Object and scene-centric activity detection using state occupancy duration modeling

Murtaza Taj and Andrea Cavallaro*
Queen Mary, University of London
Mile End Road, London E1 4NS (UK)
{murtaza.taj,andrea.cavallaro}@elec.qmul.ac.uk

Paper ID 66

Abstract

We propose a video event analysis framework based on object segmentation and tracking, combined with a Hidden Semi-Markov Model (HSMM) that uses state occupancy duration modeling. The observations generated by a multi-object detector and tracker are used as emitting symbols and the corresponding probabilities are computed using multivariate Gaussians. Next, we recognize events by estimating the most likely object state sequence using a HSMM decoding strategy, based on the Viterbi algorithm. Moreover, the duration distribution enforces the state transition after certain time and hence better models the events constrained on time intervals. We demonstrate and evaluate the proposed framework on a dataset of approximately 20K frames, and show that the duration modeling improves the event detection results by 7% to 11%, compared to state-of-the-art HMMs.

1. Introduction

The automated analysis of large volumes of data is of great interest for indexing and retrieval of surveillance videos. Algorithms capable of detecting objects and events of interest based on motion patterns and semantic understanding are highly desirable to summarize videos or to trigger alarms.

Video event detection algorithms can be classified into three main groups, namely 3-D model-based, temporal templates and trajectory-based. *3D model-based* approaches treat an object as a set of connected parts and perform detections on their activities ([4]). The activities can be modeled as generalized action cylinders ([15]) or volumetric features ([13]). *Temporal templates* use sequences of simple events to model more complex events. Examples of temporal template methods are Petri Nets ([9]) and Belief Networks ([12]). Activity and plan prototypes are also used to recognize object behaviors through perceptual processing ([6]). A temporal template generated using recency of motion in a sequence can also be used for complex event recognition ([8]). *Trajectory-based* techniques perform event detection by analyzing trajectories over certain time spans ([10]). Abnormal behaviors can be detected by performing outliers detection using unsupervised clustering ([3]). Since events are generally composed of specific sequences of operations, HMMs are appropriate to model them ([22]). Hidden Markov Models are also used to perform abnormal activity detection in crowds ([2]) by modeling normal motion paths using single HMM ([2]) or Mixture of Gaussian HMMs ([1]). The main limitation of the above mentioned HMM-based techniques is the use of an *evaluation* strategy to obtain the sequences of events, as this result in a dependence on the selected pattern.

The event detection problem can be decomposed into three main steps: (i) the *extraction* of objects of interest, (ii) the *tracking* of the objects, and (iii) the *detection* of events generated by the tracked objects. As event detection can be modeled as a random process that is segmental in nature, the piecewise stationarity assumption of Hidden Markov Models (HMMs) is well suited for event modeling.

In this paper we improve the video event analysis approach of [20] by using Hidden Semi-Markov Model (HSMM) with time distribution modeling. The time distribution allows to incorporate in the model the dependency on time for triggering events and enables a smoother state transition than a thresholded decision. Moreover, the proposed approach can be applied as object-centric or scene-centric model to better fit the events of interest. The

*This work was supported in part by the EU, under the FP7 project APIDIS (ICT-216023).

object-centric approach is used to model events associated to objects whereas the scene-centric approach suits well for environment-dependent events.

The paper is organized as follows. Section 2 provides an overview of our detection and tracking algorithms used to extract the objects of interest. The proposed event detection approach is described in Section 3, followed by the experimental results that are presented in Section 4. Finally, in Section 5 we draw the conclusions.

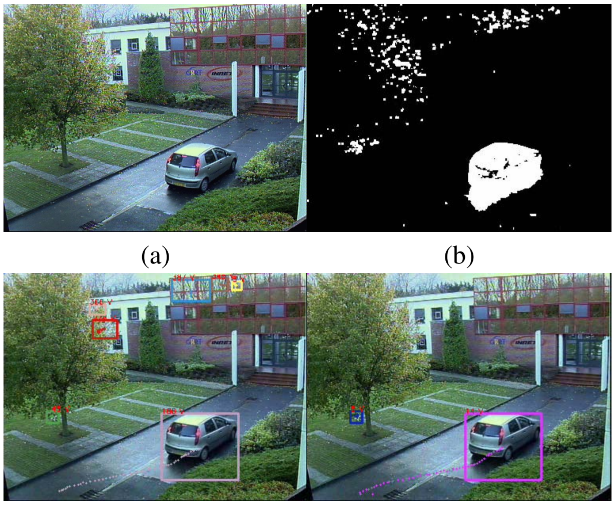
2. Object extraction and tracking

Let an object detection module generate a set of R objects $O_t = \{O_t^1, O_t^2, \cdots, O_t^R\}$ at time t. The problem is to associate objects between consecutive frames to establish the track $X_t^r = \{O_{t_0}^r \dots O_t^r\}$, up to time t, of each object O_t^r. The event analysis is then performed based on the tracks and on the available contextual information.

We perform video object extraction (foreground segmentation) using a statistical color change detector ([7]), a model-based algorithm that assumes additive white Gaussian noise introduced by the camera. The noise amplitude is estimated for each color channel separately. Given a reference image (i.e., an image without objects or an image generated by an adaptive background algorithm ([19])), the algorithm removes the effect of the camera noise based on the hypothesis that the additive noise affecting each image of the sequence respects a Gaussian distribution, with zero mean and standard deviation σ_t. The value of σ_t is computed on-line by analyzing the image difference in areas without moving objects. After the background/foreground classification, any isolated noise pixel is removed using dilation and erosion.

An important problem is that moving vegetation and fast illumination changes reduce the accuracy of the object extraction results by introducing false positive detections. For this reason, we filter the detections using a Probability Hypothesis Density filter (*PHD filter*) ([14]), which helps eliminating temporally inconsistent false positives and smoothing the results of the detections (Figure 1). Once the objects are extracted, we associate objects across consecutive frames in order to establish the track X_t^r for object r up to time t. The trajectory X_t^r is estimated with a graph matching algorithm ([21]).

Let $\{X^r\}_{r=1\dots R}$ be a set of R object detections, $v(\mathbf{x}_i^\alpha) \in V_i$ the set of vertices representing the detected objects at time i, and $e(v(\mathbf{x}_i^\alpha), v(\mathbf{x}_j^\beta)) \in E$ the set of edges of the graph $G(V, E)$. Edges represent all possible track hypotheses. Each $v(\mathbf{x}_i^\alpha)$ belongs to D, a bi-partitioned digraph (i.e., a directional graph). The candidate correspondences at different observation times are described by the gain g associated to the edges that link the vertices. The best set of tracks is computed by finding the maximum weighted path cover of G. This step can be performed using the algorithm by Hopcroft and Karp ([11]) with complexity $O(n^{2.5})$, where n is the number of vertices in G. After the maximization procedure, a vertex without backward correspondence models a new object, and a vertex without forward correspondence models a disappeared object. The depth of the graph K determines the maximum number of consecutive miss detected or occluded frames during which an object track can still be recovered.



(a) (b)

(c) (d)

Figure 1. *Comparison of object extraction results with and without the PHD Filter. (a) Original frame; (b) change detection result without using the PHD Filter; (c) bounding boxes of the detected objects without using the PHD Filter (6 false detections); (d) bounding boxes of the filtered objects after applying the PHD Filter (5 false detections have been removed).*

The gain g between two vertices is computed using the information in X_i, where the elements of the set X_i are the vectors $\mathbf{x}_i^\alpha$ defining $\mathbf{x}$, the state of the object $\mathbf{x} = (x, y, \dot{x}, \dot{y}, h, w, H)$, where (x, y) is the center of mass of the object, $(\dot{x}, \dot{y})$ are the vertical and horizontal velocity components, (h, w) are the height and width of the bounding box, and H is the color histogram. The gain for each couple of nodes, $(\mathbf{x}_i^\alpha, \mathbf{x}_j^\beta)$, is computed based on the position, direction, appearance and size of a candidate object ([21]).

This process results in the track X_t^r for each object r. The tracks of the objects are then used for event analysis, as described in the next section.

3. Event analysis

3.1. HSMM with duration modeling

Let $\lambda = \{A, B, \omega\}$ be a continuous distribution first-order Hidden Markov Model, where $\omega = \{\omega_1, \cdots, \omega_N\}$ represents the events (states) to be detected (we denote the actual state at time t as $\omega(t)$); $A = \{a_{ij}\}$ represents the state

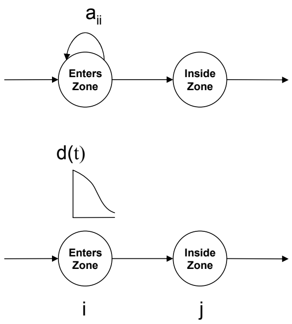


Figure 2. *Examples of self-transition modeling for an Hidden Markov Model: (top) self-transition probability (a_{ii}); (bottom) self-transition replaced with a state occupancy duration pdf.*

transition probabilities, with $a_{ij} = P[\omega(t+1) = \omega_j | \omega(t) = \omega_i]$, $1 \leq i, j \leq N$; $B = \{b_{jt}\}$ represents the emission probabilities, with $b_{jt} = P[O^r_t | \omega(t) = \omega_j]$. The emitting symbols of each state are provided by the track X^r_t of the observation O^r_t of object r.

Given the model λ and the observation sequence, we can obtain the associated optimal state sequence. Given the probability of the best sequence up to time t, the single most probable state sequence $\omega(t+1)$ at time $t+1$ can be obtained as

$$\delta_i(t) = \max_{\Omega_{t-1}} P(\Omega_{t-1}, \omega(t) = \omega_i, O_1 \cdots O_T), \qquad (1)$$

where $\Omega_{t-1} = \{\omega(1) \cdots \omega(t-1)\}$ and $O_1 \cdots O_T$ are the observations from time 1 to T.

$$\delta_i(t) = \max_{1 \leq t \leq T-1} P(\omega(t) = \omega_i | O, \lambda). \qquad (2)$$

According to the Markovian assumption, the conditional probability distribution of future states depends on the current state only and not on past states, hence using the Forward Viterbi we have

$$\delta_j(t+1) = \max_{1 \leq i \leq N} [\delta_i(t) a_{ij}] b_{j,O^l}. \quad 1 \leq l \leq T \qquad (3)$$

Finally, we compute the most likely hidden state sequence ω_T up to time $t+1$ as

$$\omega(t+1) = \arg \max_{1 \leq i \leq N} [\delta_i(t) a_{ij}]. \qquad (4)$$

This simple Hidden Markov Model is unable to completely model certain events due to the duration distribution of the observation sequence for a certain state. The Markovian assumption constraints the state occupancy distribution to be exponentially distributed ([16]). Therefore the estimation of the most likely path ω_T is problematic, because a state with high self-transition probability a_{ii} can cause the algorithm to stay in this state for a longer interval. To avoid such self-transitions, we use Hidden Semi-Markov Models (HSMM) ([18]) to enable the explicit modeling of duration probability distribution $d(t)$. The duration probability distribution is the probability of staying at least for a duration τ in the state ω_j, with $1 \leq \tau \leq D_j$ (Figure 2). To compute the most likely state sequence ω_T using the durational distribution, we maximize the joint probability $P[O, \omega^1_T | \lambda]$ by re-writing Equation 2 as

$$\delta_i(t) = \max_{O, 1 \leq t \leq T-1} P[\omega(t) = \omega_i, O | \lambda]. \qquad (5)$$

Using the forward Viterbi algorithm we can solve Equation 3 as

$$\delta_j(t+1) = \max_{\substack{1 \leq j \leq N \\ 1 \leq \tau \leq D_j}} [\max_{1 \leq i \leq N} [\delta_i(t) a_{ij}] d(t) b_{j,O^l}], \qquad (6)$$

with $1 \leq l \leq T$. Given the model $\lambda = \{A, B, \omega\}$ and the duration probability distribution $d(t)$, we can now use Equation 6 to compute the best state sequence by performing the HMM decoding using the Viterbi algorithm. The state transition probabilities a_{ij} can be defined empirically or, if there is sufficient training data, can be calculated using the Baum-Welch algorithm ([5]). In order to use the Viterbi algorithm we need first to model the duration probability distribution $d(t)$ and the observation sequence.

3.2. Duration probability distribution

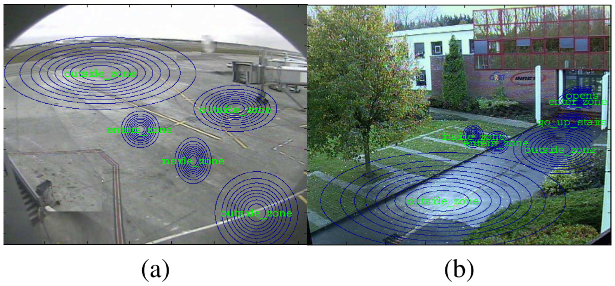
The duration probability distribution $d(t)$ can be modeled using different parametric duration distributions. We evaluate two distributions, namely the *half-normal distribution* and the *triangular distribution*, which are well adapted to the problem at hand. The half-normal distribution, $d_n(t)$, can be expressed as

$$d_n(t) = \frac{1}{\sigma} \sqrt{\frac{2}{\pi}} \exp\left(-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2\right), \qquad (7)$$

where σ is the variance, computed as $3\sigma = \tau$, and μ is the mean. The mean is the time t_e when the object transits into the state and $t_e \leq t \leq T + t_e$. The triangular distribution, $d_t(t)$, can be expressed as

$$d_t(t) = \frac{2(\tau + t_e - t)}{\tau^2}. \qquad (8)$$

In case of events with high self-transitions a uniform distribution can be used which implicitly converts HSMM to HMM. The selection of the appropriate distribution, for the specific event or activity, can be done using Chi-square test. The evaluation and analysis of the results obtained using half-normal and triangular distributions are discussed at the end of Section 3.3 and in Section 4.



(a) (b)

Figure 3. *Multivariate scene-centric distribution model. (a) Sequence AP-11 C4; (b) Sequence BE-19 C1.*

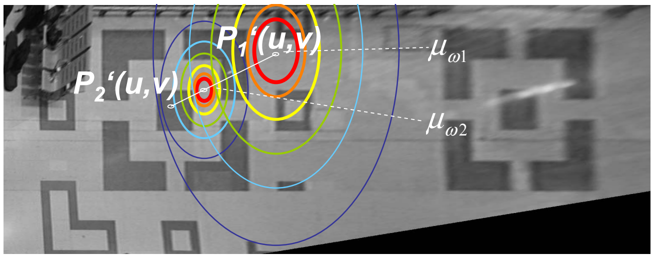


Figure 4. *Multivariate object-centric distribution model. The distribution of the states is placed on the line joining the centroids of the objects.*

3.3. Object-centric and scene-centric models

For the observation sequence model, the emitting symbol is the observation of the r^{th} object O_t^r at time t. The emission probabilities b_{jt} are modeled as a continuous function describing the state. We propose and evaluate two models to estimate b_{jt}, namely a scene-centric and an object-centric model. In the *scene-centric* approach, the b_{jt} are modeled as a multivariate Gaussian. For each j^{th} state we use a multivariate Gaussian $\mathcal{N}_j(\mu, \Sigma)$ with mean μ and covariance Σ as

$$b_{jt} = \frac{1}{\sqrt{(2\pi)^n |\Sigma_j|}} \exp\left(-\frac{1}{2}(X-\mu_j)^T \Sigma_j^{-1} (X-\mu_j)\right), \tag{9}$$

where $n = 4$, $\mu_j = \{\mu_x, \mu_y, \mu_w, \mu_h\}$ and $|\Sigma_j|$ is the determinant of the covariance matrix Σ_j, which we assume to be a diagonal matrix: $\Sigma_j = diag[\sigma_x^2, \sigma_y^2, \sigma_w^2, \sigma_h^2]$. The values for μ_j and Σ_j are set based on the contextual information specific to the task at hand (Figure 3).

In the *object-centric* approach, we model b_{jt} as a multivariate distribution composed of a mixture of a normal and a uniform distribution $\mathcal{N}_j(\mu, \Sigma, \rho, C, D)$ with mean μ, covariance Σ, weight ρ and range of uniform distribution $[C, D]$:

$$b_{jt} = \frac{\rho}{(2\pi)^{\frac{K}{2}} |\Sigma_j|^{\frac{1}{2}}} \exp\left(\sum_{k=1}^{K}\left[\frac{(\theta_k - \mu_{\theta_k})^2}{2\sigma_{\theta_k}^2}\right]\right) + \\ + \frac{(1-\rho)}{\pi} \prod_{k=1}^{K} \left[\frac{\psi_{\theta_k}}{\sigma_{\theta_k}}\right], \tag{10}$$

where K=2; θ_1=x and θ_2=y. Therefore σ_x and σ_y are the standard deviations, respectively. The functions ψ_k are piecewise binary and defined as

$$\psi_x = \begin{cases} 1 & \text{if } C_x < x < D_y \\ 0 & \text{otherwise} \end{cases}, \tag{11}$$

and

$$\psi_y = \begin{cases} 1 & \text{if } \zeta(C_x) < y < \zeta(D_y) \\ 0 & \text{otherwise} \end{cases}, \tag{12}$$

where $\zeta = \pm\sigma_y \sqrt{1 - (\frac{x - x_c}{\sigma_x})^2} + y_c$, with (x_c, y_c) representing the object centroid around which the model is built, and $\pi \prod_{k=1}^{2} \sigma_{\theta_k}$ is the area of an ellipse. $|\Sigma_j|$ is the determinant of the covariance matrix, with $\Sigma_j = diag[\sigma_x^2, \sigma_y^2]$ and therefore $|\Sigma_j| = \sigma_x \sigma_y$ in Equation 10. The values of the elements in Σ_j depend on the state to be modeled, whereas the value of μ is assigned dynamically. This is the key point of the proposed *object-centric* modeling. The value of μ of the first state is set as the centroid of the reference object O_t^{ref} (Figure 4). The remaining state distributions are then placed around O_t^{ref} to estimate the possible state of O_t^{ref} with respect to the objects O_t^r. The μ of the other states are positioned on the line passing through the centroid of the two objects (O_t^{ref} and O_t^r) at a distance that is a function of the variances of the states to be detected. The rationale for using Gaussian functions instead of hard boundaries and fixed threshold is to increase the flexibility of the algorithm in order to detect several different events in different scenarios.

Let us see two examples of event modeling using the scene-centric and the object-centric models for the PETS[1], the CAVIAR[2] and the ETISEO[3] datasets. We use the *object-centric* HMM shown in Figure 5 to model the events in the PETS and the CAVIAR datasets. In this case, we model three events, namely the *attended*, *unattended* and *abandoned* baggage event. For the PETS sequences, the baggages are detected based on their size and aspect ratio (ranging between 1 and 1.8). For the *attended baggage (ω_1)* event, $\sigma_x = \sqrt{2 * 36}$ and $\sigma_y = \sqrt{2 * 96}$ respectively, whereas for the *unattended baggage (ω_2)* and the *abandoned baggage (ω_3)* events the values are $\sigma_x = \sqrt{36/2}$ and $\sigma_y = \sqrt{96/2}$. These values are based on the calculation that, for this scenario, $1m$ in world-coordinates corresponds in the ground plane to 36 pixels along the x-axis and to 96 pixels along the y-axis. A baggage is considered *unattended* when its related object (the *owner*) is $2m$ away. A baggage is considered *abandoned* when its related ob-

[1] http://www.cvg.rdg.ac.uk/PETS2006/index.html
[2] http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/
[3] http://www.silogic.fr/etiseo/index.html

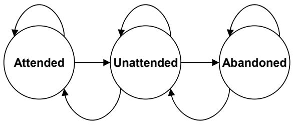


Figure 5. *HMM model for baggage detection on the PETS and CAVIAR datasets. Each state represents an event. The initial state is selected as the state with the maximum emission probability b_{jt} at time t_0.*

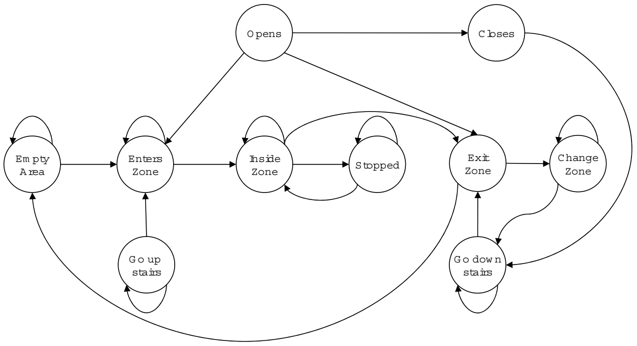


Figure 6. *Scene-centric HMM model for activity monitoring on the ETISEO dataset. Each state represents an event. The initial state is selected as the state with the maximum emission probability b_{jt} at time t_0.*

	HMM	HSMM-TRI	HSMM-HN
AP	0.882	0.980	0.956
BE	0.790	0.966	0.980
Total	0.847	0.965	0.975

Table 1. *Performance comparison between the proposed HSMM algorithm with half-normal and triangular distribution for state occupancy duration and event detection using HMM without state duration modeling.*

ject is $3m$ away, for at least 30 seconds. For the CAVIAR sequences, the baggages are detected in a similar fashion and the parameters of the events are defined as follows. For the *attended baggage (ω_1)* event, $\sigma_x = \sqrt{2 * 36}$ and $\sigma_y = \sqrt{2 * 36}$ respectively, whereas for *unattended baggage (ω_2)* the values are $\sigma_x = \sqrt{36}$ and $\sigma_y = \sqrt{48}$ and for *abandoned baggage (ω_3)* the values are $\sigma_x = \sqrt{24}$ and $\sigma_y = \sqrt{24}$.

The *scene-centric* HMM model is used for activity monitoring for the ETISEO dataset (Figure 6). In this case we model ten events, namely *enter zone*, *inside zone*, *exit zone*, *change zone*, *opens*, *closes*, *go up stairs*, *go down stairs*, *empty area*, and *stopped object*. The definition of the areas of interest is part of the contextual information provided with the dataset. The most likely state sequence ω_T^r for each object r is computed by applying the *forward Viterbi algorithm* after every 25 to 50 observations. The last state $\omega(t)$ of the state sequence is then used as the initial state $\omega(0)$ for next computation. The event detection algorithm using the *forward Viterbi algorithm* for HSMM is summarized in Algorithm 1.

Algorithm 1 *Event Detection*

$\omega = \{w_1, w_2, \ldots, w_N\}$: events (states that an object can acquire)
a_{ij} : state transition probabilities between state i to l
μ_j : mean for each state i; Σ_j : covariance matrix for each state j
X_t^r : observation for object r at time t ; $count$: counter

```
for $t = 1$ to $end$ do
    Compute: $X_t^r$
    for $j = 1$ to $n$ do
        Compute $b_{jt}^r$ :
        $b_{jt}^r = \frac{1}{\sqrt{(2\pi)^n|\Sigma_j|}} \exp\left(-\frac{1}{2}(X_t^r - \mu_j)^T \Sigma_j^{-1} (X_t^r - \mu_j)\right)$
    end for
    $count \leftarrow count + 1$
    if $count = n$ then
        Initialize initial state $\omega_0^r$
        if $\omega_0 = -1$ then
                $\omega_0^r \leftarrow \zeta(\max_{j=1\ldots l} b_{jt}^r)$    (13)
            where $\zeta$ returns $\omega_j$ corresponding to $b_{jt}^r$
        end if
        Apply Forward Viterbi Algorithm:
                $\delta(t) = \max_i [\delta^r(t-1) a_{ij}] b_{jt}^r$
            $\delta_j(t) = \max_{\substack{1 \leq j \leq N \\ 1 \leq \tau \leq D_j}} [\max_{1 \leq i \leq N} [\delta_i(t-1) a_{ij}] d(t) b_{j,O^l}]$   $1 \leq l \leq T$
                $\omega_T^r = \arg\max_i [\delta^r(t-1) a_{ij}]$
                $\omega_0^r \leftarrow \omega_t^r$
    end if
end for
```

Table 1 shows the performance comparison between the proposed algorithm (with the two different duration distributions) and the HMM-based algorithm without state duration modeling ([20]). The comparison was done on the ETISEO sequences AP-11 (C1 and C4) and BE-19 (C1) using the CREDS protocol which provides a weighted sum of the true positive, false positive and false negative detections ([17]). It is possible to notice that the duration modeling in HMM improves the results and that the modeling using the triangular distribution outperformed by 3.75% the half-normal distribution. In summary, the HSMM model with triangular distribution performed at 96%, the HSMM model with half-normal distribution obtained a score of 92.5% and the HMM scored 85%.

4. Experimental results

We demonstrate the proposed algorithm on standard event detection sequences from the PETS 2006, CAVIAR ('leaving bags behind') and ETISEO datasets. These sequences (whose details are given in Table 2) include indoor and outdoor scenarios with pedestrians, vehicles, objects and their interactions. The PETS dataset contains

Dataset	Seq.	Cameras	Resol.	N. of frames	Fr. rate
ETISEO	AP-11	C4 , C7	720x576	805 , 805	12.5
	BE-19	C1 , C4	768x576	1025 , 950	25
	RD-6	C7	720x576	1201	25
PETS	S1	C3	720x576	3022	25
	S3	C3	768x576	2372	25
	S5	C3	720x576	3402	25
	S6	C3	720x576	2802	25
CAVIAR	CL1	NA	384x288	1441	25
	CL2	NA	384x288	1357	25
Total number of frames				19182	-

Table 2. *Summary of the datasets used in the experiments.*

high-quality sequences (duration: 94 to 136 seconds), the CAVIAR dataset contains low-resolution sequences of medium quality (duration: 54 to 57 seconds), and the ETISEO dataset contains sequences of lower quality (duration: 40 to 64 seconds).

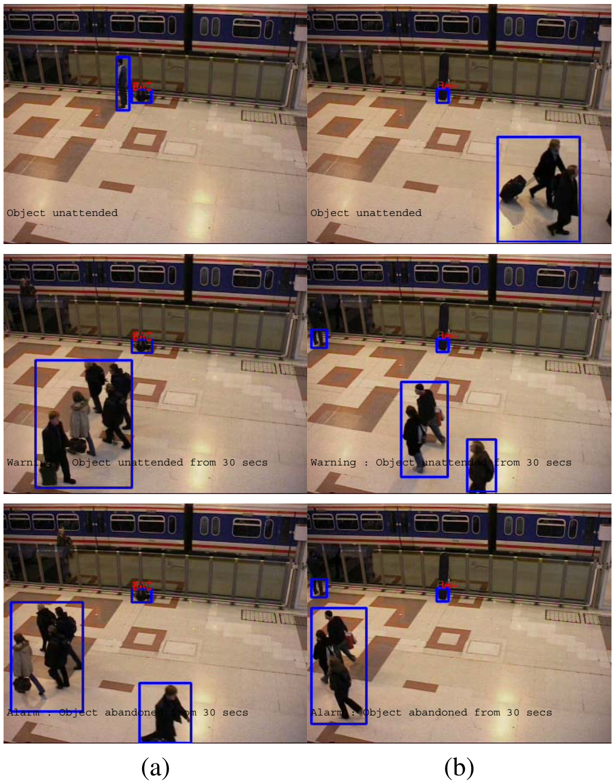

Object unattended
Object unattended
Warning : Object unattended from 30 secs
Warning : Object unattended from 30 secs
Alarm : Object abandoned from 30 secs
Alarm : Object abandoned from 30 secs
(a)
(b)

Figure 7. *Sample event detection results from the PETS 2006 dataset using the object-centric event modeling. (a) Sequence S1 (frames* 2004, 2754 *and* 2790*); (b) Sequence S5 (frames* 2083, 2833 *and* 2890*).*

To evaluate the performance of the event analysis results, we compute three measures: the *accuracy*, the *precision* and the *sensitivity*. Let FP be the number of false positive detections, TP the number of true positive detections, and FN the number of false negative detections. Moreover, let GT be the starting or ending frame number corresponding to an event in the ground truth and AD the corresponding frame number identified by the event detector for the same event. The accuracy quantifies the frame-level performance of the algorithm. The *accuracy* is defined as $\gamma = \left[1 - \frac{|GT-AD|}{NF}\right] \times 100$, where NF is a normalizing factor representing maximum allowed difference between AD and GT. Precision and sensitivity are sequence-level measures. The precision is the measure of the robustness against false positives. The sensitivity is the measure of the robustness against false negatives. The *precision* is defined as $TP/(TP + FP)$ and the *sensitivity* is defined as $TP/(TP + FN)$.

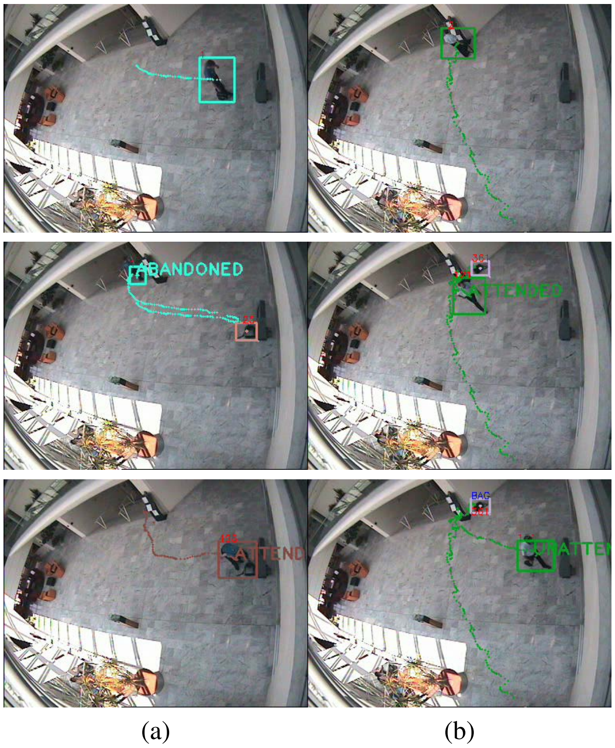
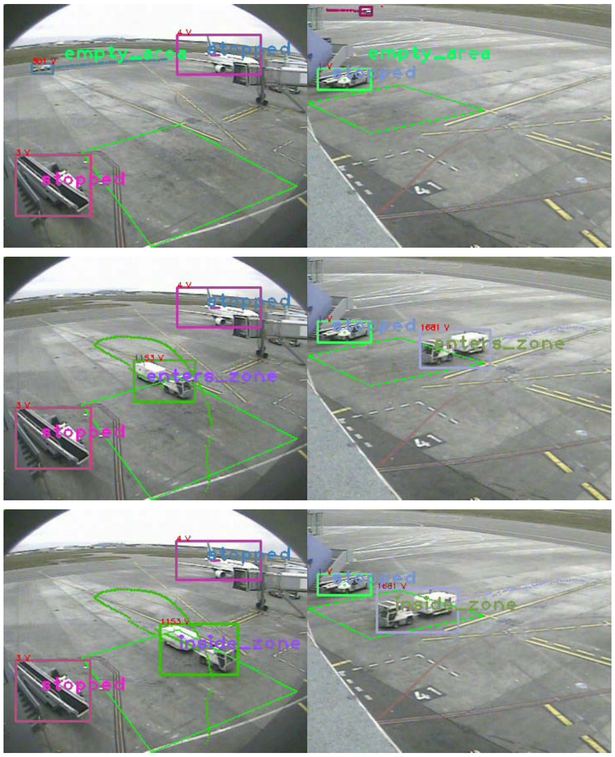


Figure 8. *Example of left baggage detection on the CAVIAR dataset using the object-centric event modeling. (a) Abandoned and attended baggage event in sequence CL1 (frame* 1014, 1070 *and* 1334*); (b) attended and unattended baggage event in sequence CL2 (frame* 548, 670 *and* 721*).*

Figure 7 shows sample event detection results on the sequences S1 and S5 of the PETS 2006 dataset. The images show the detection of the object around which the model is built (the bag) and the subsequent sequence of events, namely a *warning* (unattended baggage) and an *alarm* (abandoned baggage). To evaluate the results, we computed the accuracy of the detection: for S1, the accuracy for the *warning* event is 90.5% and for the *alarm* event is 92.9%; for S3, the accuracy is 100% for both events;



(a) (b)

Figure 9. *Sample tracking and event detection results for the ETISEO dataset using the scene-centric event modeling for ETI-VS2-AP-11 (frame* 23*,* 690 *and* 750*). The detected events are* stopped, empty area, enter zone *and* inside zone. *(a) Camera 4; (b) Camera 7.*

	Start frame			**End frame**		
	GT	**AD**	**Acc**	**GT**	**AD**	**Acc**
AP-11-C4 (Precision: 1.00, Sensitivity: 0.56)						
empty area	1	12	98.53	689	664	96.67
enters zone	675	664	98.53	720	728	98.93
inside zone	690	731	94.53	804	803	99.87
stopped	1	2	99.87	804	803	99.87
stopped	1	3	99.73	804	803	99.87
All			**98.24**			**99.04**
AP-11-C7 (Precision: 1.00, Sensitivity: 0.50)						
empty area	1	187	75.20	689	653	95.20
enters zone	675	658	97.73	720	695	96.67
inside zone	690	696	99.20	804	803	99.87
stopped	1	2	99.87	804	803	99.87
All			**93.00**			**97.90**
BE-19-C1 (Precision: 0.65, Sensitivity: 0.65)						
closes	335	371	95.20	453	450	99.60
opens	258	250	98.93	320	300	97.33
opens	366	395	96.13	400	407	99.07
stopped	270	283	98.27	1025	1024	99.87
All			**97.13**			**98.97**
BE-19-C4 (Precision: 0.87, Sensitivity: 0.35)						
inside zone	185	180	99.33	245	338	87.60
opens	77	101	96.80	150	180	96.00
opens	737	717	97.33	780	776	99.47
stopped	170	206	95.20	950	1048	86.93
All			**97.17**			**92.50**
RD-06-C7 (Precision: 1.00, Sensitivity: 0.25)						
stopped	570	559	98.53	710	743	95.60
All			**98.53**			**95.60**

Table 3. *Event detection precision and sensitivity for* 5 *test sequences of the ETISEO dataset.*

for S5, the accuracy is 88.8% and 83.02% for *warning* and *alarm*, respectively, and for S6 the accuracy is 98.5% and 95.5%. Both precision and sensitivity for PETS are unitary as the object-centric approach selects events associated with detected objects only and the baggage was detected. Figure 8 shows sample event detection results on the sequences $CL1$ and $CL2$ of the CAVIAR dataset. Figure 8(a) shows the detection of the *abandoned* and *attended baggage* events, which are generated as the person first abandoned the baggage and then reappears and approaches the baggage. Figure 8(b) shows that the person has left the baggage at the end of the stairs moving toward the kiosk machine and hence the *attended* and then *unattended baggage* events are generated. In accordance with the ground truth available for the CAVIAR dataset, we compute the accuracy of the detection of the activities related to the baggage. For $CL1$, the event initialization accuracy is 95% and the event termination accuracy is 94.66%. The event initialization accuracy for $CL2$ is 97.33% and the termination accuracy is 95.60%. The reason for these values is that the automated detection spans an interval that is a subset of the ground truth interval. This is due to the merging of the blob of the baggage with that of the person when the bag is placed on the floor. This results in a delayed detection of the event. Similarly, when the baggage is picked up, the two objects are merged thus resulting in an anticipation of the event. Improvements in the object detection accuracy will help in further enhancing the event detection accuracy. Similarly to the PETS dataset, the precision and sensitivity scores for the CAVIAR dataset are unitary as all events are detected.

Figure 9 shows detection results on the ETISEO dataset for the *enter zone, inside zone, stopped* and *empty area* events. The green rectangle drawn on the tarmac is the zone considered for triggering the events *enter zone, inside zone* and *empty area*. The *stopped* event is detected anywhere in the scene. Table 3 shows the accuracy for the detected events in all ETISEO sequences. The videos with the results for object tracking and event detection are available at http://www.elec.qmul.ac.uk/staffinfo/andrea/event.html.

5. Conclusions

We presented an event detection framework based on object-centric and scene-centric Hidden Semi-Markov modeling (HSMM). First, multiple object extraction is achieved using color based change detection followed by a PHD filter and then graph theory is used for data association. Event detection is performed on the HSMM using the Viterbi decoding strategy to estimate the sequence of events performed by each object. We showed that HSMM has better capabilities of representing events than HMM due to the embedding of the state occupancy duration modeling, which not only resulted in an improvement in accuracy of 7% to 11%, but also provides capabilities of detecting time constrained events without hard thresholds. We showed that the scene-centric approach can better model the activities associated to the contextual information, whereas activities related to the objects, irrespective of the environment, can be better modeled using the object-centric approach. The framework was evaluated on standard event detection datasets; with 19182 frames of indoor and outdoor standard test sequences.

Our current work addresses the use of proposed approach in multi-camera networks. Moreover, we will investigate the problem of event detection of multiple interacting objects.

References

[1] E. L. Andrade, S. Blunsden, and R. B. Fisher. Detection of emergency events in crowded scenes. In *IEE Int. Symp. on Imaging for Crime Detection and Prevention*, London, UK, June 2006.

[2] E. L. Andrade, S. Blunsden, and R. B. Fisher. Modelling crowd scenes for event detection. In *Proc. of IEEE Conf. on Pattern Recognition*, Hong Kong, China, August 2006. IEEE Computer Society.

[3] N. Anjum and A. Cavallaro. Unsupervised fuzzy clustering for trajectory analysis. In *Proc. of IEEE Int. Conf. on Image Processing*, San Antonio, Texas (USA), September 2007.

[4] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3d shapes from image streams. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, South Carolina, USA, June 2000.

[5] D. Burshtein. Robust parametric modeling of durations in hidden markov models. *IEEE Trans. on Speech and Audio Processing*, 4(3):240–242, 1996.

[6] C. Castel, L. Chaudron, and C. Tessier. What is going on? a high level interpretation of sequences of images. In *Proc. of the European Conf. on Computer Vision*, Cambridge, UK, April 1996.

[7] A. Cavallaro and T. Ebrahimi. Interaction between high-level and low-level image analysis for semantic video object extraction. *EURASIP Journal on Applied Signal Processing*, 6:786–797, June 2004.

[8] J. W. Davis and A. F. Bobick. The representation and recognition of human movement using temporal templates. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Washington, DC, USA, June 1997.

[9] N. Ghanem, D. DeMenthon, D. Doermann, and L. Davis. Representation and recognition of events in surveillance video using Petri nets. In *Proc. of IEEE Int. Conf. on Computer Vision and Pattern Recognition*, pages 112–112, Washington DC, USA, June 2004.

[10] S. Hongeng and R. Nevatia. Multi-agent event recognition. In *Proc. of IEEE Int. Conf. on Computer Vision*, pages 84–91, Vancouver, Canada, July 2001.

[11] J. Hopcroft and R. Karp. An $n^{2.5}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Computing*, 2(4):225–230, December 1973.

[12] S. S. Intille and A. F. Bobick. A framework for recognizing multi-agent action from visual evidence. In *Proc. of the National Conf. on Artificial intelligence*, pages 518–525, Orlando, Florida, USA, September 1999.

[13] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *Proc. of IEEE Int. Conf. on Computer Vision*, volume 1, Beijing, China, October 2005.

[14] E. Maggio, E. Piccardo, C. Regazzoni, and A. Cavallaro. Particle phd filter for multi-target visual tracking. In *ICASSP*, pages I–1101 – I–1104, Honolulu, USA, April 2007.

[15] T. S. Mahmood, A. Vasilescu, and S. Sethi. Recognizing action events from multiple view points. In *Proc. of IEEE Workshop on Detection and Recognition of Events in Video*, Madison, Wisconsin, USA, June 2001.

[16] E. Marhasev, M. Hadad, and G. A. Kaminka. Non-stationary hidden semi markov models in activity recognition. In *Proc. of the AAAI Workshop on Modeling Others from Observations*, 2006.

[17] RATP, France. *Call for Real-Time Event Detection Solutions (CREDS) for Enhanced Security and Safety in Public Transportation*, July 2005.

[18] M. Russell and R. Moore. Explicit modelling of state occupancy in hidden markov models for automatic speech recognition. In *Proc. of IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 10, pages 5–8, 1985.

[19] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22:747–757, August 2000.

[20] M. Taj and A. Cavallaro. Multi-camera scene analysis using an object-centric continuous distribution hidden markov model. In *Proc. of IEEE Int. Conf. on Image Processing*, San Antonio, Texas (USA), September 2007.

[21] M. Taj, E. Maggio, and A. Cavallaro. Multi-feature graph-based object tracking. In *CLEAR, Springer LNCS 4122*, pages 190–199, Southampton, UK, April 2006.

[22] D. Zotkin, R. Duraiswami, and L. Davis. Multimodal 3-d tracking and event detection via the particle filter. In *Proc. of IEEE Workshop on Detection and Recognition of Events in Video*, pages 20–27, Vancouver, Canada, July 2001.