Autonomous Visual Events Detection and Classification without Explicit Object-Centred Segmentation and Tracking

Tao Xiang, Shaogang Gong and Dennis Parkinson
Department of Computer Science
Queen Mary, University of London, London E1 4NS, UK
{txiang,sgg,dennisp}@dcs.qmul.ac.uk

Abstract

Modelling events is one of the key problems in dynamic scene analysis when salient and autonomous visual changes occuring in a scene need to be characterised effectively as meaningful events. We propose a new approach for modelling such temporal events based on the local intensity temporal history of pixels. The method provides a computationally very effective temporal measure for detecting autonomous events. Events are represented and detected first at the pixel level and then at a blob level (grouped pixels) autonomously. The Expectation-Maximisation (EM) algorithm is employed to cluster events with automatic model order selection using modified Minimum Description Length (MDL). Experiments are presented to demonstrate that meaningful clusters of blob-level events can be formed without object segmentation and tracking.

1 Introduction

Understanding visual behaviour is essential for dynamic scene analysis in visual surveillance and monitoring. In visual surveillance, human activities and behaviours are recorded in image sequences usually taken by a fixed camera. One of the essential tasks for an automated vision system is to model normal and abnormal behaviours. We consider that visual behaviours of activities are often composed of spatially and temporally structured autonomous visual events. By autonomous events, we imply that both the number of meaningful events and their whereabout in the scene are automatically learned and detected rather than manually labelled or hypothesised as mostly reported in the literature.

Numerous efforts have been made to model moving object behaviours in general [1, 7]. Most previous approaches for modelling behaviour heavily relied upon segmentation and tracking of object in the scene [5, 6, 11, 12]. A visual event is commonly defined based on a moving object with constraints on its size, colour or shape. A sequence of events is represented by the tracked trajectory of the object of interest. These trajectories are further clustered to form typical trajectory templates (with variance, if available) for modelling behaviour. Therefore, the entire modelling processing relies critically on the accuracy and consistency of segmentation and tracking which are often ill-conditioned due to the presence of multiple objects, occlusion and non-linearity of the trajectories.

Recently several attempts have been made to circumvented the problem of segmentation and tracking using semantical events correlation [10] and by learning localised or pixel-wise change. In particular, object grouping and segmentation were avoided by either profiling behaviour based on autonomous pixel-level events [8] or extracting features for the whole image based on pixel-level analysis [3]. However, these pixel-wise approaches can be rather sensitive to noise due to ignoring the spatial correlation among neighbouring pixels and computationally expensive due to the large number of detected events.

Alternatively, modelling events at both pixel and blob (autonomously grouped pixels) levels can be exploited in order to utilise more spatial information from the scene. To this end, the work presented in this paper focuses on modelling both pixel-level and blob-level autonomous events. In Section 2, Pixel Change History (PCH) is proposed to characterise pixel-wise temporal visual information in order to detect pixel-level events. PCH is based on the local intensity temporal history of each pixel. Crucially, it can be computed very efficiently which is essential for real-time applications. PCH is combined with an adaptive mixture background model to form a new representation for detecting and classifying pixel-level events. It also provides an important cue for characterising blob-level events which are defined on the basis of grouped pixel-level events. In Section 3, blob-level events are computed using unsupervised clustering based on Expectation-Maximisation (EM) with automatic model order selection using modified Minimum Descriptive Length (MDL). Experiments are presented in Section 4 to demonstrate that although no explicit object-centred segmentation and tracking were performed, meaningful event clusters can be formed consistently. Conclusions are drawn in Section 5.

2 Modelling Pixel-Level Autonomous Events

2.1 Pixel Change History (PCH)

Our objective is to find a suitable representation which is capable of distinguishing at the pixel level temporal changes of different nature and scale occurred in the scene. Due to the large number of pixel-wise changes we need to consider at each frame, the representation must be computationally inexpensive for real-time performance.

Temporal wavelets can be adopted for multi-scale analysis. However, the computational cost for a multi-scale wavelet at the pixel level is very expensive. Alternatively, Motion History Image (MHI) can be used to detect visual changes by keeping a history of change which decays over time. It has been used to build holistic motion templates for the recognition of human movement [2] and moving object tracking [9]. An important advantage of MHI is that although it is a representation of the history of pixel-wise changes, only one previous frame needs to be stored. It is also easy to implement and adds little computational cost to the system. However, at each pixel, information of 'old' changes will be lost when 'new' changes are present due to MHI's holistic accumulating nature. Pixel Signal Energy is another option [8]. Temporal filters were employed to measure the average magnitude of pixel-wise temporal energy over a backward window. The size of the backward window determines the number of frames (history) needed to be stored. The experiments from [8] indicate however that it is sensitive to noise and also computationally expensive.

We propose a new representation for pixel-wise change based on a combination of Motion History Image and Pixel Signal Energy. We call it Pixel Change History (PCH)

defined as:

$$P_{\varsigma,\tau}(x,y,t) = \begin{cases} min\left(P_{\varsigma,\tau}(x,y,t-1) + \frac{255}{\varsigma}, 255\right) & \text{if } D(x,y,t) = 1 \\ max\left(P_{\varsigma,\tau}(x,y,t-1) - \frac{255}{\tau}, 0\right) & \text{otherwise} \end{cases} \quad (1)$$

where $P_{\varsigma,\tau}(x,y,t)$ is the PCH for a pixel at (x,y), $D(x,y,t)$ is a binary image indicating the foreground region, ς is the accumulation factor and τ is the decay factor. When $D(x,y,t) = 1$, instead of jumping to the maximum value, the value of a PCH increases gradually through the accumulation factor. When no significant pixel-wise visual change is observed in the current frame, pixel (x,y) will be treated as part of background and the corresponding pixel change history starts to decay. The speed of decay is controlled by the decay factor ς. The accumulation factor and the decay factor give us the flexibility of characterising the pixel-wise change over time. In particular, large values of ς and τ imply that the history of visual change at (x,y) is considered over a longer backward temporal window. In the meantime, the ratio between ς and τ determines how much weight is put on the recent change. The PCH over the entire image is equivalent to Motion History Image when ς is set to 1.

Similar to the Pixel Signal Energy considered in [8], Pixel Change History captures a simple (zero order) but important feature of pixel-level change: the average magnitude of the change. However, it also captures high order features including speed, trend (uphill or downhill) and the phase of a change. In this paper, we focus on modelling the average magnitude of the pixel-wise change.

2.2 Pixel-Level Events Detection and Classification

It is commonly considered that semantics associated with a visual event largely depends on the context of an application. For example, a 'car stopping' in a car park is normal and should not be treated as an event, while on a motorway, 'car stopping' is usually abnormal. We ultimately wish to have an automated method to extract semantics (the meaning of) from visual change. For now, let us first introduce some generic semantics. For a typical scenario of a busy scene in the public place such as in a supermarket, we are interested in automatically detecting and classifying localised and persistent movement of alien objects (e.g. people stop and browse) and change of background (e.g. the introduction of static alien objects or the removal of background objects). To this end, we suggest to combine adaptive Gaussian mixture background model with PCH.

Adaptive mixture models are commonly used to memorise and maintain the background colour distribution [8, 11]. The foreground pixels detected by adaptive mixture models correspond to pixel-level changes that are either short term caused by (1) instant moving alien objects or long term caused by (2) localised movements of alien objects, (3) introduction of static alien objects or (4) the removal of background objects. However, it cannot differentiate their differences. Adaptive mixture background models are insensitive to persistent movements of background objects such as waving tree leaves. On the other hand, if the binary image $D(x,y,t)$ in Equation (1) is the temporal subtraction between the current frame and the dynamic background maintained by an adaptive mixture model, our notion of foreground pixels are then represented by PCH. To filter out the short term pixel-level changes (type (1) above) that we are not interested in, we first

define pixel-level events as foreground pixels that satisfy:

$$P_{\varsigma,\tau}(x, y, t) > T_H \tag{2}$$

where T_H is a threshold. We also define pixel-level 'events' as all the foreground pixels in order to perform an unsupervised auto-clustering for detecting blob-level events, with the short term pixel-level changes filtered out in the blob level. This will be described in detail in the next section. A comparison on the two approaches based on these two definitions of pixel-level events respectively is given in Section 4.

We further consider that pixel-level events can be classified according to the causes of change. More precisely, for any pixel-level event, if the corresponding intensity value satisfies:

$$|I(x, y, t) - I(x, y, t-1)| > T_M \tag{3}$$

where T_M is a threshold, it is caused by localised movement of alien objects. Events that do not satisfy the above condition are caused by the introduction of static alien objects or the removal of background objects. For example, a pixel-level event caused by a browsing person and a pixel-level event caused by the removal of an object from a shelf in a shopping mall may have very similar PCH value, but the former event satisfies Condition (3) while the latter does not, thus they can be differentiated.

3 Modelling Blob-Level Autonomous Events

3.1 Detection and Representation of Blob-Level Events

Behaviour profiling has been attempted directly based on pixel-level events [8]. However, the large number of events detected and the noise sensitivity caused by ignoring spatial correlation of pixel-level events hampered the success of the approach. To address this problem, we consider unsupervised grouping of pixel-level events not only according to spatial proximity but also by temporal correlation.

Let us first consider simply grouping pixel-level events spatially. The connected component method is adopted to group the detected pixel-level events into blobs, represented by bounding boxes. Small blobs are removed by a size filter. If pixel-level events refer to all the foreground pixels, only those blobs with average PCH larger than a threshold T_B will be defined as blob-level events and kept for further processing. Each blob-level event is given by a feature vector:

$$[x, y, w, h, R_f, R_m] \tag{4}$$

where (x, y) is the central position of the corresponding bounding box in the image, (w, h) is the bounding box dimension, R_f represents the percentage of the bounding box occupied by pixel-level events and R_m represents the percentage of pixel-level events which satisfy Condition (3).

3.2 Blob-Level Events Classification

After blob-level events are detected, behaviour profiling can be performed by first clustering the events into different classes. We assume that each class of blob-level event corresponds to an activity or an important phase of an activity such as the start or the

end of an activity. The classification result is thus important for further profiling of a behaviour.

In order to detect the presence of any meaningful events and their whereabout in the scene, clustering are performed in a 6-D feature space given by the feature vector defined in (4). Examples of this 6-D feature space are illustrated using the projection of the three largest principal components shown in Figure 3. Depending on the representation of events, different unsupervised clustering methods can be employed. We adopt Expectation-Maximisation (EM) with automatic model order selection using modified Minimum Description Length (MDL) principle [4].

MDL is employed to extend maximum likelihood estimation to the model order unknown situation. Let us consider there are n independent training data $\{\mathbf{y}_1, .., \mathbf{y}_n\}$, belonging to class w and $w = \{1, .., K\}$. The estimated model order $\hat{K}$ by a standard MDL algorithm is given by:

$$\hat{K} = argmin \left\{ -\sum_{i=1}^{n} lnf\left(\mathbf{y}_i | w, \hat{\theta}(K)\right) + \frac{\zeta(K)}{2} ln(n) \right\} \tag{5}$$

where $f\left(\mathbf{y}_i | w, \hat{\theta}(K)\right)$ is the class-conditional density function, $\hat{\theta}(K)$ are the mixture parameters estimated by a maximum likelihood algorithm such as EM and $\zeta(K)$ is the number of parameters needed for a K-component mixture. If full covariance matrix is used, we have:

$$\zeta(K) = K - 1 + \frac{d^2 + 3d}{2} K \tag{6}$$

where d is the dimensionality of the feature space.

The first term in the bracket of Equation (5) corresponds to the maximised likelihood, measuring the system entropy, while the second term measures the number of bits needed to encode the model parameters, serving as a penalty term for too complex mixtures (i.e. too large K). One major problem with the standard MDL lies on the fact that each component in the mixture can only 'see' the $m_j n$ data (m_j is the weight for the jth component) belonging to it, instead of the whole dataset. We adopt a modified MDL measure [4] with the model order $\hat{K}$ estimated as:

$$\hat{K} = argmin \left\{ -\sum_{i=1}^{n} lnf\left(\mathbf{y}_i | w, \hat{\theta}(K)\right) + \frac{K-1}{2} ln(n) + \frac{d^2 + 3d}{4} K ln(n) \right\} \tag{7}$$

The obtained parameters of the mixture model are used to classify blob-level events. More specifically, each correspondent feature point is classified into a class so that the Mahalanobis distance between the feature point and the mean of the class cluster is minimal.

4 Experiments

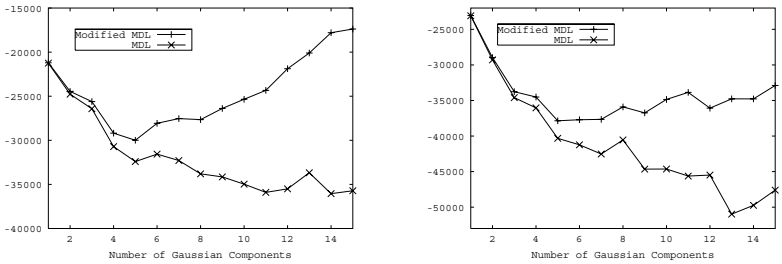
Experiments were conducted on a simulated 'shopping scenario' sequence. It is a 20 minutes video at 25Hz. Some typical scenes and autonomously detected significant events are shown in Figure 1. A shop keeper sat behind a table on the right side of the view. Drink cans were laid out on a display table. Shoppers entered from the left and either browsed

without paying or took a can and paid for it. Abnormal behaviour would be taking a can and leaving without paying. The data were sampled at 8 frames per second with total number of 5699 frames of image size 320×240.

Figure 1: Autonomous event detection in a simulated shopping scenario. Figures in the top row from left to right are the typical scenes of the shopping scenario, sampled from frame 110 to frame 330 of the 20 minutes video. Figures in the second and the third rows are a number of autonomous events detected using Approach I and Approach II respectively. Pixel-level events that satisfied Condition (3) were displayed in white and those that did not were displayed in grey. Detected blob-level events were indicated with bounding boxes.

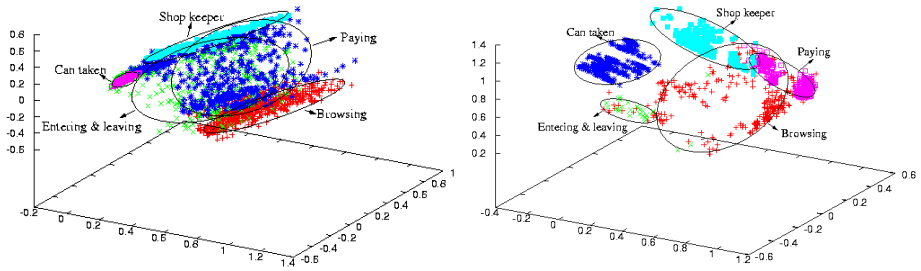
The two approaches adopted for autonomous events detection are referred as Approach I and Approach II respectively in the following. For Approach I, only those foreground pixels that satisfy Condition (2) are detected as pixel-level events and all the blobs formed are detected as blob-level events. For Approach II, all the foreground pixels are detected as pixel-level events and only those blobs with average Pixel Change History values larger than T_B are detected as blob-level events. For the adaptive Gaussian mixture background model, the parameters were set as: learning rate $\alpha = 0.002$, background model chosen threshold $T = 0.7$, six Gaussian components were maintained and a diagonal co-variance matrix was adopted. The parameters for pixel-level events detection were chosen as $\varsigma = 12$, $\tau = 10$, $T_H = 180$, $T_M = 10$ and $T_B = 100$. Only those Blobs whose sizes were larger than 40 were considered. It was observed that using both approaches, localised movements such as "shopper paying" and the removal of background objects such as "can taken" were detected automatically as significant events from visual changes, whist the normal passing-by of shoppers was ignored. For the whole sequence, 5019 and 4134 blob-level events were detected using Approach I and Approach II respectively. Some of the events detection results are shown in Figure 1. The algorithm was run on an Athelon 1.5G dual processor platform at an average speed of 6Hz.

Unsupervised learning was performed on the first 3000 frames, where 2459 events and 1922 events were detected using Approach I and Approach II respectively. EM was employed to obtain the parameters of the mixture model. It was combined with a modified MDL to determine the number of the classes of significant events in the scene and their whereabout. Figure 2 shows that 5 classes of events were automatically detected



(a) Approach I (b) Approach II

Figure 2: Automatic model order selection using MDL and modified MDL. Model orders were considered in a range of $(1, 15)$.



(a) Approach I (b) Approach II

Figure 3: Autonomous events detection and classification on the testing set.

unsupervised using either Approach I or Approach II. For comparison, automatic model order selection using standard MDL is also shown in Figure 2. Five different event classes were automatically learned in terms of their location and temporal order through unsupervised clustering, but with manual labelling to "can taken", "entering and leaving", "shop keeper", "browsing" and "paying".

A testing set was composed using the rest of the frames from the 20 minutes video. The detected and classified autonomous events from this testing set were then projected onto the three largest principal components of the 6-D feature space (shown in Figure 3). The spatial distributions of each class of events were illustrated by only showing their (x, y) co-ordinates of the central position of the corresponding bounding boxes in Figures 4 and 5.

The learned mixture models were also utilised to recognise the detected blob-level events online. The computational cost added by recognition was neglectable and the algorithm still ran at a speed of 6Hz. Although the parameters of mixture models were extracted from the training set, they were used for recognising events both in the training set and the testing set. For performance evaluation, the ground truth was labelled manually (see (a) of Figure 6). The events recognition results at each frame are shown in (b) and (c) of Figure 6. To achieve a degree of robustness in events detection and classification, an event of a particular class was considered as presence if it has been detected over a number of consecutive frames. Then, events were counted only once when they happened continuously. The performance of our algorithm was measured using the detection rate and the false detection, which is the number of results without corresponding ground truth, for each class of event. Table 1 shows the results of autonomous events detection and classification using both approaches.

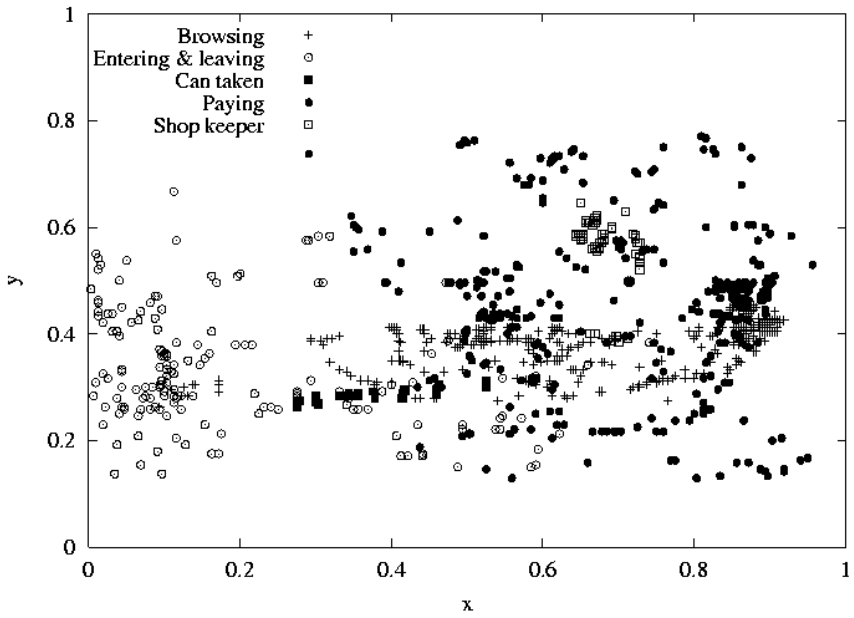


Figure 4: Classification of the testing set in the image space using Approach I. Among the 2560 blob-level events detected from the testing set, there were 929 “can taken” events, 283 “entering and leaving” events, 293 “shop keeper” events, 522 “browsing” events and 533 “paying” events.

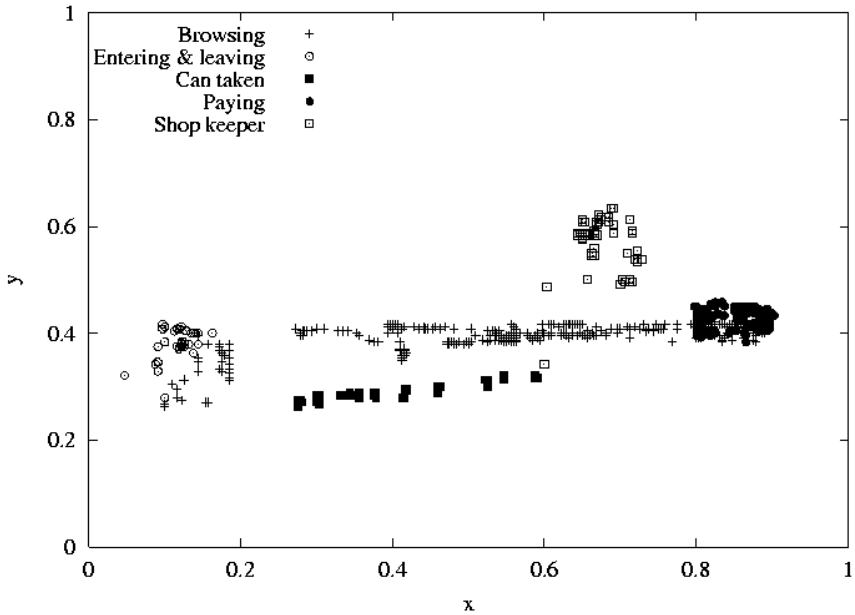
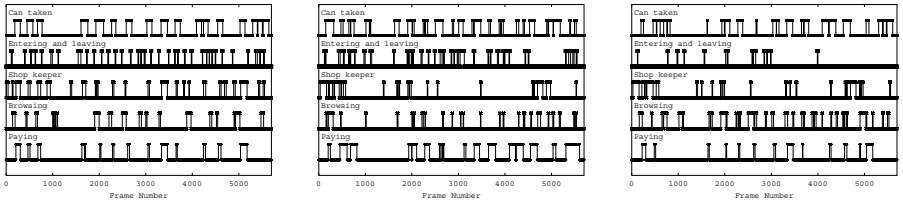


Figure 5: Classification of the testing set in the image space using Approach II. Among the 2212 blob-level events detected from the testing set, there were 1116 “can taken” events, 33 “entering and leaving” events, 316 “shop keeper” events, 406 “browsing” events and 341 “paying” events.



(a) The ground truth (b) Approach I (c) Approach II

Figure 6: Compare the ground truth with the detected blob-level events. Each "can taken" event was counted for 100 frames in the ground truth.

Events	Training set					Testing set				
	N	Det. rate		False det.		N	Det. rate		False det.	
		A I	A II	A I	A II		A I	A II	A I	A II
Can taken	7	85.7	100	0	0	10	100	100	0	0
Ent. & lev.	18	66.6	55.6	8	1	18	61.1	5.6	3	0
Shopkeeper	12	75.0	66.7	1	0	12	33.3	50.0	1	1
Browsing	10	60.0	100	3	7	8	62.5	100	9	10
Paying	8	100	75.0	6	0	6	100	100	6	1

Table 1: Events detection and recognition results. "N" stands for "number of events", "A I" and "A II" stand for Approach I and Approach II respectively, and the detection rate is in percentage.

5 Discussions and Conclusions

It can been seen from Table 1 that the events of "can taken" and "paying" were detected accurately using both approaches, as was "browsing" using Approach II. The reason for the low detection rate of "shop keeper" events was because the movements of the shop keeper were frequently occluded by the shoppers. Some shoppers entered and left the view without slowing down, thus no localised movement was performed, which resulted in the poor detection rate of "entering and leaving". Other errors were mainly in the detection of "paying" and "browsing" events. With Approach I, many "browsing" events were detected as "paying", leading to low detection rate for "browsing" and large number of false detections for "paying". With Approach II, the starting and ending phases of "Paying", as well as some "entering and Leaving" events were frequently detected as "browsing", leading to large number of false detections for "browsing". A fusion of the two approaches could give more accurate event recognition.

It was noticed that quite a lot of "paying" and "browsing" events were spatially very close and featured similar movements. This will potentially pose a problem for the current algorithm. For example, when a shopper stands in front of the shop keeper, it is impossible to tell whether he is going to pay or he is just browsing unless we take into consideration the fact that whether any drink can was or was not taken a moment ago. Even when the shopper has a can in hand, he still can walk back and continue browsing without paying. That is normal in any real shopping scenario. Perhaps we should not expect the system to resolve this ambiguity unless higher order spatio-temporal correlations between

different classes of events can be fully explored. These correlations could be both spatial and temporal. Behaviour profiling can be performed robustly when these correlations are learned.

To summarise, Pixel Change History (PCH) has been proposed as a novel effective representation for modelling autonomous visual events. Pixel-level events and blob-level events were automatically detected and classified using a combined representation of PCH and adaptive mixture background model. The experimental results show that the detected blob-level events can be classified into meaningful classes without object-centred tracking. The work done so far only represents the first step toward a more comprehensive behaviour profiling. Our future work will be focused on the fusion of different events detection approaches for more accurate autonomous events detection and recognition and the exploration of higher order spatio-temporal correlations between different classes of events for automatic extraction of high-level semantics.

References

[1] J.K. Aggarwal and Q. Cai. Human motion analysis: a review. *CVIU*, 73(3):428–440, 1999.

[2] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *IEEE Trans. PAMI*, 23(3):257–267, 2001.

[3] O. Chomat, J. Martin, and J. Crowley. A probabilistic sensor for the perception and the recognition of activities. In *Proc. ECCV*, pages 487–503, 2000.

[4] M. Figueiredo and A.K. Jain. Unsupervised learning of finite mixture models. *IEEE Trans. PAMI*, 24(3):381–396, 2002.

[5] I. Haritaoglu, D. Harwood, and L.S. Davis. w^4: Real-time surveillance of people and their activities. *IEEE Trans. PAMI*, 22(8):809–830, 2000.

[6] S. Intille, J. Davis, and A. Bobick. Real-time closed-world tracking. In *Proc. CVPR*, pages 697–703, 1997.

[7] T. Moeslund and E. Granum. A survey of computer vision-based human motion capture. *CVIU*, 81(3):231–268, 2001.

[8] J. Ng and S. Gong. Learning pixel-wise signal energy for understanding semantics. In *Proc. BMVC*, pages 695–704, 2001.

[9] J.H. Piater and J.L. Crowley. Multi-modal tracking of interacting targets using gaussian approximation. In *Proc. IEEE Workshop on PETS*, pages 141–147, 2001.

[10] J. Sherrah and S. Gong. Continuous global evidence-based Bayesian modality fusion for simultaneous tracking of multiple objects. In *Proc. ICCV*, pages 42–49, 2001.

[11] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. PAMI*, 22(8):747–758, August 2000.

[12] T. Wada and T. Matsuyama. Multiobject behavior recognition by event driven selective attention method. *IEEE Trans. PAMI*, 22(8):873–887, 2000.