Video Activity Extraction and Reporting with Incremental Unsupervised Learning

Luis Patino and François Bremond
INRIA Sophia Antipolis - 2004, route des Lucioles - BP 93
06902 Sophia Antipolis Cedex, FRANCE
{Jose-Luis.Patino_Vilchis, Francois.Bremond}@sophia.inria.fr

Murray Evans, Ali Shahrokni and James Ferryman
School of Systems Engineering, University of Reading
RG6 6AY United Kingdom
{m.evans,a.shahrokni,j.m.ferryman}@reading.ac.uk

Abstract

The present work presents a new method for activity extraction and reporting from video based on the aggregation of fuzzy relations. Trajectory clustering is first employed mainly to discover the points of entry and exit of mobiles appearing in the scene. In a second step, proximity relations between resulting clusters of detected mobiles and contextual elements from the scene are modeled employing fuzzy relations. These can then be aggregated employing typical soft-computing algebra. A clustering algorithm based on the transitive closure calculation of the fuzzy relations allows building the structure of the scene and characterize the ongoing different activities of the scene. Discovered activity zones can be reported as activity maps with different granularities thanks to the analysis of the transitive closure matrix. Taking advantage of the soft relation properties, activity zones and related activities can be labeled in a more human-like language. We present results obtained on real videos corresponding to apron monitoring in the Toulouse airport in France.

1. Introduction

The technical and scientific progress requires human operators to handle more and more quantities of data. Recordings can go to very large quantities of data either by monitoring a smart piece for long periods of time (days, weeks, ...) or in surveillance of large infrastructures with a large network of cameras, microphones and other sensors. Although most vision systems specialize on recognizing predefined events (or behaviours) , little research has been done on the field of data-mining to analyse such large quantities of data and discover the behaviours encountered and give a comprehensive analysis of the ongoing activity. While a reliable monitoring system is principally aimed at the safety/security issues, it could also be of great help for infrastructure designers and managers. For the everyday operation of the monitored space, it is important to provide environmental figures. Some situational reporting can provide the locations and numbers of people in the monitored areas (occupation map) or the user activity (e.g. parking vehicle). In this work, we aim at building a system to analyse and extract valuable information, which is generally hidden in the raw data, to 1) learn what monitored areas are normally occupied, and then 2) characterize and report activity. This is achieved mainly through trajectory analysis. The monitoring system is mainly composed of two different processing components. The first one is a real time analysis subsystem for the detection and tracking of objects. This is a processing that goes on a frame-by-frame basis. The second subsystem works off-line and achieves the activity extraction from the video. This subsystem is composed of two modules: The trajectory analysis module where we perform the analysis of trajectories by clustering, the activity analysis module where we obtain behavioural patterns of interaction and build activity maps. The featured system is adaptive with on-line learning capabilities. The remainder of this paper is structured as follows. In the rest of this section, we give a short overview of the related work. The object detection and tracking process is given in section 2. We explain how we model the scene in section 3. The methodology for trajectory clustering is given in section 4 and that for activity extraction is presented in section 5; the experimental results are to be found in section 6. Finally, Section 7 draws the main conclusions and describes our future work.

1.1. Related Work

Extraction of the activity contained in the video by applying data-mining techniques represents a field that has only started to be addressed. Although the general problem of unsupervised learning has been broadly studied in the last couple of decades, there are only a few systems which apply them in the domain of behaviour analysis. Because of the complexity to tune parameters or to acquire knowledge, most systems limit themselves to object recognition [9]. For behaviour recognition, three main categories of learning techniques have been investigated.

• The first class of techniques learns the parameters of a video understanding program. These techniques have been widely used in case of event recognition methods based on neural networks [6], Bayesian classifiers [11, 8] and HMMs [4, 1]. • The second class consists in using unsupervised learning techniques to deduce abnormalities from the occurring events [19, 20]. • The third class of methods focuses on learning behaviour based on trajectory analysis. This class is the most popular learning approach due to its effectiveness in scene and behaviour modelling [12, 17] and in detecting normal/abnormal behaviours. For example, Piciardelli et al. have proposed either an splitting algorithm [13] or single-class SVM clustering [14] applied on very structured scenes (such as roads). Anjum et al. [2] employ PCA to seek for trajectory outliers. Similarly, Antonini et al. [3] transform the trajectory data employing Independent Component Analysis (ICA), while the final clusters are found employing an agglomerative hierarchical algorithm. Hidden Markov Models (HMM) have also been employed to detect different states of pre-defined normal behaviour [4, 15]. All these techniques are interesting, but little has been said about the semantic interpretability of the results. Indeed, more than trajectory clusters, we are interested in extracting meaningful activity information with semantic, which can be interpreted. This work comes thus into the frame of behaviour extraction from trajectory analysis, however we have in addition a higher semantic level that employs proximity relations between resulting clusters of detected mobiles as well as between clusters and contextual elements from the scene to, first, build the structure of the scene and, then, characterize the ongoing different activities of the scene. Employing such proximity relations represents a novel contribution in the domain of behaviour learning.

2. Real-time processing object detection and tracking

The detection and tracking is performed using multiple cameras with an overlapping field of view, and consists of three stages: Detection in the image plane, tracking in the image plane, fusion and tracking in 3D.

2.1. Detection

Detection is performed by combining change detection and motion detection. The first detector is the Adaptive Gaussian Mixture Model of Zivkovic [21]. This method builds on the standard Gaussian Mixture Model approach but permits an adaptive number of components per pixel. This generally produces good object silhouettes and runs very fast, but care has to be taken setting the learning rate to ensure that large objects are fully segmented, but equally that newly moving objects previously part of the background do not leave ghost detections.

To complement the change detector, a motion detector is employed. In this method, the three most recent frames are used $\{I(t), I(t-1), I(t-2)\}$ to determine the motion in the most recent frame $I(t)$. A set of corner features is detected in frame $I(t+1)$ using the method of [16]. These features are then tracked forwards to frame $I(t)$ and backwards to frame $I(t+2)$ using the sparse optical flow method of [10]. This results in two direction vectors for each feature, $[d_{0\to1}, d_{1\to2}]$. Features are clustered based on their motion with a constraint on the maximum distance between any two features. A triangulation of each cluster of features is performed such that the cluster can be rendered to a binary motion mask.

The two binary motion masks, from the change detector and the motion detector, are combined through a simple logical AND. Detections are the result of a connected components analysis of the fused binary motion mask

2.2. Image Plane Tracking

Tracking in the image plane is performed using two simple templates and a KLT feature tracker. The KLT feature tracker is used to track faster moving objects, while the template is used to optimise the location of the target and to retain track of objects for which there is no detection.

When the detector returns a detection, it can either be associated to an existing tracked target, or to a new target. When a new target is created, two small images are created. One is a grey-scale image of the size of the detection bounding box, while the other is an RGB image of the same size. The grey-scale image is the *detection mask template* D_t, and is initialised from the binary motion mask of the current image M_t, while the RGB image is the appearance template A_t and is initialised from the RGB pixel values of the current image I_t. Thus, on initialisation, if the top left corner of the detection bounding box is at image coordinates x, y:

$$D_t(u,v) = \begin{cases} 0 & \text{if } M_t(x+u, y+v) = 0 \\ 255 & \text{otherwise} \end{cases} \quad (1)$$

$$A_t(u,v) = I_t(x+u, y+v) \quad (2)$$

When a detection is associated to a new target, the detection and appearance templates are updated as a running average, given n as the learning rate:

$$D_t(u,v) = D_{t-1}(u,v) + \frac{M_t(x+u, y+v)}{n} \quad (3)$$

$$A_t(u,v) = D_{t-1}(u,v) + \frac{I_t(x+u, y+v)}{n} \quad (4)$$

Should the detection indicate a change in the width or height of the bounding box, the template images can be easily expanded or cropped as required.

At each frame, the appearance template can be compared to any location in the image by simple image difference, then the difference between the template and the image for a position (x, y) in the image is:

$$\Delta(I, A, x, y) = \sum_u \sum_v \frac{D(u,v)}{w} \delta(u,v) \quad (5)$$

where $\delta(u,v)$ is some difference (e.g. Euclidean) between the colour of the pixel $A(u,v)$ and the pixel $I(x+u, y+v)$.

The position of the template in the image is optimised using the simple Stochastic Diffusion Search [5], where a set of agents are created and distributed across the search space. The agents then evaluate their position, communicate, and reposition until the optimal location is deduced.This is simple and quite fast. Each tracked target maintains a set of KLT features that are tracked between frames. If the motion is large, the SDS search is started from the location predicted by the feature motion. Otherwise, the search is initiated from the previous location, reducing the effect of small motions caused by noise.

2.3. Multi-camera Fusion and 3D Tracking

The final stage of tracking is performed in the 3D coordinate system of the scene (though tracking itself is performed in 2D on the ground plane). Camera calibration is used with the ground plane constraint to back project the image position of each detected object to a position on the ground plane of the scene. This is followed by a Nearest Neighbour Data Associate Filter based fusion, and Kalman filter tracking, much as is described in [18]. In summary: for any given target tracked on the ground plane, its position frame-to-frame is predicted by a Kalman filter. A "validation gate" is used to limit the number of detections that can be associated to the tracked object based on the distance of the observation from the predicted position. The nearest observation from each camera is then used to update the Kalman filter.

3. Scene modelling

Modeling the spatial context of the scene is essential for recognition and interpretation of activity. By contextual areas we understand those semantic regions of the scene where people activities are expected to be different from one another. Contextual areas in the scene have thus a central role to understand activities as they allow analysing possible interactions between mobile and environmental objects of the scene and thus establish a semantic meaning. In our current application there are three key areas where servicing the plane takes place: The frontal and rear loading areas for baggage loading/unloading and the tanker area for the plane refueling. While the former two are well defined zones where vehicles must position themselves precisely, the latter is a broad zone where the tanker can freely stop for servicing. Similarly, other zones have been defined for the ground-personal in the airport to carry out specific operations. The main contextual zones, *Zctx*, are depicted in figure 1. The relevant scenario areas are: Entry/Exit areas and Parking/Servicing areas.

Figure 1. Left panel: Camera view of the Plane parked for servicing. Right panel: Apron top view with the main contextual zones manually defined. **(1).** GPU_Zone **(2).** Frontal unload area **(3).** Rear unload area **(4).** Frontal_Transporter_Zone **(5).** Rear_Transporter_Zone **(6).** Tanker_Zone

4. Trajectory analysis

The trajectory for object O_j is defined as the set of points [$x_j(t)$, $y_j(t)$] corresponding to their position points; x and y are time series vectors whose length is not equal for all objects as the time they spend in the scene is variable. Two key points defining these time series are the beginning and the end, [$x_j(1)$, $y_j(1)$] and [$x_j(end)$, $y_j(end)$] as they define where the object is coming from and where it is going to. We build a feature vector from these two points. Additionally, we also include the directional information given as [$cos(\theta_j)$, $sin(\theta_j)$], where θ_j is the angle which defines the vector joining [$x_j(1)$, $y_j(1)$] and [$x_j(end)$, $y_j(end)$]. A mobile object seen in the scene is thus represented by the feature vector

$$v_j = [x_j(1), y_j(1), x_j(end), y_j(end), cos(\theta), sin(\theta)] \quad (6)$$

This feature vector constitute a set of simple descriptors that have proven experimentally to be enough to describe activities in a large variety of domains, mainly because they are the most salient and reflect direction intention for semantic interpretation, but also because they are appropriate for real world videos depicting unstructured scenes where trajectories of different types have strong overlap.

In addition, so as to enable dynamic adaptation to newly observed data, we need a system able learn the activity clusters in an on-line way. On-line learning is indeed an important capability required to perform behaviour analysis on long-term basis. A first approach proposed in the state-of-the-art for on-line clustering is the Leader algorithm [7]. Given a distance *D* between any pair of objects, and a threshold *T*, the algorithm constructs a partition of the input space (defining a set of clusters) and a leading representative for each cluster, such that every object in a cluster is within a distance *T* of the leading object. The threshold *T* is thus a measure of the diameter of each cluster. For *i=1 to k*, the clusters CL_i, are numbered CL_1, CL_2 , CL_3, ..., CL_k. The leading object representative associated with cluster CL_i is denoted by L_i. The algorithm makes one pass trough the dataset, assigning each object to the cluster whose leader is the closest and making a new cluster, and a new leader, for objects that are not close to any existing leaders. However, the algorithm is extremely sensitive to the threshold parameter defining the minimum activation of a cluster *CL*. A new input object defined by its feature vector v_j will be allocated to cluster CL_i if v_j falls into its input receptive field (hyper-sphere whose radio is given by $\|r_i\|$=*T*). Defining *T* is application dependent. It can be supplied by an expert with a deep knowledge of the data or employing heuristics. In this work we propose to learn this parameter employing a training set and Machine learning.

Let each cluster CL_i be defined by a radial basis function (RBF) centered at the position given by its leader L_i:

$$CL_i(v) = \Phi\left(L_i, v, T\right) = \exp(-\|v - L_i\|^2 T^2) \quad (7)$$

RBF modelling allows for a straightforward way of on-line learning. The RBF function has a maximum of 1 when the difference between its leader L_i and the input *v* is 0 and thus acts as a similarity detector with decreasing values outputted whenever *v* strides away from L_i. We can make the choice that an object element will be included into a cluster if $CL_i(v) \geqslant 0.5$, which is a natural choice. The cluster receptive field (hyper-sphere) is controlled by the parameter *T*. Now, consider $C = \{CL_1 \cdots CL_k\}$ is a clustering structure of a data set $X = \{v_1, v_2, ..., v_N\}$; $\{L_1,\ldots,L_k\}$ are the leaders in this clustering structure and $P = \{P_1 \cdots P_s\}$ is a defined partition of the data and $\{M_1,\ldots,M_s\}$ are the main representatives (or Leaders) in the defined partition. We can define an error function given by

$$E = \frac{1}{N}\sum_{j=1}^{N} E_j \quad (8)$$

$$E_j = \hat{\Phi}\left(L(v_j), v_j, T\right) - \Phi\left(M(v_j), v_j, T\right) \quad (9)$$

$L(v_j)$ is the Leader associated to v_j in the clustering structure C. $M(v_j)$ is the Leader associated to v_j in the 'true' partition P. The error gives thus an indication of how many elements are misclassified according to the partition P. Minimising this error is equivalent to refine the clustering structure C or equivalently adjusting the parameter T controlling the cluster receptive field. A straightforward way to adjust T and minimise the error is employing an iterative gradient-decent method:

$$T(t+1) = T(t) - \eta\frac{\partial E(t)}{\partial T} \quad (10)$$

With the purpose of tuning parameter *T*, and for this application, we have defined a training dataset containing sixty nine synthetic trajectories. These trajectories were manually drawn on an empty scene and given semantic labels according to the end-user. Thus with this synthetic dataset we are able to tune the system to fulfill end-user requirements.

The proposed gradient-descent methodology was applied to the training dataset. The threshold *T*, in the leader algorithm, is initially set to a large value (which causes a merge of most trajectory types). After convergence, the threshold *T* has a value of T=0.7964, which is then selected for our analysis. The set of Leaders defined from this process will also guide the further partition of the incoming data. Remark that for this application we have not encountered local minima problems. However, as gradient-descent algorithms are clearly exposed to this problem, it could be envisaged to verify whether the minima found is indeed the global optima. A multiresolution analysis would be of help for this.

5. Activity analysis

5.1. Behaviour definition

We aim at creating a system for the recognition and interpretation of human activity and behaviour, and extract new information of interest for end-users. Low-level tracking information is thus expected to be transformed into high-level semantic descriptions conveying useful and novel information. In our application, we establish a semantic meaning from the scene model presented in section 3. The behaviour knowledge can be thus expressed with semantic concepts, instead of using quantitative data, thanks to the defined contextual zones. Let us assume we have defined *p* contextual zones on the scene model. Two different kinds of behaviours can then be identified:

• From Zone $Zctx_q$ to Zone $Zctx_{q'}$
• At Zone $Zctx_q$

In order to cope with the uncertainty aspects, contextual zones are modelled as elliptical shapes with a Gaussian probability density function being associated. Each ellipse, $\varepsilon(a,b,c)$, is thus defined by its major and minor axis a, b respectively and its centre c. The membership degree that a point p (x, y) can have to a defined zone, $Zctx_q$, is then given by

$$Zctx_q(x',y') = \exp\left(-\left(\frac{x'}{\sigma_{a(Zctx_q)}}\right)^2\right)\exp\left(-\left(\frac{y'}{\sigma_{b(Zctx_q)}}\right)^2\right) \tag{11}$$

where (*x*', *y*') is the image point *p*' after projection of *p* into the major and minor axes which define the elliptical zone. That is $p' = A(p-c)$ and A is the rotation matrix defined by the major and minor axis of the ellipse.

The likelihood that the entry/exit points belonging to a trajectory cluster CL_i can be associated with the semantic given by a zone $Zctx_q$ is the mean value of the membership degree of these points to that zone.

5.2. Scene model update

Because it is not possible to define a-priori all activity zones, the manually defined Contextual zones do not suffice to describe all possible situations or evolving actions in the monitored scene, but only those matching the previously modelled zones of interest. We thus learn the complementary activity zones from the results obtained on trajectory clustering. We employ the RBF entry/exit (beginning/end) spatial zone of influence Zcl_i of a trajectory cluster CL_i.

$$Zcl_i(x,y) = \Phi\left(L_i(1),x,T\right)\Phi\left(L_i(2),y,T\right) \tag{12}$$

Remark that in this case employing $L_i(1)$ and $L_i(2)$ means $Zcl_i(x,y)$ is built from the entry points of trajectory cluster CL_i. We then look two establish a similarity relation between the different zones defined by the clusters. On the end, new zones are given by the fulfillment of two relations: cluster CL_i influential zone Zcl_i is similar to cluster CL_j influential zone Zcl_j and cluster CL_j influential zone Zcl_j does not overlap an a-priori defined contextual Zone $Zctx_q$. These relations are defined:

$R1_{ij}$: Zone Zcl_i is similar to Zone Zcl_j

$$R1_{ij} = \sum_{k=1}^{3}\left[\sum_{(x,y)\in(X_{ik},Y_{ik})} Zcl_j(x,y)\right] \tag{13}$$

and $X_{ik} = \left\{\frac{(k+1)}{3}T\cos(\theta) + L_i(1)\right\}$,
$Y_{ik} = \left\{\frac{(k+1)}{3}T\sin(\theta) + L_i(2)\right\}$ with $\theta = 0, ..., \frac{\pi}{8}, ..., 2\pi$

That is, points belonging to concentric circles to L_i are employed for the similarity comparison between CL_i and CL_j. This allows avoiding equity problems with clusters defined on sparse regions (some clusters may be defined with a much larger number of points than others).

$R2_{iq}$: Zone Zcl_i overlaps Zone $Zctx_q$

$$R2_{iq} = \sum_{k=1}^{3}\left[\sum_{(x,y)\in(X_{ik},Y_{ik})} Zctx_q(x,y)\right] \tag{14}$$

It is possible to transform R2 into a new relation, R3, which links CL_i and CL_j if both clusters are related to the same Zone $Zctx_q$ through the fulfillment of R2. The relation between CL_i and CL_j is then given by

$$R3_{ij} = \max_q \min\left[R2_{iq}, R2_{qj}\right] \tag{15}$$

Remark that $\overline{R3}$, the complement to $R3$ given by $\overline{R3} = -R3$, represents the relation linking CL_i and CL_j if both clusters are not related to any contextual Zone ($Zctx_q$). $R1$ and $\overline{R3}$ can be aggregated employing a soft computing aggregation operator such as $R = R1 \cap \overline{R3} = \max\left(0, R1 + \overline{R3} - 1\right)$ and made transitive with:

$$R \circ R\left(x,y\right) = \max_z \min\left(R\left(x,z\right), R\left(z,y\right)\right) \tag{16}$$

R is now a transitive similarity relation with R indicating the strength of the similarity. If we define a discrimination level α in the closed interval [0,1], an $\alpha - cut$ can be defined such that

$$R^\alpha\left(x,y\right) = 1 \Leftrightarrow R\left(x,y\right) \geqslant \alpha \tag{17}$$

It is thus implicit that $\alpha_1 > \alpha_2 \Leftrightarrow R^{\alpha_1} \subset R^{\alpha_2}$; thus, the R^α form a nested sequence of equivalence relations, or from the classification point of view, R^α induces a partition $\pi^\alpha = \{Z_i^\alpha\}$ of $X \times Y$ (or X^2) such that $\alpha_1 > \alpha_2$ implies π^{α_1} is a refinement of π^{α_2}.

At this point, the difficulty comes down to select the appropriate $\alpha - cut$ such that π^α from R^α represents the best partition of the data. This is still a difficult and open issue that we choose to approach by selecting the alpha-values, which induce a significant change from π^{α_k} to $\pi^{\alpha_{k+1}}$.

To automatically detect those significant partition changes we choose to study the cluster area and number of clusters induced at each partition π^α. We achieve this in the frame of a multiresolution analysis. By analysing induced partitions at coarse resolutions, it is possible to smooth out small details and select the $\alpha - cut$ levels associated with important changes. From the monitored scene, it would be useful to distinguish among different information levels: (i) grouped activity on large spaces, (ii) very detailed individual activity, (iii) somewhere meaningful in-between the last

two. For this reason, when performing activity zone discovery, we automatically select the three highest change-inducers $\alpha - cut$ levels from the previous analysis. The result is then that we end up with a three levels hierarchy of activity zones.

5.3. Semantic update

It is important to observe that as the system stands no particular semantic information can be drawn for the discovered activity zones. To solve this problem, we rely again on the semantic that can be deduced from the contextual areas of the scene, as we know that this is the link to establish possible interactions between mobile and environmental objects of the scene. To this end we consider two new relations: R4, The comparison of areas between discovered and contextual areas, and R5, The distance relationships between discovered and contextual areas:

$R4_{iq}$ = Zone Zcl_i is similar in area to Zone $Zctx_q$
$R5_{iq}$ = Zone Zcl_i is near to Zone $Zctx_q$

$$R = R4 \cap R5 = \max(0, R4 + R5 - 1) \qquad (18)$$

From R, we know for each discovered zone what is the 'best' contextual zone to refer to. The zone areas are calculated from the convex hull enveloping either Z_i^α or $Zctx_q$. The distance between zones is that between the nearest vertex points of each convex hull.

6. Results

The algorithm can be applied to any given period monitoring the servicing of an aircraft in the airport docking area. In order to evaluate whether the zone model update works properly and to asses the correctness of the new learned zones, we took out from the a-priori knowledge, most contextual zones defined in section 5.2 and left only the 'Frontal unload area' and 'Rear unload area'. We then took one video sequence with available Ground-truth annotation and containing the most relevant activity events of the sequence. These are: 'GPU positioning', 'Handler deposits chocks', 'Frontal unloading operation', 'Frontal loading operation', 'Rear loading operation', 'Push back vehicle positioning'. We can thus infer that those essential areas to be learned for activity reporting are the GPU, Push back, and loading/unloading related areas.

The algorithm for activity extraction and scene model update is then applied according to the fulfillment of relations $R1$ and $\overline{R3}$ given in section 5.2. The final relation R, which verifies the transitive closure, is thresholded for different $\alpha - cut$ values going from 0 to 0.9 and with a step value of 0.05. The $\alpha - cut$ values defining the different granularities (or information levels) for the scene are then obtained from the multiresolution analysis of the mean cluster

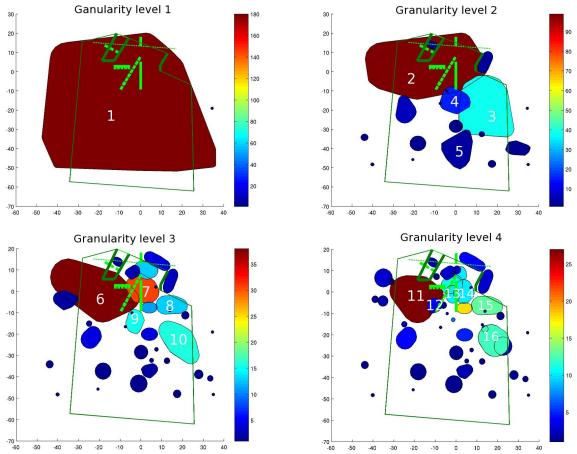

Figure 2. Activity maps at different granularity resolutions. Numbers in zones indicate the most frequently employed discovered zones: **(1).** ERA and large surroundings **(2).** just north-west of and inside ERA **(3).** just south-east of and inside ERA **(4).** 4 meters away south-west of Frontal unload area **(5).** 6 meters away south of Rear unload area **(6).** just west of and inside ERA **(7).** just north of Frontal unload area **(8).** just south of Tow tractor parking area **(9).** just north-west of Rear unload area **(10).** 11 meters away east of Rear unload area **(11).** just west of and inside ERA **(12).** 9 meters away south-west of Cabin access area **(13).** just north-west of Frontal unload area **(14).** just west of Tow tractor parking area **(15).** just south of Tow tractor parking area **(16).** 6 meters away east of Rear unload area

area, range value of cluster areas, and number of clusters, which are obtained at each $\alpha - cut$ level. The algorithm calculates automatically the three highest change-inducers $\alpha - cut$ levels, which will define three information levels for scene activity reporting.

Figure 2 shows the π^α partitions corresponding to the selected $\alpha - cut$ levels. The first granularity level is set for $\alpha - cut$=0, which merges all activity outside the user-defined contextual zones and thus creates one single broad new zone of global activity outside contextual zones. The second granularity level corresponds to grouped activity on large spaces and is defined as the lowest $\alpha - cut$ value from those three highest change-inducers $\alpha - cut$ levels. In contrast, the fourth granularity level, which is the most detailed activity corresponds thus to the partition induced by the highest change-inducer $\alpha - cut$ level. The third granularity level, corresponds to a compromise between detailed and large activity description (and is defined by the remaining $\alpha - cut$ level). In this way, the different partitions can be seen as activity maps with different granularity levels. The new discovered zones most employed at each granularity level are enumerated on the figure.

We calculated the overlap between the learned zones and those manually defined. This overlap can be observed in figure 3, while the quantitative result of the comparison is given in table 1. From the obtained results, there is

a fair amount of contextual zones (8/11) having an overlap of at least 30% with a learned zone calculated from the system, which could actually be considered as True Positives from the recognition point of view. There are three zones with a relative low overlap value of 16% and below. These contextual zones could be thought as False Negatives given by the system. However, several factors are to be considered. First, it must be said that the activity events contained by the Ground-truth do not point to any information allowing to deduce that the Tanker_Zone and Rear_Bulk_Transporter_Zone are actually employed and containing significant activity for the processed video sequence so it is hard to say whether these zones represent indeed False Negatives. Secondly, what regards the Tanker_Zone, this is a broad area where the Tanker vehicle is allowed to stop, however it is unlikely that the refueling activity will spread over all such defined Tanker_Zone, and thus discovering an activity zone at this resolution level with that extent is rather difficult. Lastly, it must be said that the Ground-truth does not provide any information regarding what are the mobiles involved on the activity event, nor on their spatial position. Moreover, all vehicles may not stop always on the same position. For this video sequence, the activity related to the Rear Transporter is simply shifted by some meters on the east direction. Our system actually helps not only to discover completely new activity zones but also to redefine those zones which can be changing dynamically from one operation sequence to another.

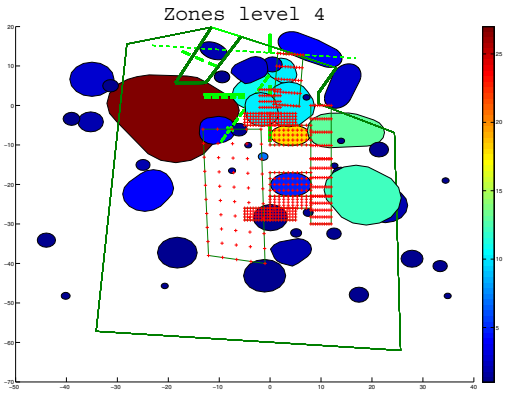

Figure 3. Level 4 activity map for the processed sequence. Red '+' markers represent manually defined contextual zones

As mentioned in section 5.3, it is important to attach a semantic meaning to each of the new discovered zones. This is achieved, as mentioned before, through fulfillment of relations $R4$ and $R5$ linking the new discovered zones to the user defined contextual zones by their area similarity and their spatial closeness. For instance, for the enumerated zones in figure 2, the deduced semantics are given in the figure legend. The whole activity observed from the scene can then be reported following the behaviour definition given in section 5.1 and moreover at the different granularities calculated by the system. For instance, for the rear loading/unloading activity, the obtained report is given in table 2.

Reference Zone	Recognition percentage
Left_Frontal_PassangerDoor_Zone	56%
Left_Rear_PassangerDoor_Zone	76%
Right_Rear_PassangerDoor_Zone	77%
Rear_BackLoading_Zone	41%
GPU_Zone *	35%
Frontal_PassangerDoor_Zone	52%
Frontal_Transporter_Zone *	32%
PushBack_Zone	49%
Tanker_Zone	11%
Rear_Transporter_Zone *	16%
Rear_Bulk_Transporter_Zone	13%

Table 1. Recognition result for contextual areas expected to be seen in the scene model. Those marked with '*' are directly related with the 'GPU positioning' and 'loading/unloading' events.

Proportion	# Mobiles	Description
		level 4
5%	9	10 meters away east of Rear unload area
3%	5	at Rear unload area
2%	3	10 meters away east of Rear unload area to just south of Tow tractor parking area
		level 3
8%	14	11 meters away east of Rear unload area
2%	3	at Rear unload area
1%	2	11 meters away east of Rear unload area to just south of Tow tractor parking area
1%	1	5 meters away west of Frontal unload area to 11 meters away east of Rear unload area
		level 2
20%	37	just south-east of and inside ERA
1%	1	just south-east of and inside ERA to just north-west of Rear unload area
1%	1	Tow tractor parking area to just south-east of and inside ERA
		level 1
100%	183	ERA and large surroundings

Table 2. Activity reporting (not exhaustive) with proportion percentage, number of mobiles and semantical description related to rear loading areas.

7. Conclusions and Future work

In this paper, we have described an artificial cognitive vision system for activity extraction and reporting in a visual surveillance/monitoring task. The system presented works in unsupervised manner from detected mobile trajectories principally in two steps. First, similar trajectories are grouped employing a clustering algorithm. We employ a simple, yet advantageous incremental algorithm able to create new clusters if necessary with new oncoming data without needing to reprocess previous data. We have tuned the algorithm by learning the coefficients indicating how flexible the cluster can be updated with new data. We are thus

able to perform analysis on long-term basis. In a second step, the spatial information obtained from trajectory clusters regarding main points of entry and exit from mobile objects is modeled employing fuzzy relations. By applying specific relation aggregation operators, we are able to deduce the different areas of activity in the scene. These can either be new learned zones or a refinement of existing contextual zones previously defined. This topological scene model update allows to infer the behaviour of the observed mobile objects in the scene. Such behaviour is given in a close to a natural language reporting form. Moreover, we study the scene activity at different granularities which give the activity description in broad terms, or with detailed information thus managing different information levels. Delivering such reports with activity maps, figures and numbers and a semantic description of the ongoing behaviours is an essential step for modern cognitive vision system aiming at automatic activity interpretation. Our current results show to be consistent in terms of zone discovery and semantic information delivery with apriori annotated information available for the studied application. The system, however, is still lacking some key descriptive information such as the mobile object type (e.g. person, vehicle) and needs also to exploit the temporal information also contained within the mobile object trajectories. These two aspects constitute our future work to enhance our system and deliver more complex activity descriptions.

Acknowledgements

This work was partially funded by the EU FP7 project CO-FRIEND with grant no. 214975.

References

[1] E. L. Andrade, S. Blunsden, and R. B. Fisher. Hidden markov models for optical flow analysis in crowds. *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, 1, 2006. 2

[2] N. Anjum and A. Cavallaro. Single camera calibration for trajectory-based behavior analysis. In *2007 IEEE Conference on Advanced Video and Signal Based Surveillance, AVSS 2007*, pages 147–152. IEEE, 2007. 2

[3] G. Antonini and J. Thiran. Counting Pedestrians in Video Sequences Using Trajectory Clustering. *IEEE Transactions on Circuits and Systems for Video Technology*, 16:1008–1020, 2006. 2

[4] F. Bashir, A. Khokhar, and D. Schonfeld. Object trajectory-based activity classification and recognition using hidden markov models. *Image Processing, IEEE Transactions on*, 16(7):1912 –1919, 2007. 2

[5] J. Bishop. Stochastic searching networks. In *Proceedings of IEE Conference on Artificial Neural Networks*, pages 329–331, 1989. 3

[6] G. Foresti, C. Micheloni, and L. Snidaro. Event classification for automatic visual-based surveillance of parking lots. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 314–317. IEEE, 2004. 2

[7] J. A. Hartigan. *Clustering algorithms*. John Wiley & Sons, Inc., New York, 1975. 4

[8] S. Hongeng, R. Nevatia, and F. Bremond. Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, 96(2):129 – 162, 2004. Special Issue on Event Detection in Video. 2

[9] I. Laptev. Improvements of object detection using boosted histograms. In *British Machine Vision Conference*, volume 3, pages 949–958, 2006. 2

[10] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981. 2

[11] F. Lv, X. Song, B. Wu, V. Singh, and R. Nevatia. Left luggage detection using bayesian inference. *Proceedings of the 9th IEEE International Workshop on*, 2006. 2

[12] D. Makris and T. Ellis. Automatic learning of an activity-based semantic scene model. In *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2003.*, pages 183 – 188, 21-22 2003. 2

[13] C. Piciarelli, G. Foresti, and L. Snidaro. Trajectory clustering and its applications for video surveillance. In *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2005.*, pages 40–45. Ieee, 2005. 2

[14] C. Piciarelli, C. Micheloni, and G. Foresti. Trajectory-based anomalous event detection. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1544 –1554, 2008. 2

[15] F. Porikli. Learning object trajectory patterns by spectral clustering. In *2004 IEEE International Conference on Multimedia and Expo (ICME)*, volume 2, pages 1171–1174. IEEE, 2004. 2

[16] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 593 – 600, 1994. 2

[17] C. Stauffer. Estimating tracking sources and sinks. In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW '03. Conference on*, volume 4, pages 35 –35, 16-22 2003. 2

[18] D. Thirde, M. Borg, J. Aguilera, H. Wildenauer, J. Ferryman, and M. Kampel. Robust real-time tracking for visual surveillance. *EURASIP Journal on Advances in Signal Processing*, 2007, 2007. 3

[19] T. Xiang and S. Gong. Video behaviour profiling and abnormality detection without manual labelling. *Tenth IEEE International Conference on Computer Vision, 2005. ICCV 2005*, 2, 2005. 2

[20] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Semi-supervised adapted hmms for unusual event detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, 1, 2005. 2

[21] Z. Zivkovic. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 2006. 2