Signal Processing 89 (2009) 1723–1738

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Detecting abnormal human behaviour using multiple cameras

Panagiota Antonakaki *, Dimitrios Kosmopoulos, Stavros J. Perantonis

Computational Intelligence Laboratory, Institute of Informatics and Telecommunications, National Center for Scientific Research "Demokritos", Athens, Greece

ARTICLE INFO

Article history:
Received 6 August 2008
Received in revised form
9 March 2009
Accepted 11 March 2009
Available online 5 April 2009

Keywords:
Behaviour understanding
Trajectory
Hidden Markov Model
Support vector machine
Homography

ABSTRACT

In this paper a bottom-up approach for human behaviour understanding is presented, using a multi-camera system. The proposed methodology, given a training set of normal data only, classifies behaviour as normal or abnormal, using two different criteria of human behaviour abnormality (short-term behaviour and trajectory of a person). Within this system an one-class support vector machine decides short-term behaviour abnormality, while we propose a methodology that lets a continuous Hidden Markov Model function as an one-class classifier for trajectories. Furthermore, an approximation algorithm, referring to the Forward Backward procedure of the continuous Hidden Markov Model, is proposed to overcome numerical stability problems in the calculation of probability of emission for very long observations. It is also shown that multiple cameras through homography estimation provide more precise position of the person, leading to more robust system performance. Experiments in an indoor environment without uniform background demonstrate the good performance of the system.

1. Introduction

Motion analysis in video and particularly human behaviour understanding has attracted many researchers [24], mainly because of its fundamental applications, which include video indexing, virtual reality, human–-computer interaction and smart surveillance. Smart surveillance in itself is one of the most challenging problems in computer vision. Its goal is to automatically model and identify human behaviours, calling for human attention only when a suspicious behaviour is detected. With the increasing number of cameras in many public areas, the related research becomes more appealing and is offered more application possibilities.

This work deals with the classification of behaviours as normal or abnormal. Based on the remark that abnormal behaviour is considered to be rather infrequent (and thus abnormal), we choose to model normal behaviour and define as abnormal any behaviour deviating from that normality model. Our methodology applies two classification criteria:

(1) short-term behaviour;
(2) trajectory.

The short-term behaviour refers to the type of behaviour that can be localized in a spatio-temporal sense, i.e. is brief and within restricted space. Examples of such behaviours are walking, standing still, running, moving abruptly, etc.

In the related literature the aforementioned classification criteria are mostly treated separately and, furthermore, few works concentrate on learning only normal behaviours. The methodology provided herein provides the discrimination of anomaly due to abnormal short-term motion, as happens in the case of abrupt motion, as well as anomaly due to long-term motion, as in the case of abnormal trajectory.

* Corresponding author.
E-mail addresses: ganton@iit.demokritos.gr (P. Antonakaki), dkosmo@iit.demokritos.gr (D. Kosmopoulos), sper@iit.demokritos.gr (S.J. Perantonis).

0165-1684/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2009.03.016

Recently, several researchers have dealt with the problem of anomaly detection, which is the process of behaviour classification as normal or abnormal. A variety of methods, ranging from fully supervised [9,10] to semi-supervised [36] and unsupervised systems [21,22,18], have been proposed in existing literature, which we further review in Section 2. It should be noted, however, that most of the existing approaches do not use multi-camera information, except for [38], where multiple video streams are combined via a coupled Hidden Markov Model.

Our methodology contributes in current research in several ways:

- The presented approach reflects two different criteria of labelling an observed behaviour as normal or abnormal, since the final abnormality decision depends on the output of two different classifiers with independent inputs: short-term behaviour information and trajectory information.
- The behaviours are classified according to the target object's position on the *ground plane*, based on homography (see Section 4) which provides higher accuracy compared to pure image-based techniques.[1]
- We introduce a continuous Hidden Markov Model (cHMM) as an one-class classifier, using the notion of length-normalized log-probability (see Section 6.1).
- A novel algorithm implementing a Forward Backward procedure for the emission probability estimation in HMMs is proposed, handling numerical instability resulting from long sequences (see Section 6.2).

The rest of the paper is organized as follows. In Section 2 recent literature is reviewed, hinting as to the problems the proposed method tackles. Section 3 provides an overview of the proposed architecture. In Section 4 we explain briefly how homography is used to obtain information on the position of target objects on the ground plane. In Section 5 short-term behaviours are defined in terms of a set of extracted features. Section 5.2 describes in detail the classification process which is based on short-term behaviours. In Section 6, on the other hand, trajectories' classification is presented by elaborating how we have used a continuous Hidden Markov Model as an one-class classifier (Section 6.1). As an added value, Section 6.2 contains the description and foundation of a modified algorithm for the Forward Backward procedure of probability estimation tackling long sequences in contemporary computers. Finally, in Section 7 we provide the experimental results and Section 8 concludes this paper through a brief discussion on the lessons learned.

2. Related work

A typical surveillance system is divided into two layers, which include *low level* and *high level* processes, respectively, as depicted in Fig. 1.

[1] An early version of this work has been presented in [20].

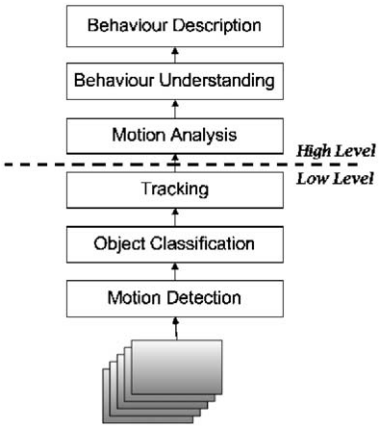

Fig. 1. The main framework for video surveillance systems.

The *low level* contains such methods as motion detection, object classification and tracking. In motion detection research is focused on either static or adaptive background subtraction or temporal differencing algorithms, aiming to isolate the foreground pixels that participate into any kind of motion observed in a given scene. Object classification is the process of classifying detected objects into such classes as humans or vehicles, appearing in a given scene. Following motion detection and object classification, detected objects are located in the course of time and their trajectories are extracted via tracking.

High level processes use motion information from the low level in order to finally identify the type or nature of a moving object's activity. Motion-based techniques are mostly used for short-term activity classification (e.g. walking, running, fighting), and do not take into account object trajectories. These techniques actually calculate features of the motion itself and perform recognition of behaviours based on these features' values. Such methods have been presented by Bobick et al. in [5] where motion energy images (MEIs) and motion history images (MHIs) are used to classify aerobic type exercises. Taking this work another step further, Weinland et al. in [34] focus on the extraction of motion descriptors analogous to MHIs, called *motion history volumes*, from multiple cameras. Then, these history volumes are classified into primitive actions. Efros et al. in [11] compute the optical flow [14] of a given object to recognize short-term behaviours through a nearest-neighbour classification.

Several methods that take into account the object's trajectory for behaviour classification use the centroid of the target object [1,19,27,15] or points of interest in a given image [4]. These methods, however, fail to take into account the short-term actions, for example the case where a man threateningly moves his hands. Most of the existing methods also face problems like view dependency, and occlusion when they extract trajectories from one camera.

HMMs and their variations have been widely applied on trajectory classification, e.g. [7,17,2,32], due to their unsupervised training, their simplicity and computational

efficiency and mainly because motion can be viewed as a short-term stationary signal. Abstract Hidden Markov Models are used by Nguyen et al. in [26] to deal with noise and duration variation, while Wang et al. in [33] use conditional random fields for behaviour recognition in order to be able to model context dependence in behaviours. In our approach we use a continuous HMM to model trajectory, using a methodology that allows the model to be used as an one-class classifier.

Our presented approach focuses on the anomaly detection aspect of behaviour understanding, which differentiates it from the aforementioned methods. However, recent research has provided several anomaly-detection-focused approaches that we briefly review here. These approaches can be classified based on whether they are supervised, semi-supervised or unsupervised.

In [9,10] the authors use supervised approaches that need the classes of both normal and abnormal behaviour to have an adequately large number of labelled instances, provided as a priori information. In our method, on the other hand, the training set only consists of normal instances of data. The semi-supervised method of [36], which only uses normal data, has a different approach in that it creates a set of marginally normal instances as abnormal to constitute an estimation of the abnormal class. In our work, we have used the derived feature of length-normalized log-probability to define the normal class, without attempting to generate abnormal instances at all. On the other hand, we also take into account motion-based features used in an one-class SVM to detect further abnormalities.

A set of unsupervised methods in existing literature use large databases [37,6] containing all the observed normal behaviour patterns, matching any new instances against the database represented instances. In our work, we have a single composite model (including HMM and SVM classification) for all normal instances, thus avoiding the need for database storage and look-up. Jiang et al. in [18] start by representing normal trajectories by a single HMM model per trajectory, clustering and retraining these HMMs until a given condition holds. Other than the fact that, in the work presented herein, we also cover the case of short-term behaviours besides trajectory, we model the full set of normal trajectories into a single HMM from the beginning. Therefore, less calculations are required. Lee et al. in [21] use n-cut clustering over motion energy images to determine outliers, which are then judged as abnormal. This approach is different from ours in that it requires repetition of the n-cut clustering when a new instance is to be judged. Another approach is found in [22], where a multi-layer finite state machine representation is used to model activities. According to [22], an abnormal activity is judged by the number of times a valid transition fails to be performed when matching the activity to the model state machine. Our approach uses probabilistic tools as the HMM instead of finite state machines to model uncertainty within the normal activities' modelling. In [35], a single feature vector represents position, motion and shape information, which is used in a clustering process to detect abnormality. In our approach we extract separate information for each classifier, attempting to model more precisely two aspects of motion. This kind of modularity allows switching between using one or both classifiers for the detection of either abnormal short-term behaviours, abnormal trajectory, or both. Furthermore, one can use information from each classifier to determine the type of abnormality detected.

In behaviour understanding, only few works employ homography estimation. Park et al. in [28] have used homography to extract object features and, using spatio-temporal relationships between people and vehicles, extract semantic information from interactions calculated from relative positions. Ribeiro et al. in [30] have estimated homography and enabled an orthographic view of the ground plane which eliminates perspective distortion origination from a single camera. Then, they have calculated features in order to classify the data in four activities (active, inactive, walking, running).

In existing literature two basic assumptions are usually made in order to extract features. The first is that the targets move almost vertically to the camera z-axis or within a range that is small compared to the distance from the camera. This assumption ensures that the size variation of moving objects is relatively small. The second assumption is that humans are planar objects, so that homography-based image rectification can be possible. However, even though this later assumption may be true when the cameras are close to being vertical to the ground plane, as in the case of cameras viewing from high ceilings, it does not stand in general. In our method we get over these limitations, as can be deduced from the section on homography estimation (Section 4).

3. Proposed methodology

The proposed methodology is based on the fusion of data that we collect from several cameras with overlapping fields of view. We perform classification using two different one-class classifiers, a support vector machine (SVM) and a continuous Hidden Markov Model, with each classifier having different feature vectors as input. The final decision on the behaviour is made by taking into account outputs from both classifiers.

The system architecture is presented in Fig. 2. The *low level* addresses the problem of motion detection and blob analysis, providing the upper level with two different features vectors per instance. We note that an object's blob is defined to be the set of the foreground pixels that belong to that object. Background subtraction is applied for motion detection and a bounding box is extracted. The blobs apparent within the viewing area of each camera are used to extract the objects' principal axes. These principal axes in combination with the corresponding homography calculations are used to locate each object, i.e. determine the points where the target object touches the ground plane. From the coordinates of the latter points we calculate the trajectories of the objects.

Additional object information, namely the object's centroid, blob size and shape are made available during the preprocessing step. Furthermore, a histogram is extracted from the moving object's shape depicting the

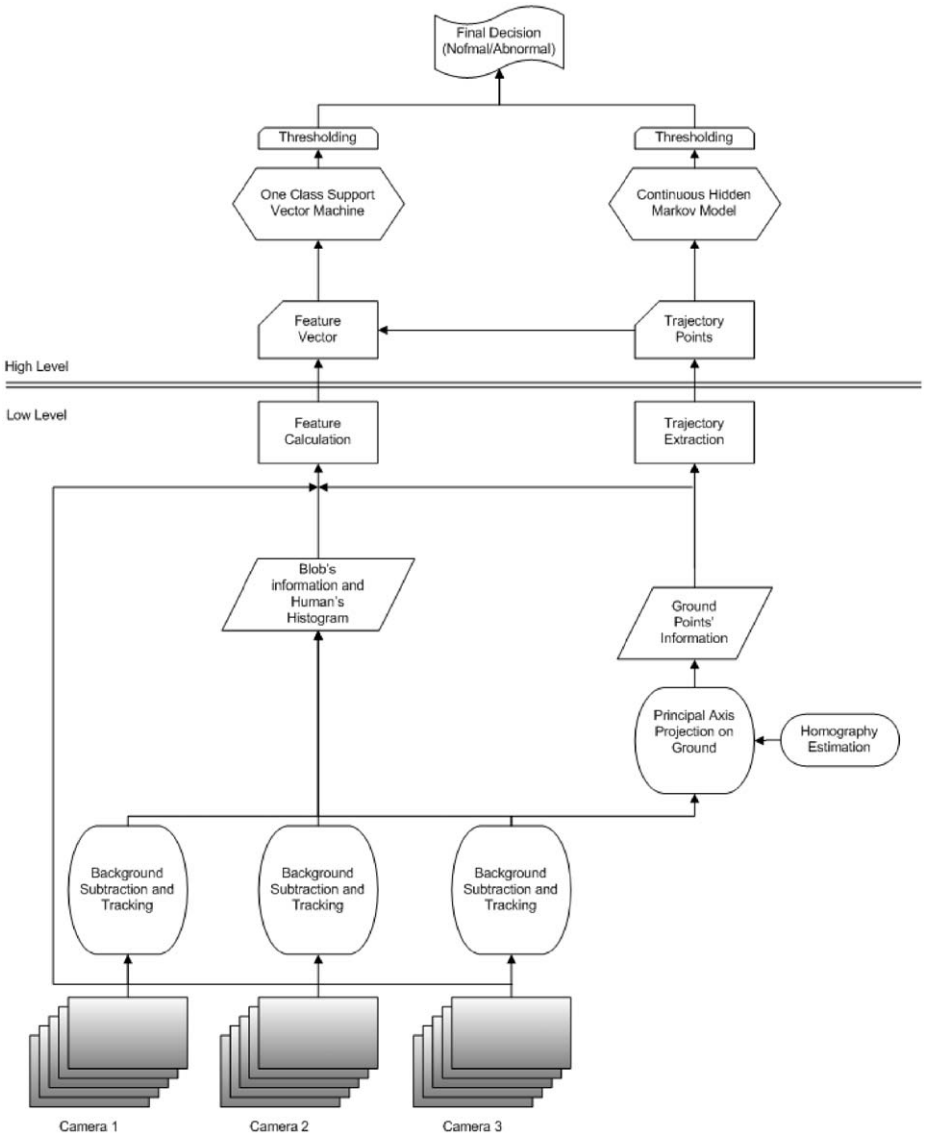

Fig. 2. System overview.

moving object's blob projection on the *y*-axis. The overall set of elementary features is used for the creation of the final two feature vectors per instance: one vector for each classifier.

The two classifiers used at this point are able to decide about the normality of the observed behaviour under two different views:

- The first classifier (one-class support vector machine (SVM)) decides if the short-term behaviour is normal or not, supplied with feature vectors computed by taking into account both the background subtraction and the ground plane information. The features provided as input describe the short-term motion information, which we argue that constitute the short-term behaviour information.
- The second classifier is a continuous Hidden Markov Model (cHMM), also used as one-class classifier, which supplied with the trajectory of every instance-object. This classifier can decide whether a given trajectory follows the model of normal trajectories.

Our method has been implemented to work in two modes: offline and real-time. In the offline mode, the decision concerns the classification of a time window of arbitrary length, which can be used for example for the characterization of video shots for video retrieval purposes. In its real-time aspect, the system makes a deci sion in every frame whether to issue alerts as the events happen. This decision is made by taking into consideration a time window of relatively small duration concerning recent camera information (images). This

aspect can be used for security purposes, aiding a human supervisor.

In the recognition step, if either classifiers gives "abnormal" characterization as an output, the system characterizes the scene as abnormal. This means that we take as output the logical "or" of outputs, given that a value of *true* indicates abnormality.

4. Preprocessing

The proposed methodology uses a preprocessing step that includes background subtraction for moving target segmentation and then target localization using homography information. For the background subtraction, we adopted the adaptive Gaussian mixture background model for dynamic background modelling [39]. Similar or better methods could have been used for the same purpose, without changing our overall approach, and the reader is referred to the related literature for further information.

For target localization we have employed a homography-based approach. The planar homographies are geometric entities whose role is to provide associations between points on different planes, which are the ground and the camera planes in our case. In our indoor environment the target moves on the ground plane, so mapping between planes is possible. In the following we explain briefly how the approach works.

The scene viewed by a camera comprises a predominant plane, the ground. We assume that a homogeneous coordinate system is attached to the ground plane, so that a point on the plane is expressed as $\mathbf{P}_\pi = (x_{\pi1}, x_{\pi2}, x_{\pi3})^T$. If this point is visible to the camera, which is a matter of proper camera configuration, the homogeneous coordinates of this point on the camera plane are given by $\mathbf{P}_c = (x_{c1}, x_{c2}, x_{c3})^T$. The homography $\mathbf{H}$ is a 3×3 matrix, which relates $\mathbf{P}_\pi$ and $\mathbf{P}_c$ as follows:

$$\mathbf{P}_\pi = \mathbf{H} \cdot \mathbf{P}_c \;\Leftrightarrow\; \begin{bmatrix} x_{\pi1} \\ x_{\pi2} \\ x_{\pi3} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \cdot \begin{bmatrix} x_{c1} \\ x_{c2} \\ x_{c3} \end{bmatrix} \tag{1}$$

Let the inhomogeneous coordinates of a pair of matching points $\mathbf{x}_c = (x_c, y_c)$ and $\mathbf{x}_\pi = (x_\pi, y_\pi)$ on the camera plane (pixel coordinates) and the ground plane correspondingly. Then

$$x_\pi = \frac{x_{\pi1}}{x_{\pi3}} = \frac{h_{11} \cdot x_c + h_{12} \cdot y_c + h_{13}}{h_{31} \cdot x_c + h_{32} \cdot y_c + h_{33}} \tag{2}$$

$$y_\pi = \frac{x_{\pi2}}{x_{\pi3}} = \frac{h_{21} \cdot x_c + h_{22} \cdot y_c + h_{23}}{h_{31} \cdot x_c + h_{32} \cdot y_c + h_{33}} \tag{3}$$

Each point correspondence gives an equation and four points are sufficient for the calculation of $\mathbf{H}$ up to a multiplicative factor, if no triplet of the used points contains collinear points. The calculation of $\mathbf{H}$ is a procedure done once offline and in practice many points are used to compensate for errors.

The positioning of each target is done similarly to [16]. A background subtraction algorithm extracts the silhouettes of the targets, which move on the ground plane. From each silhouette we extract the vertical principal axis

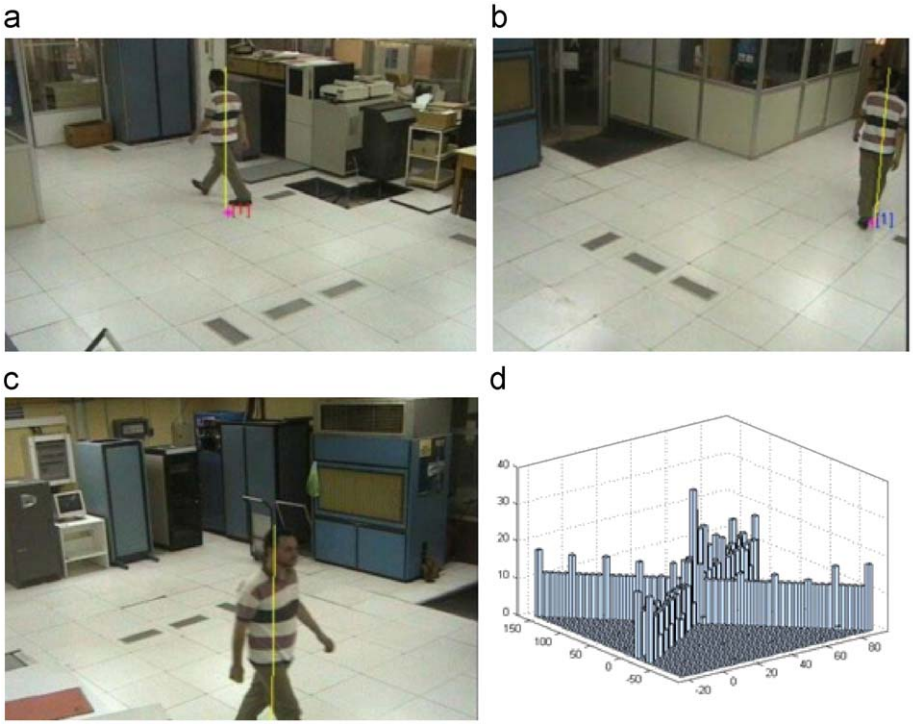

Fig. 3. View from three cameras and extraction of the principal axis projection on the ground plane from two of the cameras. In (c) the projection is not visible, however, the corresponding accumulator is still created in (d). In (d) three accumulators are visible—two of them very close to each other.

and we project it on the ground plane by replacing $(x_c, y_c, 1)^T$ and $(x_\pi, y_\pi, 1)^T$ in (1). The projection from each camera casts a "line" on the ground plane as depicted in Fig. 3. The maxima of those projected lines indicate the positions of the monitored targets, i.e. where the vertical principal axis touches the ground. The method is not strongly affected when the target pose is not vertical, because a vertical principal axis is still extracted from silhouettes. In such cases the indicated position is not the exact position of the feet touching the ground but the one indicated by the vertical axes, which may be a bit displaced. However, also in such cases the method still gives good position estimations.

5. Short-term behaviours

Our first source of information for evaluating behaviour is the so-called short-term behaviour. Our methodology represents short-term behaviour with a feature vector that consists of motion-based features. In the recognition step an one-class support vector machine is used, trained only with normal instances.

5.1. Feature calculation

In motion representation and analysis, our methodology uses information obtained by preprocessing, namely the object's bounding box, the object's blob and sequential positions. In Fig. 4, all preprocessing-extracted information are illustrated.

Elaborating, from the background subtraction process we extract the position of the object's centroid inside the bounding box, the bounding box's width and height and the object's blob. Figs. 4a–c show the captured frames from each camera with the corresponding bounding boxes. Figs. 4d–f show the background subtraction masks, from where the blob is extracted.

The blob histogram is calculated based on the blob information. The histogram of the blob indicates the number of pixels that belong to the blob for every y coordinate. Figs. 4g–i show the histograms of the given blob.

From homography estimation we calculate the object ground position and thus the trajectory which is expressed as a sequence of (x, y) vectors on the ground plane. Fig. 4j illustrates the object's trajectory in the scene, calculated from all views.

The short-term activity is represented by a seven-dimensional feature vector, as follows:

$$f = (v(t), \widehat{v_T}(t), R_T(t), F(t), \Delta F(t), \max(\Delta H(t)), \max(\Delta SD(t))) \quad (4)$$

The features' calculation is presented in detail in Table 1, with the features being separated into four categories according to what type of information they depend on. The first two features, speed and algebraic mean speed, are computationally inexpensive and time efficient calculated only from trajectory data. Algebraic mean blob difference is also time efficient calculated only from the background subtraction data on the object's bounding box. Mean optical flow and mean optical flow percentage difference are derived from simple operations on optical flow. For these two features we use data from both the object's bounding box as well as the full images of the video sequences. Optical flow is computationally expensive, but is robust and discriminative [14]. The last two features are computationally inexpensive, and they are extracted from the blob histogram. We have said that the histogram reflects the number of the pixels that consist the foreground object per y coordinate. But, if we weigh out the histogram with the total number of the histogram's pixels, we have a probability distribution function (pdf), $p_c(y_j)$, that represents the probability of an object's pixel to lie in a given coordinate in the bounding box, y_j.

Taking into account that features are extracted for every single video frame and constitute the frame's feature vector, we elaborate on the calculations presented in Table 1.

(1) $v(t)$, is the Euclidean norm (over x- and y-axes of the ground plane) of the instantaneous object's speed, calculated from the current frame and the previous frame object's position.
(2) Algebraic mean speed, $\widehat{v_T}(t)$, is the *algebraic* mean value of an object's speed within a time window that consists of the T last frames, including the frame on t_0. This value is calculated based on the *algebraic* sum of the x and y coordinates of the speed's vector, which is more robust against noise than $v(t)$.
(3) On the same grounds, the calculation of mean blob difference, $R(t)$, is based on the algebraic sum of the bounding boxes' area change within a shifting frame window T' comprising the last e.g. 5–10 frames. ($w_c(j)$, $h_c(j)$ represent the width and the height of the blob for camera c for $t = j$.
(4) Optical flow, F_i is first calculated on every frame and for each camera i, but only for the object's edges inside the bounding box. Then, the optical flow value is normalized by the number of the pixels that participate in the calculation—which are the pixels of the edges—and the bounding box area. Then we compute the mean optical flow value from all cameras.
(5) Mean optical flow difference is the difference between the current and the previous value of the mean optical flow divided by the previous value. This offers the percentage of optical flow change. We calculate the features for each camera and we keep the maximum value over all cameras.
(6) Max entropy histogram difference, $\max(\Delta H(t))$, is based on the Shannon entropy, $H(t)$, that is a measure of the uncertainty associated with a random variable. This means that the more a given pdf resembles a uniform pdf, the greater the entropy value. The main idea is that when an abrupt motion occurs, the differences in entropy's values will be significantly greater than those of a normal slow motion.
(7) Max standard deviation difference, $\max(\Delta SD(t))$, is also calculated from the object blob's histogram. Standard deviation of the histogram (std) is a measure of the spread of its values. The change on a

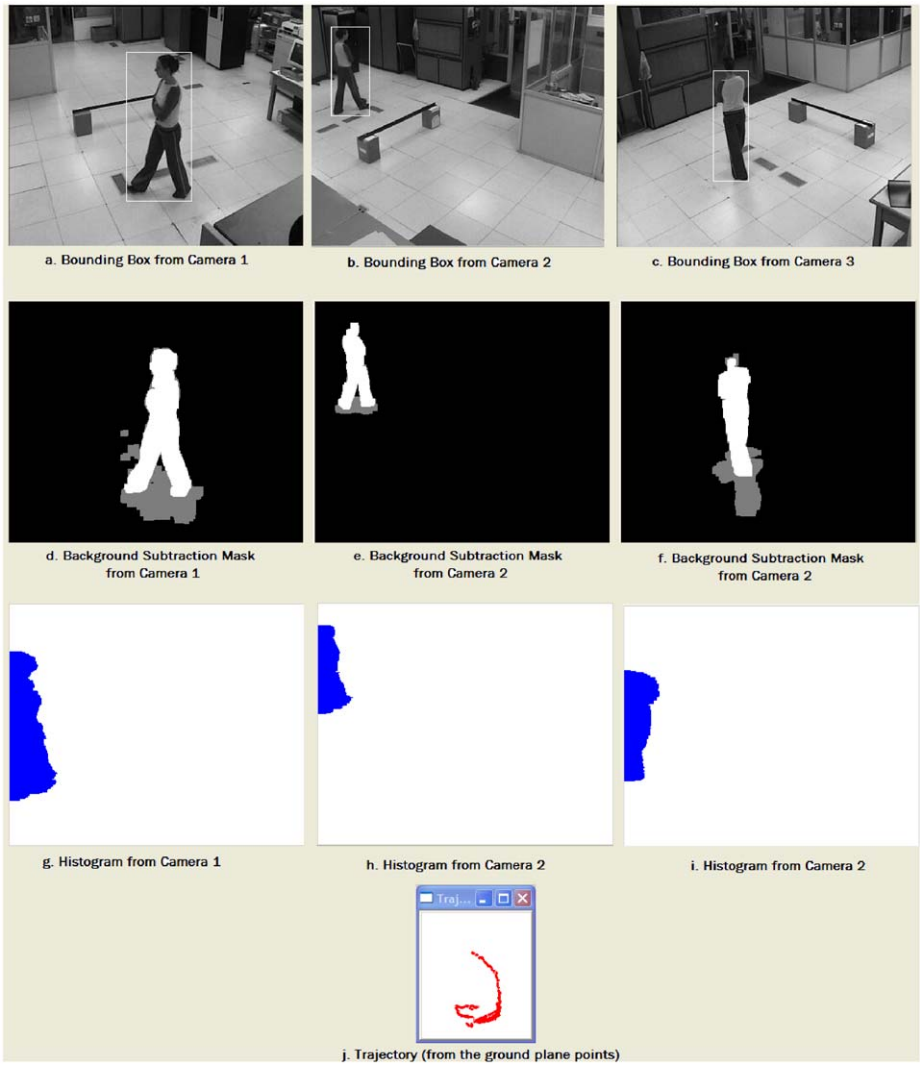

Fig. 4. (a)–(c) Frames captured from each camera with bounding boxes. (d)–(f) Background subtraction masks and blob indication per camera. (g)–(i) Histogram of the object's blob for each camera. (j) Trajectory formed by the calculated ground points.

histogram's standard deviation value from one point of view,$\Delta SD(t)$, can give us important information for the motion of the object in that it indicates within-bounding-box movement. We calculate the features for each camera and we keep the maximum value over all cameras as the final feature value.

5.2. Short-term behaviours classification

The decision whether a short-term behaviour is normal or not can be taken by employing an one-class SVM as proposed by Scholkopf [31]. The selected model does not require a labelled training set to determine the decision surface. The one-class SVM is similar to the standard SVM in that it uses kernel functions to perform implicit mappings and dot products and that the solution is only dependent on the support vectors. Such an approach can be justified by the fact that normal behaviours are easier to observe and thus whatever deviates from them can be defined as abnormal. Thus we do not need to model explicitly abnormal behaviours and we do not need labelling of data, as long as our assumption on the sparsity of abnormality stands. This is what makes this approach unsupervised.

The one-class SVM builds a boundary that separates the training data class from the rest of the feature space. For more details the reader is referred to [23].

6. Trajectories classification

Our second information source for evaluating behaviour is the trajectory. In a museum scenario, the trajectory of a person entering from the designated entrance, then approaching the cashier to buy a ticket,

Table 1
Features calculated and used for classification.

Features	Type
1. Speed	$v(t)=\sqrt{(x(t)-x(t-1))^2+(y(t)-y(t-1))^2}$
2. Algebraic mean speed	$\widehat{v_T}(t)=\sqrt{\left(\frac{1}{T}\sum_{i=t-T+1}^{t}v_x(i)\right)^2+\left(\frac{1}{T}\sum_{i=t-T+1}^{t}v_y(i)\right)^2}$
3. Algebraic mean bounding box difference	$R(t)=\frac{\sum_{i=1}^{numCam}R_i(t)_{T'}}{numCam}$ where $R_c(t)_{T'}=\frac{1}{T'}\sum_{j=t-T'+1}^{t}\frac{w_c(j)\cdot h_c(j)-w_c(j-1)\cdot h_c(j-1)}{w_c(j-1)\cdot h_c(j-1)}$
4. Mean optical flow	$F(t)=\frac{\sum_{i=1}^{numCam}F_i}{numCam}$ where F_i is the normalized optical flow from camera i
5. Mean optical flow difference	$\Delta F(t)=\frac{F(t)-F(t-1)}{F(t-1)}$
6. Max entropy difference	$\max(\Delta H(t))=\max_i\frac{H_i(t)-H_i(t-1)}{H_i(t-1)}$, with $1\leqslant i\leqslant numCam$ where $H_c(t)=-\sum_{j=1}^{N}p_c(y_j)\cdot\log p_c(y_j)$ with $p_c(y_j)$ the histogram value in y_j location for camera c and N the bounding box's height
7. Max standard deviation	$\max(\Delta SD(t))=\max_i\frac{std_i(p_i(y)_t)-std_i(p_i(y)_{t-1})}{std_i(p_i(y)_{t-1}}$, difference with $1\leqslant i\leqslant numCam$

then browsing into the room and looking around, and finally exiting from the designated exit should be characterized as normal. Trajectories of persons entering from the exit without first visiting the ticket stand, or going the wrong direction should be labelled as abnormal.

Some works in literature use rules to define the restricted areas and therefore distinct normal from abnormal trajectories. We apply an one-class learning strategy, as in the short-term behaviours, by training our time series classifier using only the normal trajectories. Each sample is a position vector (x,y) of the target in the global coordinate system in each frame (calculated as described in Section 4). The extracted normal trajectories (sequences of (x,y) vectors) are used for training a continuous Hidden Markov Model [29] and constitute the model observations.

For convenience, we use the compact notation $\lambda=(\mathbf{A},\mathbf{B},\pi)$ to indicate the complete parameter set of the model, where:

- A is the state transition probability distribution matrix.
- B is the observation probability density function per state matrix.
- π is the initial state probability distribution.

The original Baum Welch algorithm is used for the training step, while for the recognition step we propose a modified Forward Backward procedure (see Section 6.2). The methodology presented here proposes solution to two problems:

- the use of the Hidden Markov Model as an one-class classifier.
- the efficient likelihood calculation in the forward--backward for long sequences, taking into account current machine limitations.

6.1. One-class continuous Hidden Markov Model

The problem of discriminating between normal/abnormal trajectories concerns the definition of a measure that would give sufficiently different values for the two classes. The variable length of the trajectories poses additional difficulties. Long, normal trajectories would have cHMM generation probability values comparable to small values of short, abnormal trajectories, so the observation's length factor needs to be removed.

If we can prove that for a normal observation sequence (O_{normal}) and for an abnormal one ($O_{abnormal}$) the following condition must hold:

$$\frac{\log P(O_{abnormal}|\lambda)}{length(O_{abnormal})}\leqslant\frac{\log P(O_{normal}|\lambda)}{length(O_{normal})} \tag{5}$$

then we will be able to use it as a classification measure. In (5) the logarithms help us sharpen the differences between values below 1, and the division with the sequence's length normalizes the computed measure.

The anomaly detection problem begins with the definition of "what can be labelled as normal". We may define as normal the trajectories that between two time instances t and $t+1$, the probabilities of the corresponding observations are proportional to each other, and their fraction can be viewed as a random variable Δ. Taking into consideration that O_t is the observation sequence from $time=0$, until $time=t$, the random variable Δ depends only on the model, $\lambda(A,B,\pi)$ [29].

Thus, given the model and two consecutive observations O_t, O_{t+1}, there is a variable Δ, with an expected value $\delta=E[\Delta]$ such that

$$P(O_{t+1})\simeq\delta\cdot P(O_t)\Rightarrow\frac{P(O_{t+1})}{P(O_t)}\simeq\delta \tag{6}$$

with $0<t+1\leqslant T$. This assumption is derived by the facts that:

- Δ depends only on the model;
- normal trajectories have a high probability of being generated by the model;
- the expected value represents the average amount one "expects" as the outcome of the random trial when identical odds are repeated many times.

We can also see that, $0<\delta\leqslant 1$ because $P(O_{t+1})\leqslant P(O_t)$.

According to (6), we can expand the calculations as follows:

$$\begin{aligned}&P(O_{t+1})\simeq\delta\cdot P(O_t)\Rightarrow P(O_{t+1})\simeq\delta^t\cdot P(O_1)\\&\Rightarrow\log P(O_{t+1})\simeq t\cdot\log\delta+\log P(O_1)\\&\Rightarrow\frac{\log P(O_{t+1})}{t+1}\simeq\frac{1}{t+1}\cdot(t\cdot\log\delta+\log P(O_1))\end{aligned}$$

which results after replacing t with $t-1$ in the following:

$$\frac{\log P(O_t)}{t} \simeq \frac{1}{t}\cdot((t-1)\cdot\log\delta+\log P(O_1)),\quad \forall t: 0<t\leqslant T \tag{7}$$

As abnormal, we define the trajectories for which the probability of their corresponding Δ value will be very low. For those trajectories, we assume that there exists a transition from time k to time $k+1$ where, due to either the transition probability a_{ij} or the observation probability $b_j(O)$, the Δ value probability (i.e. the probability to have such a Δ value for the given model) decreases significantly, because the value of Δ_{k+1} for the given time point $k+1$ becomes lower than expected:

$$\exists k: \frac{P(O_{k+1})}{P(O_k)}=\Delta_{k+1},\quad p(\Delta)\ll 1,\quad \Delta_{k+1}\ll\delta \tag{8}$$

Before that k, the trajectory can be characterized as normal i.e.

$$\forall t: t<k,\quad \frac{P(O_{t+1})}{P(O_t)}=\delta \tag{9}$$

From the above we have

$$\begin{aligned} P(O_{k+1}) &= \log\Delta_{k+1}+\log(\delta^{\lambda-1}\cdot P(O_1)) \\ &\Rightarrow \frac{\log P(O_{k+1})}{k+1}=\frac{1}{k+1}\cdot(\log\Delta_{k+1} \\ &\quad +(k-1)\cdot\log\delta+\log P(O_1)) \end{aligned} \tag{10}$$

For the discrimination problem (see Eq. (5)), the following must hold:

$$\frac{\log P(O_{k+1})}{k+1}\ll\frac{\log P(O_k)}{k} \tag{11}$$

By letting $t=k$ in (7) and using (10) in (11) we have

$$\begin{aligned} &\frac{1}{k+1}\cdot(\log\Delta_{k+1}+(k-1)\cdot\log\delta+\log P(O_1)) \\ &\ll\frac{1}{k}\cdot((k-1)\cdot\log\delta+\log P(O_1)) \end{aligned} \tag{12}$$

Because k represents time, $k>0$. On the other hand Δ_{k+1} and d represent the value of the probabilities' ratio, so $0<\Delta_{k+1},\delta<1$. According to that remark we can assume that for sufficiently large sequences, e.g. for $k\leqslant 10$, $1/k\simeq 1/(k+1)$ in (12) due to the fact that $\log\delta,\log d\ll 1$. Thus, Eq. (12) can be

$$\begin{aligned} &\log\Delta_{k+1}+(k-1)\cdot\log\delta+\log P(O_1)\ll(k-1)\cdot\log\delta+\log P(O_1) \\ &\Rightarrow\log\Delta\ll 0 \end{aligned} \tag{13}$$

Since $\Delta_{k+1}\ll\delta$, Δ_{k+1} is a sufficiently small value that gives $\log\Delta_{k+1}\ll 0$. Given that (13) is valid, the initial assumption, Eq. (5), is true. Therefore, (5) can be used as criterion for abnormal trajectory detection.

6.2. Log likelihood approximation in long sequences

As mentioned previously, the continuous Hidden Markov Models have problems with long sequences. This is due to the multiplications in the Forward Backward algorithm, which is used to calculate the observation probability given the model. The constant decrease of the observation probability results to a very low value, which end up underflowing current computers' number storage. Solutions like sampling the trajectory, only partially solve the problem.

In order to tackle the problem, one may rescale the conditional probabilities using carefully designed scaling as proposed in [29]. We, however, have devised a method for the approximation of the log-probability of a long sequence that gives the advantage of computational simplicity and in parallel keeps the properties required for normal and abnormal trajectories' classification (Eq. (5)). Our approximating methodology avoids the calculation of the scaling factor and uses integer instead of real values. We have named this method *observation log-probability approximation* (OLPA).

Given the trained continuous Hidden Markov Model and within the recognition step, in order to compute the probability of a known observation sequence the Forward Backward algorithm is used [29]. This algorithm consists of the following steps:

(1) Initialization: $\alpha_1(i)=\pi_i\cdot b_i(O_1)$.
(2) Induction: $\alpha_{t+1}(j)=[\sum_{i=1}^{N}\alpha_t(i)a_{ij}]b_j(O_{t+1})$.
(3) Termination: $P(O|\lambda)=\sum_{i=1}^{N}\alpha_T(i)$.

To compensate for the constant decrease in the likelihood in long sequences we modified the above algorithm so that instead of multiplications we use additions of logarithms. Some background assumptions are given next.

By definition if $\lfloor x\rfloor$ is the floor of x number, $|\log\alpha-\lfloor\log\alpha\rfloor|<1$. Thus, we can approximate $\log P(O|\lambda)/length(O)$ with $\lfloor\log P(O|\lambda)\rfloor/length(O)$. Now, due to the fact that for long sequences $\alpha\equiv P(O|\lambda)$ is below 1 and that $\log\alpha\to -Infinity$, one may assume that $\log\alpha\simeq\lfloor\log\alpha\rfloor$. This approximation is acceptable, because the estimation error is bounded (less than 1). Long normal sequences give small values of cHMM probabilities, due to successive multiplications, making the logarithm of those probabilities to be too high to let the 1 to be damaging. Assuming this approximation is acceptable, it can be inserted to Forward Backward algorithm.

First, we define functions necessary for computations in cHMM algorithms, using logarithms:

$$\begin{aligned} \lfloor\log(a\cdot b)\rfloor &= \lfloor\log a+\log b\rfloor\simeq\lfloor\lfloor\log a\rfloor+\lfloor\log b\rfloor\rfloor \\ &= \lfloor\log a\rfloor+\lfloor\log b\rfloor \end{aligned}$$

Additionally the following applies for a sequence of x_i, the bigger of which is x_{max}:

$$\begin{aligned} &x_{max}\leqslant\sum x_i\leqslant n\cdot x_{max} \\ &\Rightarrow\log(x_{max})\leqslant\log\Big(\sum x_i\Big)\leqslant\log(n)+\log(x_{max}) \end{aligned}$$

The order of magnitude for x_i is 10^{-9} or less and for n is 10, so $\log(\sum x_i)\simeq\max_i(\log(x_i))$ or $\lfloor\log(\sum x_i)\rfloor\simeq\lfloor\max_i(\log(x_i))\rfloor$.

According to all the above we can conclude to a modification of Forward Backward algorithm, using the same dynamic programming idea: let $Log a\equiv\lfloor\log a\rfloor$, and $\tilde{\alpha}$ be the approximated α, then the following approximations apply:

- $\tilde{\alpha}_1(i)=\mathrm{Log}(\pi_i\cdot b_i(O_1))=\lfloor\log\pi_i+\log b_i(O_1)\rfloor$

- $\tilde{\alpha}_t(i) = \mathrm{Log}((\sum_{j=1}^{N} \alpha_{t-1}(j) \cdot a_{ij}) \cdot b_j(O_t))$
  $= \lfloor \log((\sum_{j=1}^{N} \alpha_{t-1}(j) \cdot a_{ij}) \cdot b_j(O_t)) \rfloor$
  $= \lfloor \log(\sum_{j=1}^{N} \alpha_{t-1}(j) \cdot a_{ij}) + \log b_j(O_t) \rfloor$
  $\simeq \lfloor \log(\sum_{j=1}^{N} \alpha_{t-1}(j) \cdot a_{ij}) \rfloor + \lfloor \log b_j(O_t) \rfloor$
  $\simeq \max_j(\lfloor \log \alpha_{t-1}(j) \cdot a_{ij} \rfloor) + \lfloor \log b_j(O_t) \rfloor$
  $\simeq \max_j(\lfloor \log \alpha_{t-1}(j) + \log a_{ij} \rfloor) + \lfloor \log b_j(O_t) \rfloor$
  $\simeq \max_j(\lfloor \log \alpha_{t-1}(j) \rfloor + \lfloor a_{ij} \rfloor) + \lfloor \log b_j(O_t) \rfloor$
  $= \max_j(\tilde{\alpha}_{t-1}(j) + \lfloor \log a_{ij} \rfloor) + \lfloor \log b_j(O_t) \rfloor$

- $\tilde{P}(O|\lambda) = \mathrm{Log}(\sum_{i=1}^{N} \alpha_T(i))$
  $= \lfloor \log \sum_{i=1}^{N} \alpha_T(i) \rfloor$
  $\simeq \max_i(\lfloor \log \alpha_T(i) \rfloor)$
  $= \max_i \tilde{\alpha}_t(i)$

According to the above approximations, we can express the algorithm as follows.

(1) Initialization: $\tilde{\alpha}_1(i) \simeq \lfloor \log \pi_i \rfloor + \lfloor \log b_i(O_1) \rfloor$.
(2) Induction:
$\tilde{\alpha}_t(i) \simeq \max_j(\tilde{\alpha}_{t-1}(j) + \lfloor \log a_{ij} \rfloor) + \lfloor \log b_j(O_t) \rfloor)$.
(3) Termination: $\tilde{P}(O|\lambda) \simeq \max_i \tilde{\alpha}_t(i)$.

This observation log-probability approximation helps us overcome the problem of consecutive multiplications, by making it possible to use sum of integers. Our achieved goal was to be able to calculate an approximation of the probability of a long sequence that would otherwise be impossible to compute, due to machine limitations.

7. Experiments

As a scene for our experiments we have used our lab, where we installed three cameras, as illustrated in Fig. 5a, and there we tried to simulate some common scenarios.[2] We have simulated a protected exposition room, where only one visitor is allowed and he or she has to follow a certain path for entering and exiting. Also, only certain short-term behaviours are allowed. As short-term behaviour we label the action taken by a single person within a time period of 25 frames that correspond approximately to 1 s in real world. An artificial barrier inserted in the scene does not allow entering the experiment area from a certain side and there also exists an "emergency exit". When someone visits areas which are not allowed, we consider to have a case of abnormal activity (see Fig. 5b). Similarly, when areas are visited in the wrong order (e.g. entering from the exit or exiting from the entrance) according to the modelled continuous Hidden Markov Model, this activity is also labelled as abnormal. Furthermore, we consider normal short-term activity to be something like "walking", "standing still" or "active" and in no case "running" or "abrupt motion". The experiments measure the performance of two variations of our process, namely the offline and the real-time process.

Our cameras are the AXIS 214PTZ (network cameras), from which the frames are received through HTTP requests. The communication with the cameras is performed through an IP network. For frame synchronization we used a Network Time Protocol (NTP) server which gives time stamps to each frame, so the closest frame triplet is considered to match a single time frame.

In our system, we use the LibSVM [8] library to train an one-class SVM model with a radial basis function (RBF) kernel. The training set consists of feature vectors of *normal behaviours only*. The radial basis function has been chosen based on experimental results, where we had used all the alternatives (polynomial, linear, sigmoid). SVM parameters were also optimized through trial and error.

In order to calculate the features associated with the optical flow, many restrictions were taken into consideration and various normalizations were applied, to avoid noise and reduce the computational cost. Problems were mainly due to our baseline background subtraction, as well as to the noise in the cameras' unfiltered image data.

To the end of reducing the computational cost, we have limited the optical flow's calculation only in the foreground regions. We have also used edge detection to avoid noise in the extraction of the optical flow. It is well known that the optical flow vectors may have high values in background regions that become unoccluded by a moving target, even though these regions do not move at all. This would significantly affect our classification scheme and had to be avoided. To overcome this problem, we have applied the Canny method for edge detection [25] within the blobs' boundaries (see Fig. 6). Then, we have calculated the optical flow only for the pixels belonging to these edges.

Due to the complex background, edges from the background added noise to our calculations, thus we have made use of some of the first frames from each video in order to extract background edges and subtract them from the final optical flow calculation. This choice is justified by the fact that we expect to have the highest amount of optical flow around the edges, while the optical flow is expected to be low within homogeneous regions, thus the most useful information for our classification is not lost. In Fig. 6, you can see all the processing steps described here. It should be noted that the learning process based on the first few frames can be considered as part of the initial system calibration (also see Section 4).

As already indicated, the background in our input data was natural (non-uniform) and we had to deal with noise. In our experiments we used classic surveillance cameras with low resolution (352×288), while the images captured were compressed with JPEG compression method, resulting to loss of image quality and to the creation of artifacts that sometimes affected the background subtraction. Therefore, we used the a priori knowledge of a human target's size in order to avoid bounding boxes of inexact sizes. The trivial rule used was that the bounding box can have a maximum width and height and all other bounding boxes were to be omitted. The threshold for

[2] The custom corpus used within our experiments can be made available to any interested party, via e-mail correspondence.

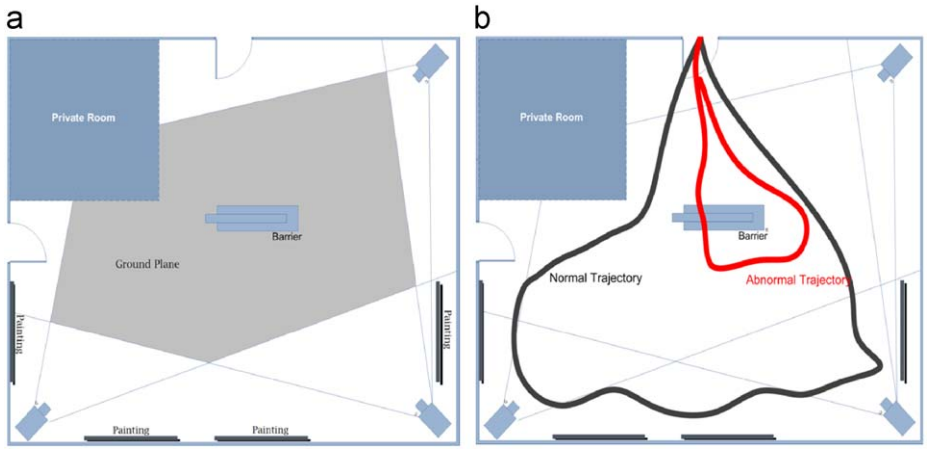

Fig. 5. (a) View of our experimental room (exposition room). (b) Normal and abnormal trajectory example. In the latter the target goes over the barrier.

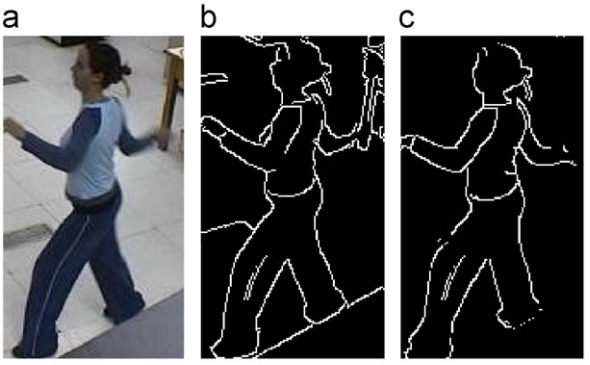

Fig. 6. (a) Foreground object inside its bounding box. (b) Edges extracted with Canny inside the bounding box. (c) Edges extracted with Canny inside the box without the edges of the background.

considering the size of a bounding box as acceptable was experimentally determined. Obviously, this heuristic is dependent on the input video and has serious defects, for example in the case where a target human lies on the floor or extends his hands. A more robust approach for background detection and removal should be used to eliminate the limitations posed.

In order to determine which of our features were the most promising for the desired classification setting we used a subset of our data, where both normal and abnormal instances had been labelled. Using an information gain criterion and a 10-fold cross validation methodology, we have found that the most promising feature is the *max entropy difference* (see Table 1 in Section 5.1). The overall ranking of the other features based on the information gain criterion is: algebraic mean speed, max standard deviation difference, speed, mean optical flow, algebraic mean blob rate and mean optical flow rate. Of course the labelled data were only used in this process, which we hoped would offer more intuition on what features offer higher discriminative potential.

7.1. Testing the one-class cHMM assumption

To see whether the $P(O_{t+1})/P(O_t)$ ratio for normal trajectories can be described based on a predefined probability density function, that can in turn be represented by its expected value, we trained a cHMM model with normal trajectories only.[3] Then, we generated several sequences O using this cHMM. These sequences should obviously be considered normal. We then calculated the ratio $P(O_{t+1})/P(O_t)$ for all values of t, i.e. all subsequences of individual O sequences. In Fig. 7 we show the results of the logarithm of probabilities $\log P(O_{t+1})/P(O_t)$ to offer more detail, since the magnitude of the probability values is very low. What Fig. 7 shows is that a normal distribution appears to offer a good approximation of the actual distribution of ratio values, even though the ratio values appear to be bounded.

7.2. OLPA performance

For long observation sequences we expect, based on the analysis in Section 6.2, that our probability calculation algorithm (OLPA) will give us results strongly correlated to the results the Forward Backward procedure returns. Experiments show that, indeed, the Forward Backward algorithm and the OLPA algorithm have strongly correlated results in short observation sequences as well. We have performed a t-test to show that the mean values of the distributions of the normalized-log P (returned by the Forward Backward algorithm) and normalized-Log P (returned by OLPA) are the same within statistical error (p-value <0.05). Additionally, we have calculated Pearson and Kendall correlation (to allow for non-Gaussian data) between the two probability estimations and, as is illustrated in Fig. 8, the samples of the two distributions are very strongly correlated (>0.98), with a p-value much lower than the usual threshold of 0.05.

[3] We have used the JaHMM library [13].

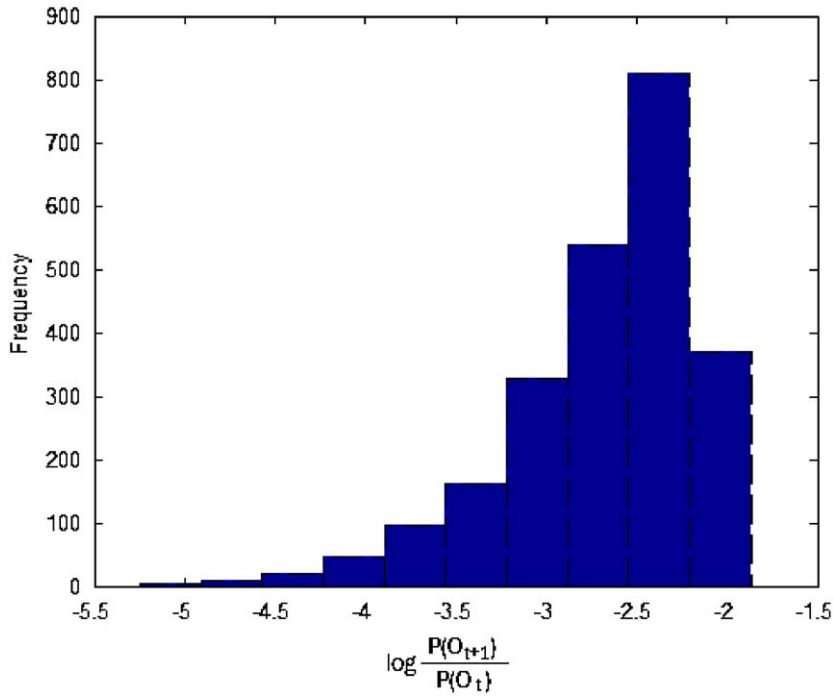

Fig. 7. Fraction of logarithms of cHMM probabilities in normal trajectories.

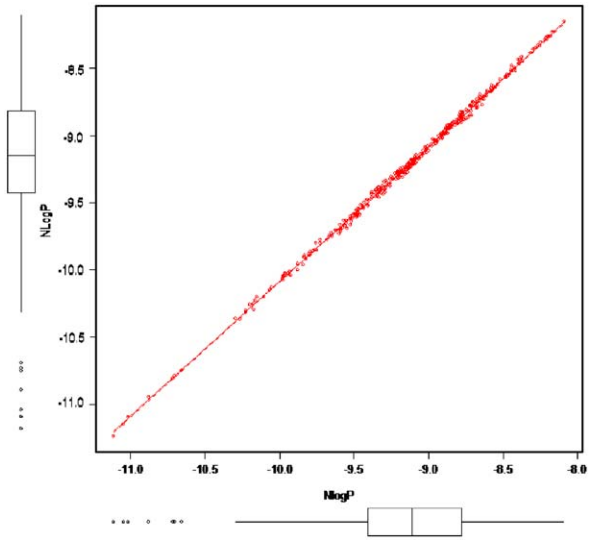

Fig. 8. Correlation between samples of the two distributions, normalized log P (Forward Backward algorithm) and normalized Log P (OLPA).

Our next experiments were performed in two steps, offline training and testing, and real-time testing.

7.3. *Offline experiments*

We have performed a 10-fold cross validation method to test the effectiveness of our system using the offline approach. Fifteen videos with normal and five videos with abnormal behaviours were captured. Each of the videos lasts between 3000 and 6000 frames and contains one to five different long-term behaviours, resulting in a total of 42 normal behaviours and 22 abnormal behaviours. Each behaviour has been performed by one of three different actors, through random selection. Out of the 22 abnormal behaviours, 14 are abnormal based on the motion features (e.g. abrupt motion) and 19 are abnormal based on the trajectory—which means that some behaviours are abnormal for both criteria used. It should be noted that the same activities performed by different actors can differ greatly. The videos with normal behaviours illustrate a person entering the room, buying a ticket, browsing and looking around for several minutes and exiting the room using a preset path. The abnormal behaviours consist of running, abrupt motion or unexpected trajectory.

Our experiments, for offline testing, consist of a test set formed by four normal behaviours per fold, as well as 22 abnormal behaviours that were used in all the folds. In the offline procedure each classifier makes a decision of the whole behaviour's abnormality. The system signals abnormality if any of the constituent classifiers has indicated abnormality.

The final decision of the observed behaviour's abnormality is taken by thresholding both classifiers' (SVM and cHMM) outputs. The thresholds are automatically calculated in the training step, which takes place offline before the operation of our system. To be more specific, during the training step, videos with normal behaviours are input to the system, features are calculated and two classifier models (one-class SVM and cHMM) are trained and stored. Then, using n-fold cross validation to ascertain

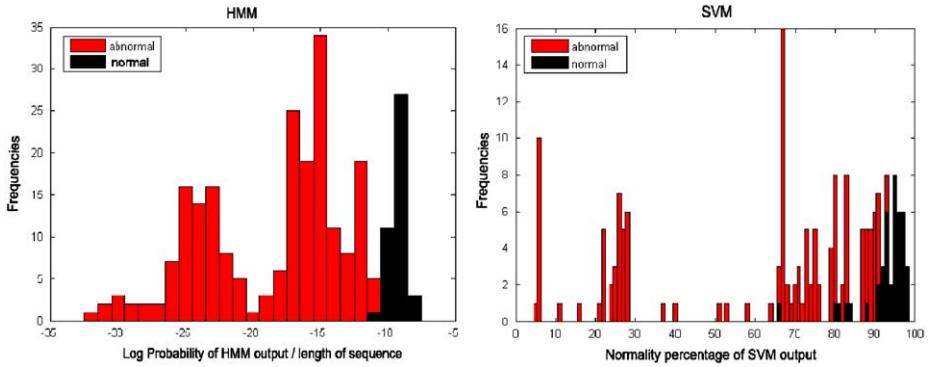

Fig. 9. (a) Percentage normality in normal and abnormal behaviours for support vector machine. (b) Output of continuous Hidden Markov Model for normal and abnormal behaviours. Black colour is for normal behaviours and red for abnormal behaviours. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

generality, the cHMM's output probabilities are stored in order to be processed and used to extract the thresholds based on distributional characteristics (mean value, standard deviation and minimum value; also see Eq. (14)). For the decision concerning the SVM classifier, we also extract a threshold which indicates the maximum number of abnormal frames we allow within a normal, predefined length sequence of frames. Therefore, SVM decisions are also used to determine this second threshold. At this point the system is considered to be calibrated. In case someone wishes to apply the system at a different location, only the training step needs to be repeated and the system will be applicable to the new environment.

The experiments prove that the system is highly automated, as minimum human interference is needed during the training step and the results are very encouraging. We remind the reader that in the background subtraction step the first 250 frames are used for training, where no person is inside the scene. Those frames are used to extract the background edges (also see Section 5.1). Features identifying short-term behaviour are extracted and used to train an one-class SVM with a radial basis function kernel. Simultaneously, trajectories were extracted in order to be inserted into a continuous HMM for training.

The threshold values have been calculated based on the training test. In Fig. 9, distributions of SVM and cHMM outputs for normal as well as abnormal behaviours are shown. Fig. 9a depicts the normality percentage for normal and abnormal behaviours within a time window that includes the whole behaviour, i.e. how many feature vectors are recognized as normal in the entire behaviour. We used a t-test in order to ensure that the two density functions are different and the resulted p-value was $<1\%$. Because of the fact that the two pdfs are not Gaussian, we have also applied the Kolmogorov–Smirnov test or KS-test [3] that does not require normal pdfs. The Kolmogorov–Smirnov test indicated that, indeed, the normal and abnormal samples come from different pdfs (p-value $= 2.09e-07$). Fig. 9b shows the cHMM's output for normal and abnormal behaviours. The two tests (t-test and KS test) were also applied to these results with both p-values substantially below 1%. According to the remark that normal and abnormal pdfs are different for both classifiers, thresholding their outputs was a logical decision.

For SVM-based classification we set the threshold to be the following function of the mean and the standard deviation of the distribution of the number of allowed abnormal frames within a normal sequence:

$$threshold_{SVM} = mean(Hsvm_{normal}) - 2.5 \cdot std(Hsvm_{normal}) \quad (14)$$

For HMM outputs the minimum value of the distribution of normalized log-probabilities of the normal instances was considered to be the threshold value that separates normal trajectories from the abnormal ones:

$$threshold_{HMM} = \min(Hhmm_{normal}) \quad (15)$$

where Hsvm is the histogram of SVM's outputs and Hhmm is the histogram with HMM's outputs.

7.4. Real-time experiments

In both the online and offline approaches the same training set (therefore the same models) and thresholds have been used. The only difference is that in the online approach we had the system emit a decision for every frame instead of for the whole behaviour. The system performance in both approaches is encouraging, as will be shown in the following paragraphs.

Real-time experiments follow a slightly different approach. Each frame is labelled as normal or abnormal depending on both classifiers' decision. All the videos contain 34 479 normal frames, i.e. frames for which the behaviour should be judged as normal, and 5260 abnormal frames. From the 4537 frames 1251 have motion-based abnormality and 4537 have trajectory-based abnormality. The SVM classifier classifies a frame, but the SVM-based decision also takes into account the labels of the previous 24 frames, based on the percentage of abnormal frames within this history of 25 frames. The cHMM returns a normalized log-probability value which characterizes the object's sampled trajectory since the object's first appearance in the scene and up to the current frame. The final system result for each frame is the logical

"or" of these two outputs, where the value "true" indicates a decision of abnormality for a given frame.

7.5. Overall system performance

Precision and recall have been calculated for the offline and the real-time experiments. For each approach we give the performance for both the SVM and HMM classifier models separately, as well as for the whole system in Table 2.

Even though the overall system performance is very satisfactory, we should note that the precision of motion-based abnormal instances, through the use of the SVM classifier, appears to be low. This indicates that we should further optimize SVM parameter values to the given classification problem, as it has been seen in literature that SVM performance can be highly dependent on the selected parameters. However, the simultaneous use of both classifiers helps the system perform highly for the given dataset.

7.6. Multiple cameras vs. one camera

To clarify the reasons for using multiple cameras instead of one camera, we have performed a set of experiments only with the data of one camera from our lab dataset. The system's results (precision and recall) are shown in Table 3. As we can see the system's performance is lower than the one produced by multiple cameras, due to the fact that one camera is not able to give as robust ground point estimation of the object as the estimation given by multiple cameras. Moreover, multiple cameras provide the benefit of more information, especially in the case where the object is not within the view of one of the available cameras.

Table 2
Precision and recall for the 3-camera system on our dataset.

	SVM		HMM		Overall	
	Precision	Recall	Precision	Recall	Precision	Recall
Offline						
Normal	0.9048	0.9286	1	0.9762	1	0.9286
Abnormal	0.7674	0.7071	0.95	1	0.88	1
Real-time						
Normal	0.9875	0.9228	0.9960	0.9770	0.9960	0.9105
Abnormal	0.2419	0.6788	0.8478	0.9704	0.8478	0.9375

The column "Overall" indicates the performance of the combined decision.

Table 3
Average precision and recall for the single-camera system on our dataset.

	SVM		HMM		Overall	
	Precision	Recall	Precision	Recall	Precision	Recall
Offline						
Normal	0.9788	0.8375	1	0.95	1	0.8
Abnormal	0.6708	0.9464	0.913	1	0.7366	1
Real-time						
Normal	0.9945	0.9148	0.9953	0.9597	0.9975	0.8525
Abnormal	0.2696	0.8569	0.7544	0.9637	0.5042	0.9861

Table 4
Precision and recall of the single-camera system, when applied on the CAVIAR dataset.

	Offline overall		Real-time overall	
	Precision	Recall	Precision	Recall
Normal	0.8882	0.775	0.7625	0.7309
Abnormal	0.3129	0.5125	0.2273	0.2582

It is worth pointing out that in Table 3 we average precision and recall taking into account two of our three cameras, due to the fact that the third camera could not give us proper output since the object was frequently out of its view. The multi-camera system overcomes this problem by compensating for any missing camera data. In addition, as we can observe from Tables 2 and 3, cHMM precision and recall in both offline and real-time experiments, are greater with multiple cameras than with only one camera. On the other hand, precision and recall in both offline and real-time experiments for SVM are in most cases higher in the single camera system than in the multi-camera system. These observations have led us to two main conclusions. The first is that our assumption that multiple cameras provide us with a more precise position of the object (more accurate trajectory) is correct. The second is that our application of *trivial* fusion of motion data from different cameras—we just calculated mean feature values over the three cameras—can cause a decrease of performance and should be avoided. Future work should research how motion feature values from different cameras should be combined.

In order to further allow for solid comparison, we have chosen to use a commonly used dataset for additional comparisons. The corpus chosen is the set of video sequences available for result comparison from the PETS04 workshop [12]. The sequences have already been used by the CAVIAR project. The system's performance when applied on these data is depicted in Table 4. It is worth mentioning that:

- the scenarios in this dataset are different from the scenarios we have assumed.
- no restricted areas have been defined therefore cHMM performance is not included in the results, since the results of the cHMM indicated normal trajectories and were, therefore, useless.
- in the CAVIAR dataset, there is no explicit definition of normality and abnormality. Thus, we have considered "running" and "fighting" to be abnormal, while all the rest were considered to be normal.

From the CAVIAR dataset we have used only videos from a single camera view. There were 11 normal behaviour

videos[4] and 4 abnormal.[5] The extracted different behaviours were a total of 43 normal and 8 abnormal ones. The number of frames was 12 188 normal and 2669 abnormal.

In the CAVIAR dataset evaluation of performance, the detection of abnormal behaviour appears to be more difficult than in our dataset. Given this difference in performance, we have sought the reasons for the decrease in efficiency and found some possible causes. In our use of the CAVIAR dataset, we used the whole videos described as cases of "walking", "browsing" and "meeting" as input for normal behaviour. We then discovered that a quick (running) motion can be found within a walking video, inducing noise in the discriminative ability of the speed-based features. Then we saw that occlusion may have caused problems, due to the fact that there are data from only one camera. The edge-detection process and the optical flow extraction fail when, for example, two people are too close the each other and fighting. In these cases the positioning of the targets with respect to the camera highly affects the method concerning the use of optical flow, but *only* when a single camera is used. The use of three cameras and proper fusion of information may offer better optical edge detection and, thus, optical flow values. The two identified problems partially explain the loss of recall for abnormal instances, even though more experiments should be conducted to verify these findings. One final comment would be that abnormality in such actions as fighting can be detected much more easily if one uses interaction information between actors, which was not within the scope of this work.

8. Conclusion and future work

In this paper, we have presented a set of theoretical and practical tools for the domain of behaviour recognition, which have been integrated within a unified, automatic, bottom-up system based on the use of multiple cameras performing human behaviour recognition in an indoor environment, without a uniform background. The approach's innovation is fourfold:

- We propose the application of two different criteria of human behaviour's abnormality used within a single methodology that needs only normal data for training.
- We have proven that the application of multiple cameras can be fruitful, when it comes to determining abnormality based on the trajectory.
- We have presented a methodology that lets a continuous Hidden Markov Model function as an one-class classifier, with very promising experimental results.
- We have accomplished to offer an alternative to the Forward Backward algorithm for the recognition step of cHMMs in order to overcome arithmetic underflow in the case of very long observation sequences, without loss of precision.

[4] Namely the normal videos were: browse1-browse4, wk1-wk3, meetSplit3rdGuy,meetWalkSplit,meetWalkTogether1-meetWalkTogether2.

[5] Namely the abnormal videos were: FightChase, FightOneManDown, FightRunAway1-FightRunAway2.

Our experimental results demonstrated the good performance of the system in the task of recognizing human behaviour's abnormality in a somewhat noisy environment, with different scenarios of action and participation of different actors. The experiments were implemented in offline and real-time conditions, with similar results, implying the robustness of the method. Furthermore, experiments with a single camera version of the system provide us the incentive to consider another, more robust method for the fusion of data in order to improve performance.

The multiple camera methodology has, so far, been tested on scenarios with only one object inside the scene, without taking account any interactions between actors. It would be worthwhile to further investigate the effectiveness of our system using more features, such as the distance of the object from each camera, in order to improve the motion-based discriminatory performance of the system. However, other methodologies could also be tested in the place of the SVM classifier.

Acknowledgements

This work is being co-funded by the Greek General Secretariat of Research and Technology and the European Union via a PENED project.

References

[1] F. Bashir, A. Khokhar, D. Schonfeld, View-invariant motion trajectory-based activity classification and recognition, Multimedia Systems 12 (1) (2006) 45–54.
[2] F. Bashir, W. Qu, A. Khokhar, D. Schonfeld, HMM-based motion recognition system using segmented PCA, in: Proceedings of IEEE International Conference on Image Processing (ICIP), Genoa, Italy, vol. 3, 2005, pp. 1288–1291.
[3] Z. Birnbaum, F. Tingey, One-sided confidence contours for probability distribution functions, The Annals of Mathematical Statistics 22 (4) (1951) 592–596.
[4] M. Black, A. Jepson, A probabilistic framework for matching temporal trajectories: condensation-based recognition of gestures and expressions, in: Proceedings of European Conference on Computer Vision (ECCV), Freiburg, Germany, vol. 1406, 1998, pp. 909–924.
[5] F. Bobick, W. Davis, The recognition of human movement using temporal templates, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (3) (2001) 257–267.
[6] O. Boiman, M. Irani, Detecting irregularities in images and in video, International Journal of Computer Vision 74 (1) (2007) 17–31.
[7] C. Bregler, J. Malik, Learning appearance based models: mixtures of second moment experts, Advances in Neural Information Processing Systems 9 (2) (1997) 845.
[8] C. Chang, C. Lin, LIBSVM: a library for support vector machines, Software available at: ⟨http://www.csie,ntu.edu.tw/cjlin/libsvm⟩, vol. 80, 2001, pp. 604–611.
[9] H. Dee, D. Hogg, Detecting inexplicable behaviour, in: British Machine Vision Conference, London, UK, 2004, pp. 477–486.
[10] T. Duong, H. Bui, D. Phung, S. Venkatesh, Activity recognition and abnormality detection with the switching hidden semi-Markov model, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, vol. 1, 2005, pp. 838–845.
[11] A. Efros, C. Berg, G. Mori, J. Malik, Recognizing action at a distance, in: Proceedings of Ninth IEEE International Conference on Computer Vision (ICCV), Nice, France, vol. 2, 2003, pp. 726–733.
[12] R. Fisher, The PETS04 surveillance ground-truth data sets, in: International Workshop on Performance Evaluation of Tracking and Surveillance, 2004.

[13] J. Francois, Jahmm-hidden Markov model (hmm): an implementation in java, 2006.
[14] B. Horn, B. Schunck, Determining optical flow, Artificial Intelligence 17 (1–3) (1981) 185–203.
[15] W. Hu, D. Xie, T. Tan, A hierarchical self-organizing approach for learning the patterns of motion trajectories, IEEE Transactions on Neural Networks 15 (1) (2004) 135–144.
[16] W. Hu, M. Hu, X. Zhou, T. Tan, J. Lou, Principal axis-based correspondence between multiple cameras for people tracking, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (4) (2006) 663–671.
[17] A. Ivanov, F. Bobick, Recognition of visual activities and interactions by stochastic parsing, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 852–872.
[18] F. Jiang, Y. Wu, A. Katsaggelos, Abnormal event detection from surveillance video by dynamic hierarchical clustering, in: IEEE International Conference on Image Processing (ICIP), San Antonio, TX, USA, vol. 5, 2007, pp. 145–148.
[19] N. Johnson, D. Hogg, Learning the distribution of object trajectories for event recognition, Image and Vision Computing 14 (8) (1996) 609–615.
[20] D. Kosmopoulos, P. Antonakaki, K. Valasoulis, D. Katsoulas, Monitoring human behavior in an assistive environment using multiple views, in: 1st International Conference on Pervasive Technologies Related to Assistive Environments PETRA' 08, Athens, Greece, 2008.
[21] C. Lee, M. Ho, W. Wen, C. Huang, T. Hsin-Chu, Abnormal event detection in video using N-cut clustering, in: International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), Pasadena, CA, USA, 2006.
[22] D. Mahajan, N. Kwatra, S. Jain, P. Kalra, S. Banerjee, A framework for activity recognition and detection of unusual activities, in: Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP), 2004, pp. 15–21.
[23] L. Manevitz, M. Yousef, One-class SVMs for document classification, Journal of Machine Learning Research 2 (2) (2001) 139–154.
[24] T. Moeslund, A. Hilton, V. Krüger, A survey of advances in vision-based human motion capture and analysis, Computer Vision and Image Understanding 104 (2–3) (2006) 90–126.
[25] H. Neoh, A. Hazanohuk, Adaptive edge detection for real-time video processing using FPGAs, CD proceedings at the 2004 Global Signal Processing Expo (GSPx) and International Signal Processing Conference (ISPC), Santa Clara, California, September 27-30, 2004.
[26] N. Nguyen, H. Bui, S. Venkatsh, G. West, Recognizing and monitoring high-level behaviors in complex spatial environments, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CSCCVPR), vol. 2, 2003, pp. 620–625.
[27] J. Owens, A. Hunter, Application of the self-organising map to trajectory classification, in: Proceedings of IEEE International Workshop Visual Surveillance, Dublin, Ireland, 2000, pp. 77–83.
[28] S. Park, M. Trivedi, Analysis and query of person–vehicle interactions in homography domain, in: Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks, Santa Barbara, CA, USA, 2006, pp. 101–110.
[29] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (1989) 257–286.
[30] P. Ribeiro, J. Santos-Victor, Human activity recognition from video: modeling, feature selection and classification architecture, Beijing, in: Proceedings of the International Workshop on Human Activity Recognition and Modelling, 2005, pp. 61–78.
[31] B. Scholkopf, J. Platt, J. Shawe-Taylor, A. Smola, R. Williamson, Estimating the support of a high-dimensional distribution, Neural Computation 13 (7) (2001) 1443–1471.
[32] G. Sukthankar, K. Sycara, Automatic recognition of human team behaviors, in: Proceedings of Modeling Others from Observations, Workshop at the International Joint Conference on Artificial Intelligence (IJCAI), Edinburgh, Scotland, 2005.
[33] T. Wang, J. Li, Q. Diao, W. Hu, Y. Zhang, C. Dulong. Semantic event detection using conditional random fields, in: Conference on Computer Vision and Pattern Recognition Workshop (CVPRW), 2006, pp. 109–114.
[34] D. Weinland, R. Ronfard, E. Boyer, Motion history volumes for free viewpoint action recognition, in: IEEE International Workshop on Modeling People and Human Interaction, 2005.
[35] T. Xiang, S. Gong, Video behavior profiling for anomaly detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (5) (2008) 893–908.
[36] D. Zhang, D. Gatica-Perez, S. Bengio, I. McCowan, Semi-supervised adapted HMMs for unusual event detection, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, 2005, pp. 611–618.
[37] H. Zhong, J. Shi, M. Visontai, Detecting unusual activity in video, in: Proceedings of IEEE Computer Vision and Pattern Recognition, vol. 2, 2004, pp. 819–826.
[38] H. Zhou, D. Kimber, Unusual event detection via multi-camera video mining, in: Proceedings of the 18th International Conference on Pattern Recognition (ICPR), vol. 3, 2006, pp. 1161–1166.
[39] Z. Zivkovic, F. van der Heijden, Efficient adaptive density estimation per image pixel for the task of background subtraction, Pattern Recognition Letters 27 (7) (2006) 773–780.