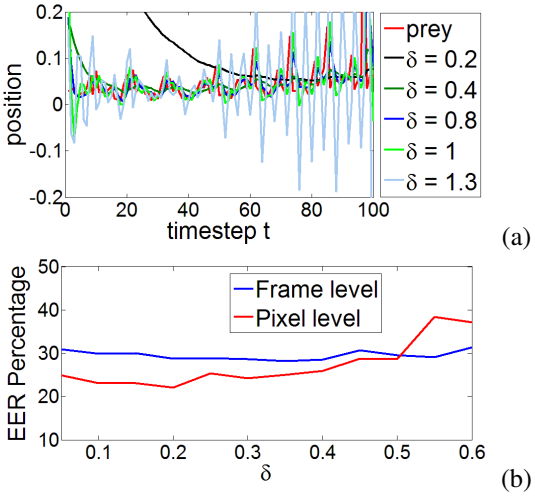


Fig. 10. Sensitivity analysis for δ. (a) Swarm's position in relation to prey (red line) for different values of δ, (b) EER for UCSD "ped1" dataset for $\delta \in [0.1, 1.2]$. Both frame level and pixel level EERs are shown.

as can be seen in Eqs. (10), (11), that corresponds to the time that each agent needs to move from its previous position to the current one. Larger values of δ make the swarm respond more quickly to the prey, but can produce overshoots as Fig.10(a) depicts for $\delta \geq 1.3$. On the other hand, very small values $\delta \leq 0.2$ lead to a system that cannot follow the prey's trajectory. In our experiments we choose $\delta = 0.4$, as it is shown that with this value the swarm is capable of following the prey to a sufficient extent, while maximizing our system's performance. Fig.10(b) shows the sensitivity of our algorithm to different values of δ.

It should be pointed out that in all the experiments the parameters described above remain fixed to their optimal values. As the extensive analysis proved, after defining the range of each variable that ensures system stability, our algorithm's performance is not particularly influenced by further variations in these parameter values.

Finally, the number of agents forming the swarm is fixed to 5, as it is empirically found that this number sufficiently represents the filtered motion dynamics of the scene without negatively affecting the algorithm's speed. Experiments showed that the use of more agents heavily increased the computational cost, with a computational time of 3.86 sec per frame if the number of agents increased to 50, while the algorithm's performance actually decreased. This can be attributed to the fact that the presence of too many agents may lead to noisy internal forces due to the density of the swarm, which eventually degrades the results. On the other hand, the use of fewer agents, as few as 2 agents for example in "ped1", also decreased algorithm's performance from 78.87% to 73.66%. The initial agents' speed and accelerations are set to zero, whereas their initial positions are randomly generated.

B. Evaluation Criteria

In order to evaluate our method, we use the same criteria as the SoA literature for benchmark datasets. Thus, the frame and pixel level criteria described in [21] are adopted for UCSD dataset in Sec. VI-C, while the Area Under the Curve (AUC) is used for the UMN and U-turn videos described in Sec. VI-D and Sec.VI-E respectively.

The frame level criterion localizes changes only in time, predicting which frames contain an anomaly, without finding its spatial location: a frame is thus characterized as abnormal if it contains at least one abnormality, wherever it is located. In contrast, the pixel level criterion includes both temporal and spatial anomaly localization, and is used in the literature [21] as follows: if at least 40% of all anomalous pixels are found (as determined by the ground truth annotation), the detection is considered successful and the frame is characterized as abnormal. True positives and false positives are then derived by comparing the spatiotemporally detected anomalies with the ground truth, leading to Receiver Operating Characteristic (ROC) curves of true positives vs. false positives to evaluate the method's performance. It should be emphasized that the pixel level criterion is a more detailed, precise and reliable evaluation measure, since it localizes anomalies in both space and time. On the other hand, the frame level criterion's results, based on the correct detection of abnormal frames, may sometimes be coincidental, resulting from false positives appearing in frames that include true anomalies, without these anomalies having actually been detected.

The evaluation metrics used are derived from the ROC curves: the Equal Error Rate (EER) corresponds to the frame level criterion, while the Detection Rate (DR) corresponds to the pixel level criterion. These metrics have been widely used in the literature for the benchmark UCSD dataset, as they provide a reliable criterion to evaluate method's performance and to compare it with other SoA works. The EER corresponds to the error rate of a system when the false positives (detections of anomalies in a normal situation) are equal to the false negatives (missed anomaly detections). This is achieved by adjusting the threshold for accepting/rejecting a change until equal errors are achieved. The lower the EER, the higher the accuracy of the system. The DR, on the other hand, refers to the successful detection rate of the anomalies happening at EER, with higher detection rates implying a better performance of our algorithm.

TABLE I
"PED1" DATASET.

Equal Error Rate (EER) and Detection Rate (DR) in "ped1"

	SF[10]	MPPCA[16]	Adam[15]	Sparse[13]	PSO[1]	BVP[25]	Roshtkhari [26]	150fps [19]	H-MDT (CRF)[22]	Ours
EER	36.5%	35.6%	38.9%	19%	21%	18%	15%	15%	17.8%	27.02%
DR	40.9%	23.2%	32.6%	46%	52%	68%	71%	59.1%	74.5%	**78.87%**

TABLE II
"PED2" DATASET.

Equal Error Rate (EER) and Detection Rate (DR) in "ped2"

Method	EER	DR
SF [10]	35%	27.6%
MPPCA [16]	35.8%	22.4%
Adam [15]	45.8%	22.4%
H-MDT(CRF) [22]	18.5%	70.1%
Ours	26.92%	**74.92%**

The AUC criterion is simply the area under the ROC curve, derived for the UMN and U-turn datasets, as this criterion is also used in the literature for those videos.

C. UCSD dataset

The UCSD dataset is comprised of two subsets "ped1" and "ped2", containing different scenes recorded from different camera angles [21]. Each "ped1", "ped2" subset is divided into a training set containing exclusively normal frames and a test set, including different kinds of anomalies. The dataset consists of crowds of medium density traversing the scene ("ped2") or moving towards and away from camera, adding some perspective ("ped1"). The UCSD dataset constitutes a challenging dataset, as it contains many occlusions, a variety of anomalies, sometimes co-occurring in the same frame, and its resolution is of low quality. Anomalies present in the test set include bicycles, skaters or other wheeled objects moving with different speeds and passing through the crowd, which are in some cases difficult to detect even for human observers. "Ped1" comprises of 34 normal training clips and 36 test clips of size of 158×238 pixels, while "ped2" consists of 16 training clips and 14 test clips of 240×360 pixels.

Table I depicts the evaluation of the proposed algorithm for the "ped1" dataset, presenting the Equal Error Rate (EER) and the detection rate (DR) for the frame and pixel level criterion respectively. The full annotation provided by [25] was used for pixel level criterion. The method is compared against 9 other SoA works, which use different approaches to detect spatiotemporal anomalies in this dataset. These include descriptors based on social force flow dynamics [10], the mixture of optical flow observations (MPPCA) [16], the use of local low level motion histograms [15], sparse reconstruction approaches in motion histograms [13], [19], particle swarm optimization (PSO) to optimize a fitness function based on SFM [1], a Bayesian video parsing approach using a set of hypotheses that jointly explains the foreground [25], a bag of video words approach [26] and finally the hierarchical mixture of dynamic textures (H-MDT) after applying CRF filtering [22], as this variation of their method led to SoA results.

As observed from Table I our method greatly outperforms all other existing methods for the pixel level criterion, while for the frame level criterion it gives comparable results. However, as stated above, the frame level criterion is a less detailed and reliable descriptor of the performance of the algorithm than the pixel level criterion, as in some cases even perfect frame level anomaly detection can be achieved "coincidentally" by only detecting false positives. Therefore, we consider that overall our method significantly improves upon the SoA.

The results for "ped2" are presented in Table II. In this case, our method is compared for both evaluation criteria with 4 SoA approaches, as in [22]. Our method once again leads to better performance for the more precise pixel level criterion, while comparable results are obtained for the frame level evaluation. In Fig.13 the ROC curves for the complete UCSD dataset are presented.

Fig.11 and Fig.12 depict screenshots from successful detections in the "ped1" and "ped2" datasets respectively. As can be seen, different kinds of "anomalies" are successfully localized even when they co-occur in the same frame, as in Fig.11(f). A remarkable achievement of the proposed method is that deviations from normal patterns can be also detected in highly occluded scenes, as Fig.12((e)-(f)) illustrate. In these cases, a bicycle and a skater respectively are correctly identified as "anomalies", even though their detection is a challenging task even for a human observer. Videos with the outcomes of our algorithm on the UCSD dataset can be found in the following link: http://mklab.iti.gr/people/vagiakal.

We conducted the same experiments without including the swarms and using instead only the OF, while the rest of our algorithm remained the same to evaluate the effect of swarm intelligence. Table III shows the results for both cases in its first two rows, demonstrating that swarms effectively "filter" the OF values, leading to more accurate results and reinforcing our decision to incorporate them in our algorithm. For both "ped1" and "ped2", a higher detection rate is achieved when using swarms, while the EER is lower. Additionally, in order to examine the impact of the appearance and motion descriptors separately on performance, results are presented in the same table for when only the motion descriptor (HOS) or only the appearance descriptor (HOG) is used. For the "ped1" dataset, the use of both descriptors led to better results than using them separately. Nevertheless, for "ped2" the inclusion of the HOG descriptor worsened the algorithm's performance. However, as Table III shows, the results when using both descriptors or only motion descriptor are still comparable, proving that our decision of combining both descriptors is justified as it can be generalised in more cases, giving better or sometimes

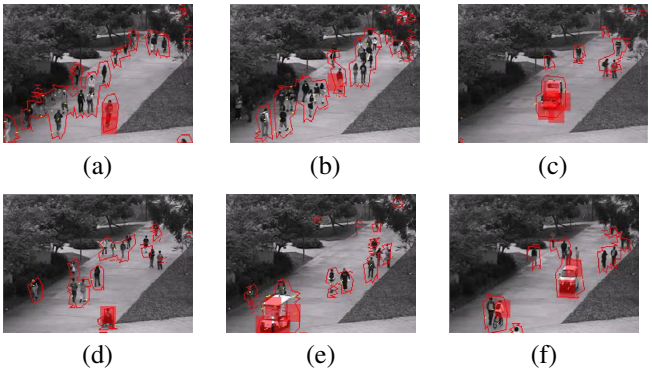


Fig. 11. Different kind of anomalies (shown in red blocks) were detected in Ped1 dataset: skaters ((a)-(b)),vehicle ((c),(e),(f)), wheeled chair (d), bicycle (f). Red contours show the foreground where the algorithm is applied while yellow points are the interest points in them.

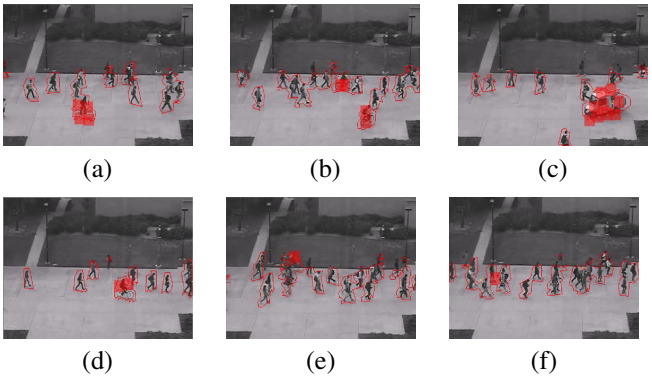


Fig. 12. Different kind of anomalies (shown in red blocks) were detected in Ped2 dataset: bicycles ((a),(b),(d),(e)), vehicle (c) and skater (f). Anomalies were successfully detected even when co-occurring in the same frame (b) or even in high occluded scenes ((e)-(f)). Red contours show the foreground where the algorithm is applied while yellow points are the interest points in them.

comparable results.

In Table IV, the computational cost of the method proposed for the UCSD dataset is presented, in comparison with 5 other SoA methods. Our method ranks on the average second in speed performance after the sparse approach of [19], which achieved a remarkably low computational cost, but at the expense of algorithm's detection performance. As it can be seen, their method achieves only a detection rate of 59.1% against ours of 78.87% and 74.92% for "ped1" and "ped2" dataset respectively. In comparison with the H-MDT (CRF) method, our approach exhibits lower cost with better detection performance for "ped1" dataset and comparable cost for "ped2", while also achieving better detection performance. Overall, Table IV shows that our algorithm achieves the best detection performance results at a low computational cost. All experiments were conducted on a 16GB RAM computer with a 3.5 GHz CPU. The algorithm runs in C++, without being optimized, meaning that the computational cost can be reduced even further, making it applicable to real world situations.

D. UMN dataset

The UMN dataset [35] consists of 7739 frames of 320×240 pixels in 3 different scenes (umn_1, umn_2, umn_3) including respectively 2, 6 and 3 scenarios of crowd escape events. The

TABLE III
PERFORMANCE OF VARIOUS VARIATIONS.

	EER		DR	
	ped1	ped2	ped1	ped2
with swarm	**27.02%**	26.92%	**78.87%**	74.92%
without swarm	28.52%	27.09%	68.74%	64.94%
only HOS	29.03%	**26.20%**	75.76%	**76.89%**
only HOG	40.11%	47.32%	69.49%	48.56%

TABLE IV
COMPUTATIONAL COST.

method	sec. per frame	Performance(DR)
Sparse[13]	3.8	46%
MDT[21]	25	45%
BVP[25]	5 - 10	68%
150fps[19]	**0.00697**	59.1%
H-MDT(CRF)[22]	1.11 (ped1)	74.5%
	1.38 (ped2)	70.1%
Ours	0.91 (ped1)	**78.87%**
	1.49 (ped2)	**74.92%**

first frames in each event depict a normal crowd situation, with people walking or standing in the scene, while "anomaly" takes place with a sudden evacuation. This data is quite straightforward, as the "anomaly" is global and can be easily detected even by only using the average frame motion. As a result, many methods have been proposed for this data, achieving near perfect scores.

The main drawback of this dataset is its limited size, in combination with the absence of a separate training set. The limited number of training frames results in a not well defined "normal" class. Our descriptor uses detailed appearance and motion information, however these change significantly, even in the "normal" frames, so it requires more training data for a better defined description of the "normal" events. As a result of its limited size, this data does not allow us to demonstrate the true potential of our method, which uses many complex features so as to be applicable to more difficult videos.

For training, normal frames of one scenario from scene 1 and two scenarios from scenes 2 and 3 were used to model

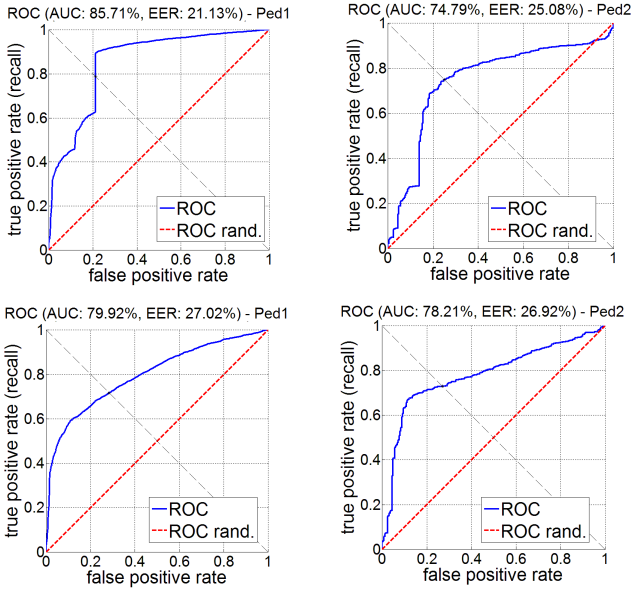


Fig. 13. ROC curves for the full UCSD dataset - top row: pixel level criterion, bottom row: frame level criterion.

normal crowd behaviors, while the rest of the frames were used for testing. Fig.14 has screenshots of our algorithm's outcome for all 3 scenes, while in Table V comparisons with 5 SoA methods [22] are provided. Anomalies are correctly detected and localized in all cases, however, some false positives appear due to the above mentioned lack of adequate training. The total performance of our algorithm reached 99.59% for umn_1, 93.38% for umn_2 and 98.08% for umn_3 using the Area Under Curve (AUC) criterion, which is used in the literature for UMN. Its performance is therefore shown to be near perfect, comparable with the SoA, with the exception of umn_2, where it achieved very high, but not perfect, results. This is attributed to the sparseness of the training data in umn_2, which were even less informative than those of umn_1 and umn_3.

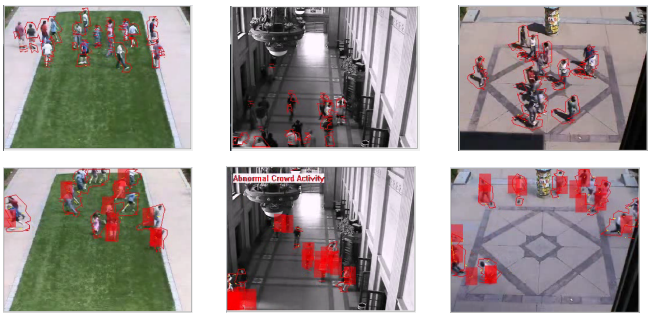


Fig. 14. Screenshots of our results for UMN. The first row depicts normal situations in all 3 scenes and the second row shows abnormal events (evacuation). Red areas demonstrate anomaly localization.

TABLE V
AUC PERFORMANCE FOR ANOMALY DETECTION IN THE UMN DATA

	SF [10]	chaotic [8]	sparse [13]	local stat. [14]	H-MDT (CRF) [22]	Ours umn_1	Ours umn_2	Ours umn_3
AUC (%)	94.9	99.4	**99.6**	99.5	99.5	99.59	93.38	98.08

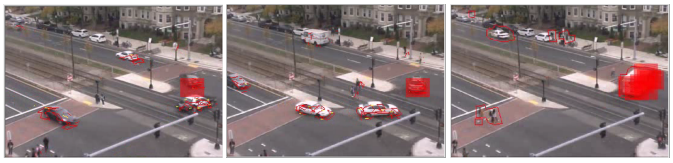


Fig. 15. Anomalies detected in red for the U-turn dataset by our algorithm.

E. U-turn

In order to confirm our method's robustness, we also applied it to a non-crowd dataset. We used the U-turn dataset of [36], which shows normal traffic in a crossroad and some cars making illegal U-turns ("anomaly"). The dataset comprises of 6117 frames of 360×240 pixels. The scenes are quite sparse and, in combination with the dataset's limited size, there is not much training data. However, even with limited training samples, all anomalies are perfectly detected and localized, as can be seen in Fig. 15. It is remarkable that in the first frame in Fig. 15, our algorithm correctly distinguishes between an illegal turn and a legal one. Around 3400 frames depicting normal traffic were used for training, and the rest were used for testing. In Fig. 16, the ROC curve of our method for the U-turn data is compared with the results provided by [22]. As it is shown, we achieve the highest AUC at the frame level, equal to $95,31\%$, with all "anomalies" having been correctly detected and localized.

F. Love Parade

The algorithm was also tested on the surveillance data of Love Parade 2010 [37], which contains videos of high density crowds. Snapshots are provided in Fig. 17 and, as it can be observed, deviations from normal crowd patterns are correctly detected and localized, despite the little motion present, and the high number of occlusions, due to the high crowd density. Around 1000 frames were used for training with the rest of the frames used for testing. The truck and ambulance are successfully detected while traversing a highly dense crowd, whereas people in the crowd jumping over railings are also detected as an anomalous behavior.

VII. CONCLUSION

In this work, we propose a novel framework for anomaly detection in different scenarios, recorded from static surveillance cameras. Swarm intelligence is exploited for the extraction of robust motion characteristics and together, with appearance features, form a descriptor capable of effectively describing each scene. Its remarkable performance in 4 completely different kinds of datasets proves the method's generality and its applicability in real life situations. The high detection rate in the UCSD dataset, that greatly outperforms various state-of-the-art approaches, especially on the most challenging pixel level criterion, demonstrates that the proposed algorithm can be effectively used for challenging crowd videos with many occlusions, local noise and local scale variations. This fact in combination with its low computational cost and its effectiveness in different environments, make our algorithm very appropriate for a variety of surveillance applications.

ACKNOWLEDGMENT

This work was funded by the European Commission under the 7th Framework Program (FP7 2007-2013), grant agree-

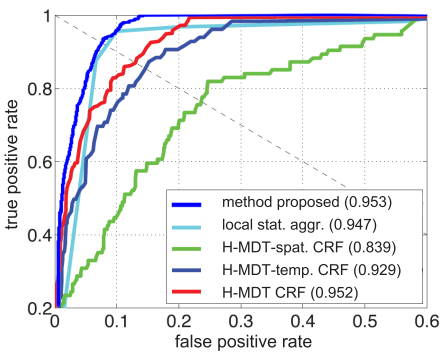


Fig. 16. ROC curve for frame level criterion in U-turn dataset

Fig. 17. Our results for Love Parade. The truck and ambulance are successfully detected going through a very dense crowd. People from the crowd climbing over the railings are also detected as anomalous behaviour.

ment 288199 Dem@Care, A-0120-RT-GC MEDUSA - Multi SEnsor Data Fusion Grid for Urban Situational Awareness.

REFERENCES

[1] R. Raghavendra, A. Del Bue, M. Cristani, and V. Murino, "Optimizing interaction force for global anomaly detection in crowded scenes," in *International Conference on Computer Vision Workshops (ICCVW), IEEE International Conference on*, 2011, pp. 136–143.

[2] N. Cuntoor, B. Yegnanarayana, and R. Chellappa, "Activity modeling using event probability sequences," *Image Processing (TIP), IEEE Transactions on*, vol. 17, no. 4, pp. 594–607, 2008.

[3] L. Wang, Y. Qiao, and X. Tang, "Latent hierarchical model of temporal structure for complex activity classification," *Image Processing (TIP), IEEE Transactions on*, vol. 23, no. 2, pp. 810–822, 2014.

[4] W. Hu, X. Zhou, W. Li, W. Luo, X. Zhang, and S. Maybank, "Active contour-based visual tracking by integrating colors, shapes, and motions," *Image Processing (TIP), IEEE Transactions on*, vol. 22, no. 5, pp. 1778–1792, 2013.

[5] A. Basharat, A. Gritai, and M. Shah, "Learning object motion patterns for anomaly detection and improved object detection," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2008, pp. 1–8.

[6] C. Piciarelli, C. Micheloni, and G. Foresti, "Trajectory-based anomalous event detection," *Circuits and Systems for Video Technology (CSVT), IEEE Transactions on*, vol. 18, no. 11, pp. 1544–1554, 2008.

[7] D. Tran, J. Yuan, and D. Forsyth, "Video event detection: From subvolume localization to spatiotemporal path search," *Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on*, vol. 36, no. 2, pp. 404–416, 2014.

[8] S. Wu, B. E. Moore, and M. Shah, "Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2010, pp. 2054–2060.

[9] J. Nascimento, M. Figueiredo, and J. Marques, "Activity recognition using a mixture of vector fields," *Image Processing (TIP), IEEE Transactions on*, vol. 22, no. 5, pp. 1712–1725, 2013.

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

[11] X. Cui, Q. Liu, M. Gao, and D. Metaxas, "Abnormal detection using interaction energy potentials," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2011, pp. 3161–3167.

[12] H. Ullah and N. Conci, "Crowd motion segmentation and anomaly detection via multi-label optimization," in *International Conference on Pattern Recognition Workshop (ICPRW), IEEE International Conference on*, 2012.

[13] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2011, pp. 3449–3456.

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

[15] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixed-location monitors," *Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on*, vol. 30, no. 3, pp. 555–560, 2008.

[16] J. Kim and K. Grauman, "Observe locally, infer globally: A space-time mrf for detecting abnormal activities with incremental updates," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2009, pp. 2921–2928.

[17] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, "Textures of optical flow for real-time anomaly detection in crowds," in *Advanced Video and Signal-Based Surveillance (AVSS), 8th IEEE International Conference on*, 2011, pp. 230–235.

[18] L. Kratz and K. Nishino, "Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2009, pp. 1446–1453.

[19] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 fps in matlab," in *International Conference on Computer Vision (ICCV), IEEE International Conference on*, 2013, pp. 2720–2727.

[20] V. Kaltsa, A. Briassouli, I. Kompatsiaris, and M. Strintzis, "Timely, robust crowd event characterization," in *International Conference on Image Processing (ICIP), IEEE International Conference on*, 2012, pp. 2697–2700.

[21] V. Mahadevan, W. Li, V. Bhalodia, and N. Vasconcelos, "Anomaly detection in crowded scenes," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2010, pp. 1975–1981.

[22] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on*, vol. 36, no. 1, pp. 18–32, 2014.

[23] Y. Ito, K. Kitani, J. Bagnell, and M. Hebert, "Detecting interesting events using unsupervised density ratio estimation," in *European Conference on Computer Vision Workshop (ECCVW), IEEE Conference on*, vol. 7585, 2012, pp. 151–161.

[24] V. Reddy, C. Sanderson, and B. Lovell, "Improved anomaly detection in crowded scenes via cell-based analysis of foreground speed, size and texture," in *Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Conference on*, 2011, pp. 55–61.

[25] B. Antic and B. Ommer, "Video parsing for abnormality detection," in *Interantional Conference on Computer Vision (ICCV), IEEE International Conference on*, 2011, pp. 2415–2422.

[26] M. Roshtkhari and M. Levine, "Online dominant and anomalous behavior detection in videos," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2013, pp. 2611–2618.

[27] G. Apostolidis and L. J. Hadjileontiadis, "Swarm decomposition: Novel nonstationary signal analysis using swarm intelligence," *IEEE Signal Processing*, 2014.

[28] A. Sobral, "BGSLibrary: An opencv c++ background subtraction library," in *IX Workshop de Viso Computacional (WVC)*, 2013.

[29] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition (CVPR), IEEE Computer Society Conference on*, vol. 1, 2005, pp. 886–893 vol. 1.

[30] K. Avgerinakis, A. Briassouli, and I. Kompatsiaris, "Recognition of activities of daily living for smart home environments," in *Intelligent Environments (IE), International Conference on*, 2013, pp. 173–180.

[31] H. M. H. Teodorescu and D. J. Malan, "Swarm Filtering Procedure and Application to MRI Mammography." scielomx, 2010, pp. 59 – 64.

[32] V. Kaltsa, A. Briassouli, I. Kompatsiaris, and M. Strintzis, "Swarm-based motion features for anomaly detection in crowds," in *International Conference on Image Processing (ICIP), IEEE International Conference on*, 2014.

[33] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, pp. 4282–4286, 1995.

[34] D. Tax and R. Duin, "Support vector data description," *Machine Learning*, vol. 54, no. 1, pp. 45–66, 2004.

[35] "Unusual crowd activity dataset made available by the university of minnesota at: http://mha.cs.umn.edu/movies/crowdactivity-all.avi/."

[36] Y. Benezeth, P.-M. Jodoin, V. Saligrama, and C. Rosenberger, "Abnormal events detection based on spatio-temporal co-occurences," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, 2009, pp. 2458–2465.

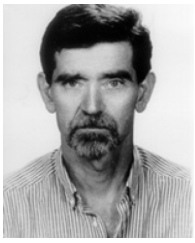
[37] B. Krausz and C. Bauckhage, "Loveparade 2010: Automatic video analysis of a crowd disaster," *Computer Vision and Image Understanding*, vol. 116, no. 3, pp. 307 – 319, 2012, special issue on Semantic Understanding of Human Behaviors in Image Sequences.

Vagia Kaltsa received the Diploma degree in Electronics and Computer Engineering from the Aristotle University of Thessaloniki in 2009. Since October 2010, she is a Ph.D. candidate at the Aristotle University of Thessaloniki and she is also working as a Postgraduate Research Fellow at the Information Technologies Institute (ITI) in Centre for Research and Technology Hellas (CERTH). Her current research interests include computer vision, video and image processing, analysis of crowds and anomaly detection from surveillance videos. She is a student member of IEEE.

Dr Alexia Briassouli received the Diploma degree in electronic engineering from the National Technical Univ. of Athens in 1999, the Masters degree in the interdisciplinary program for Systems of Signal Processing and Communications, Theory and Application at the Univ. of Patras in 2000, and the Ph.D. degree from the Department of Electrical and Computer Engineering at the University of Illinois in Urbana Champaign in 2006. From 2006-2010 she worked part-time as a visiting adjunct professor at the Dept. of Electrical and Computer Engineering in the University of Thessaly, where she taught digital image processing, video processing, signals and systems, digital signal processing, among others. Since 2006 she has been working as a postdoctoral research fellow at CERTH, conducting research, co-supervising PhD theses and participating in National and European research projects. Her current research includes the analysis of video for detection of unusual or interesting events, and activity recognition using statistical methods, signal processing and computer vision techniques. She has authored over 14 journal publications, 35 conference publications, 2 book chapters, and is one of the editors of the Springer book Comprehensive Health Monitoring and Personalized Feedback Using Multimedia Data. She has organized special sessions and workshops in conferences, such as the 1st ACM MM Workshop on Multimedia Indexing and Information Retrieval for Healthcare and has organized and supported various dissemination activities, including the ICT Exhibition in Vilnius, Lithuania in 2013, the DemAAL Dem@Care Summer School on Ambient Assisted Living, round table discussions etc. She is a member of IEEE.

Dr. Ioannis (Yiannis) Kompatsiaris is a Senior Researcher (Researcher A) with the Information Technologies Institute / Centre for Research and Technology Hellas, Thessaloniki, Greece. His research interests include semantic multimedia analysis, indexing and retrieval, social media and big data analysis, knowledge structures, reasoning and personalization for multimedia applications, eHealth and environmental applications. He received his Ph.D. degree in 3-D model based image sequence coding from the Aristotle University of Thessaloniki in 2001. He is the co-author of 76 papers in refereed journals, 35 book chapters, 8 patents and more than 250 papers in international conferences. He has been the co-organizer of various international conferences and workshops and has served as a regular reviewer for a number of journals and conferences. He is a Senior Member of IEEE and member of ACM.

Dr. Leontios J. Hadjileontiadis (S87M98-SM11) was born in Kastoria, Greece in -1966. He received the Diploma degree in Electrical Engineering in 1989 and the Ph.D. degree in Electrical and Computer Engineering in 1997, both from the Aristotle University of Thessaloniki (AUTH), Thessaloniki, Greece. Since December 1999 he joined the Department of Electrical and Computer Engineering, AUTH, Greece as a faculty member, where he is currently Full Professor, working on lung sounds, heart sounds, bowel sounds, ECG data compression, EEG-based affective computing, seismic data analysis, educational data modeling and crack detection in the Signal Processing and Biomedical Technology Unit of the Telecommunications Laboratory. His research interests are in higher-order statistics, alpha-stable distributions, higher-order zero crossings, swarm modeling, wavelets, polyspectra, fractals, dense EEG-based 3D-vector field tomography, neuro-fuzzy modeling for medical, mobile and digital signal processing applications. Prof. Hadjileontiadis is a member of the Technical Chamber of Greece, of the IEEE, of the Higher-Order Statistics Society, of the International Lung Sounds Association, and of the American College of Chest Physicians. He was the recipient of the second award at the Best Paper Competition of the ninth Panhellenic Medical Conference on Thorax Diseases97, Thessaloniki. He was also an open finalist at the Student paper Competition (Whitaker Foundation) of the IEEE EMBS97, Chicago, IL, a finalist at the Student Paper Competition (in memory of Dick Poortvliet) of the MEDICON98, Lemesos, Cyprus, and the recipient of the Young Scientist Award of the twenty-fourth International Lung Sounds Conference99, Marburg, Germany. He organized and served as a mentor to four-student teams awarded at the Imagine Cup Competition (Microsoft), Sao Paulo, Brazil (2004)/Yokohama, Japan (2005)/Seoul, Korea (2007)/New York, USA/ (2011), Sydney, Australia (2012), with projects involving technology-based solutions for people with disabilities and pain management. In this framework, he was awarded with the Champions Faculty Award 2012 in Sydney, Australia. Since 2012, he serves as an IEEE Student Branch and IEEE EMBS Branch Chancellor and in August 2013 he and his student team Symbiosis were awarded the IEEE Student Enterprise Award 2013. Prof. Hadjileontiadis also holds a Diploma in Musicology (AUTH, Thessaloniki, 2011), a Ph.D. degree in music composition (University of York, UK, 2004), and he is currently a Professor in composition at the State Conservatory of Thessaloniki, Greece.



Dr. Michael G. Strintzis (M70SM80F04) received the Diploma in electrical engineering from National Technical University of Athens, Athens, Greece, in 1967, and the M.A. and Ph.D. degrees in electrical engineering from Princeton University, Princeton, NJ, in 1969 and 1970, respectively. He then joined the Electrical Engineering Department at University of Pittsburgh, Pittsburgh, PA, where he served as an Assistant Professor during 19701976, and Associate Professor during 1976 1980. Since 1980, he has been a Professor of electrical and computer engineering at Aristotle University of Thessaloniki, Thessaloniki, Greece. He is the founder of the Informatics and Telematics Research Institute, Thessaloniki, Greece, where he served as Director from 1999 to 2009. His current research interests include 2-D and 3-D image coding, image processing, biomedical signal and image processing, and DVD and Internet data authentication and copy protection. Prof. Strintzis has served as an Associate Editor for the IEEE Transactions on Circuits and Systems for Video Technology since 1999. In 1984, he was awarded one of the centennial medals from IEEE.