Unsupervised Activity Extraction on Long-Term Video Recordings Employing Soft Computing Relations

Luis Patino[1], Murray Evans[2], James Ferryman[2], François Bremond[1], and Monique Thonnat[1]

[1] INRIA Sophia Antipolis - Méditerranée
2004 route des Lucioles - BP 93 - 06902 Sophia Antipolis, France
[2] School of Systems Engineering, University of Reading
RG6 6AY United Kingdom

Abstract. In this work we present a novel approach for activity extraction and knowledge discovery from video employing fuzzy relations. Spatial and temporal properties from detected mobile objects are modeled with 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 finding spatio-temporal patterns of activity. We present results obtained on videos corresponding to different sequences of apron monitoring in the Toulouse airport in France.

Keywords: Video data mining, vision understanding, discovery in multimedia database, soft computing.

1 Introduction

Scene understanding corresponds to the real time process of perceiving, analysing and elaborating an interpretation of a 3D dynamic scene observed through a network of sensors (including cameras and microphones). This process consists mainly in matching signal information coming from sensors observing the scene with a large variety of models which humans are using to understand the scene. Although activity models can be built by experts of the domain, this might be a hard and time-consuming task depending on the application and the spectrum of activities that may be observed. The challenge thus consists of discovering, in an unsupervised manner, the significant activities observed from a video sequence. Knowledge discovery systems (KDS) aim at helping the human operator on this aspect. KDS systems have become a central part on many domains where data is stored in a database, but little research has been only done in the field of video data-mining. It must be said the task is particularly challenging because of the difficulty in identifying the interesting patterns of activity in the video due to noise, incomplete or uncertain information inherently present in the data. Soft computing methodologies are particularly suitable for these tasks because

J.L. Crowley, B. Draper, and M. Thonnat (Eds.): ICVS 2011, LNCS 6962, pp. 91–100, 2011.

they provide the capability to process uncertain or vague information, as well as a more natural framework to cope with linguistic terms and produce natural language-like interpretable results. Fuzzy sets are the corner stone of soft computing together with other techniques such as neural networks and genetic algorithms. The relation between different existing fuzzy sets can be graded with the use of fuzzy relations [15]. Various fuzzy-based soft computing systems have been developed for different applied fields of data mining; but only a few systems employ soft computing techniques to partially characterize video activity patterns [3,6]; In this paper we present a fully unsupervised system exploiting the use of fuzzy relations for the discovery of activities from video. First we model spatial and temporal properties from detected mobile objects employing fuzzy relations. We employ typical soft-computing algebra to aggregate these relations. A clustering algorithm based on the transitive closure calculation of the fuzzy relations allows finding spatio-temporal patterns of activity with different granularities. We have applied the proposed technique to different video sequences of apron monitoring in the Toulouse airport in France.

The reminder of this paper is as follows. In the next section, we give a short overview of the related work. We give a general overview of the system architecture and present our global approach in section 3 . The object detection and tracking process is given in section 4, then the data preprocessing steps previous to activity extraction are explained (section 5). We give the activity clustering methodology in section 6. Section 7 gives the main results and evaluation. Finally, Section 8 draws the main conclusions and describes our future work.

2 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. A few systems employ soft computing techniques to characterize video activity patterns [3,6] but the methodology to self-discover new activities is still missing. Because of the complexity to tune parameters or to acquire knowledge, most systems limit themselves to object recognition . 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 [5], Bayesian classifiers and HMMs [8,13].
- The second class consists in using unsupervised learning techniques to deduce abnormalities from the occurring events [14].
- 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 detecting normal/abnormal behaviours; for instance, on abnormal trajectory detection on roads [9,12] or pedestrian trajectory characterisation [1]. Hidden

Markov Models (HMM) have also been employed to detect different states of predefined normal behaviour [2,10]. All these techniques are interesting, but little has been said about the semantic interpretability of the results. Indeed, more than trajectory characterisation, we are interested in extracting meaningful activity, where different trajectory types may be involved. This work comes thus into the frame of behaviour extraction from trajectory analysis however we have in addition a higher semantic level that employs spatial and temporal proximity relations between detected mobiles to characterise the ongoing different activities of the scene. In a similar framework, Dubba et al [4] have researched into transforming tracking data into a relational form where the relations are spatio-temporal relations among objects. This is the most comparable work to us, not only because they also look to identify activities as groups with coherent spatio-temporal information, but because they have worked on the same dataset. However, Dubba et al. employ a supervised approach (based on Inductive Logic Programming) and thus requires large quantities of annotated data. Our proposed approach is on the contrary completely unsupervised.

3 General Overview of the System

Our proposed system is mainly composed of two different processing components. The first one is for the detection and tracking of objects. The second subsystem works off-line and achieves the extraction of activity patterns from the video. This subsystem is composed of two modules: The trajectory speed analysis module, and the activity analysis module. The first is aimed at segmenting the trajectory into tracklets of fairly similar speed. The latter is aimed at extracting complex patterns of activity, which include spatial information (coming from the trajectory analysis) and temporal information related to the interactions of mobiles observed in the scene.

Streams of video are acquired at a speed of 10 frames per second. The on-line (real time) analysis subsystem takes its input directly from the data acquisition component; the video is stored in the DB parallel to the real time processing.

4 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.

4.1 Detection

Detection is performed by combining change detection and motion detection. The first detector is the Adaptive Gaussian Mixture Model of Zivkovic [16]. 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.

To complement the change detector, a motion detector is employed. In this method, the three most recent frames $\{I(t), I(t-1), I(t-2)\}$ are used 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 in [11]. These features are then tracked forwards to frame $I(t)$ and backwards to frame $I(t+2)$ using the sparse optical flow method in [7]. This results in two direction vectors for each feature, $[d_{0\rightarrow 1}, d_{1\rightarrow 2}]$. 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.

4.2 Image Plane Tracking

Tracking in the image plane is performed using two simple templates and a KLT feature tracker. 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 greyscale image of the size of the detection bounding box, while the other is an RGB image of the same size. The greyscale 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:

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

$$A_t(u,v) = I_t(x+u, y+v) \tag{2}$$

When a detection is associated to a new target, the detection and appearance templates are updated as a running average. 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.

Each tracked target maintains a set of KLT features that are tracked between frames. The overall plane tracking method is generally good enough to reliably maintain a track on large objects such as vehicles, which often stop for extended periods in the scene. It is not intended to track objects through occlusions, but rather to detect the presence of objects, and maintain the presence of static objects.

4.3 Multi-camera Fusion and 3D Tracking

The final stage of tracking is performed in the 3D coordinate system of the scene (though tracking remains 2D on the ground plane). Camera calibration is used to project the bounding boxes of per-camera tracking targets to each other camera view as four epipolar lines from the four corners of the bounding box. This provides a mechanism for rating the extent to which tracking targets are related between views, by determining the extent to which a bounding box fits between the extremal epipolar lines of a bounding box from another view. Agglomerative clustering is used to determine possible solutions for the correct fusion of targets, and an optimisation process then determines the optimal clustering for a single frame of video. Optimal solutions are retained over a temporal window, and an overall optimal association of per-camera targets to fused targets is determined, and fused tracking targets updated for every new frame.

5 Data Preprocessing

In order to discover meaningful activity clusters, it is of prime importance to have available detailed information allowing to detect the different possible interactions between mobiles. As our system is based on trajectory analysis, the first step to prepare the data for the activity clustering methodology is to extract tracklets of fairly constant speed allowing to characterise the displacements of the mobile or its stationary state.

If the dataset is made up of N objects, the trajectory tr_j for object O_j in this dataset 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. The instantaneous speed for that mobile at point $[x_j(t), y_j(t)]$ is then $v(t) = \left(\dot{x}(t)^2 + \dot{y}(t)^2\right)^{\frac{1}{2}}$. The objective is then to detect those points of changing speed allowing to segment the trajectory into tracklets of fairly constant speed so that the trajectory can be summarised as a series of displacements at constant speed or in stationary state.

The mobile object time series speed vector is analysed in the frame of a multiresolution analysis of a time series function $v(k)$ with a smoothing function, $\rho_{2^s}(k) = \rho(2^s k)$, to be dilated at different scales s. In this frame, the approximation A of $v(k)$ by ρ is such that $A_{2^{s-1}}v$ is a broader approximation of $A_{2^s}v$. By analyzing the time series v at coarse resolutions, it is possible to smooth out small details and select those points associated with important changes.

The speed change points are then employed to segment the original trajectory tr_j into a series of i tracklets tk. Each tracklet is defined by two key points, these are the beginning and the end of the tracklet, $[x^i_j(1), y^i_j(1)]$ and $[x^i_j(end), y^i_j(end)]$ as they define where the object is coming from and where it is going to and also with approximative constant speed. We build a feature vector from these two points. By globally reindexing all tracklets, let m be the number of total tracklets extracted, we obtain the following tracklet feature vector :

$$tk_m = [x_m(1), y_m(1), x_m(end), y_m(end)] \tag{3}$$

6 Activity Clustering Methodology

We understand activity as the interactions occurring between mobile objects themselves and those between mobiles and the environment. We propose in this work to model those interactions employing Soft computing techniques. The motivation is that they provide uncertain information processing capability; set a framework to work with symbolic/linguistic terms and thus allows producing natural language-like interpretable results.

6.1 Preliminary Definitions

A fuzzy set is a set of ordered pairs such as $A = \{(x, \mu_A(x)) \mid x \epsilon X\}$ and the belonging of x to A is given by μ_A. Any relation between two sets X and Y is known as a binary relation R:

$$R = \{((x, y), \mu_R(x, y)) \,|\, (x, y) \,\epsilon X \times Y\}$$

and the strength of the relation is given by $\mu_R(x, y)$. Let's consider now two different binary relations, *R1* and *R2*, linking three different fuzzy sets X, Y, and Z : R1 = x is relevant to y; R2 = y is relevant to z.

It is then possible to find to which measure x is relevant to z (noted *R=R1oR2*) by employing the extension principle:

$$\mu_{R=R1\circ R2}(x, z) = \max_y \min [\mu_{R1}(x, y), \mu_{R2}(y, z)]$$

It is interesting to verify whether the resulting relation is symmetric, $R(x, y) = R(y, x)$, reflexive $R(x, x) = 1$, which make of R a compatibility relation and occurs in most cases when establishing a relationship between binary sets. Because R was calculated employing the extension principle, R is also a transitive relation. $R(x, y)$ is a transitive relation if $\exists\, z \in X, z \in Y/R(x, y) \geqslant \max_z \min [R(x, z), R(z, y)]$. R can be made furthermore closure transitive following the next steps

Step1. $R' = R \cup (R \circ R)$
Step2. If $R' \neq R$, make$R = R'$ and go to step1
Step3. $R = R'$ Stop. R is the transitive closure where

$$R \circ R(x, y) = \max_z \min (R(x, z), R(z, y)) \tag{4}$$

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

$$R^\alpha(x, y) = 1 \Leftrightarrow R(x, y) \geqslant \alpha; R = \underset{\alpha}{\cup} \alpha R^\alpha \tag{5}$$

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 π^α of $X \times Y$ (or X^2) such that $\alpha_1 > \alpha_2$ implies π^{α_1} is a partition refinement of π^{α_2}.

6.2 Clustering of Video Data

We now set out to establish the appropriate relations between detected mobiles in the video reflecting spatio-temporal similarities in order to obtain activity patterns. With this aim, we define the following relations:

$R1_{ij}$: mobile object *O(i)* meets mobile object *O(j)*. In this case the action 'meets' must be understood spatially and thus gives a degree of spatial closeness between the two mobiles.

$$R1_{ij} = min\left(\left\|tk_i(1), tk_j(1)\right\|, \left\|tk_i(1), tk_j(2)\right\|\right), \\ \left(\left\|tk_i(2), tk_j(1)\right\|, \left\|tk_i(2), tk_j(2)\right\|\right) \quad (6)$$

$R2_{ij}$: mobile object *O(i)* starts equal to mobile object *O(j)*. Here we are attempting to relate mobile objects that share temporal closeness.

$$R2_{ij} = 1 - abs(start\_time(i) - start\_time(j)) \quad (7)$$

$R3_{ij}$: mobile object *O(i)* starts after mobile object *O(j)*. Here we are attempting to relate mobile objects that appear in a sequential manner.

$$R3_{ij} = 1 - abs(start\_time(i) - end\_time(j)) \quad (8)$$

Obtaining the patterns of activity is achieved by aggregating the above spatio-temporal relations with a typical T-norm operator.

$R = R1 \cup R2 \cup R3$ aggregates temporal similarity relations between mobiles. We calculate the transitive closure of this new relation. Analogically to section 6.1 an $\alpha - cut$ can be defined such that $R^{\alpha}(x, y) = 1 \Leftrightarrow R(x, y) \geqslant \alpha$ and R^{α} induces a new partition $\pi^{\alpha} = \{C_1^{\alpha}, ..., C_i^{\alpha}, ..., C_{n^{\alpha}}^{\alpha}\}$; each C_i^{α} represents a discovered spatio-temporal activity pattern.

7 Results and Evaluation

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 activity extraction algorithm works properly and to asses the correctness of the results, we took five video datasets (each lasting about 1 hour) with available Ground-truth annotation and containing the start and end time for the most relevant activity events of the sequence.

The procedure to find the activity clusters is applied as given in section 6.2. In this work, the final relation R, which verifies the transitive closure, is thresholded for different $\alpha - cut$ values going from 0.05 to 0.95 and with a step value of 0.10. Low $\alpha - cut$ values produce only a few number of clusters of broad resolution as most of the activity is merged spatially and temporally. $\alpha - cut$ values near to one produce activity clusters of higher resolution with more precisely defined activities. At each resolution it is possible to calculate the temporal overlap

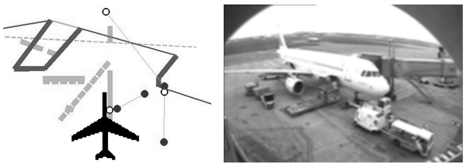

Fig. 1. Example of an activity cluster obtained. The left panel presents the tracklets of the mobiles participating in the Frontal Loading activity. Filled circles indicate the beginning of a tracklet. Empty circles indicate the end of a tracklet. The right panel presents the start frame of the activity.

Table 1. Percentage of overlap between discovered activities and the reference events contained in the ground-truth. The symbol * indicates that for all video sequences, the ground-truth event matches a discovered activity obtained for an alpha value of 0.95. NA indicates 'does not apply' (does not appear in the video sequence).

	video sequence				
Reference event	1	2	3	4	5
GPU vehicle arrival*	96%	0%	68%	90%	91%
Handler deposits chocks*	57%	81%	65%	60%	67%
Aircraft arrival*	90%	91%	71%	81%	66%
Jet Bridge positionning	56%	42%	70%	58%	70%
Frontal loading operation 1	NA	NA	NA	25%	32%
Frontal loading operation 2	NA	NA	NA	94%	NA
Frontal loading operation 3	NA	NA	NA	67%	NA
Frontal loading operation 4	NA	NA	NA	73%	NA
Rear loading operation 1	43%	52%	43%	82%	50%
Rear loading operation 2	53%	41%	38%	NA	63%
Rear loading operation 3	93%	NA	NA	NA	25%
Push-back positioning*	0%	69%	0%	89%	96%
Aircraft departure*	89%	94%	63%	81%	75%

between the extracted activity clusters and the ground-truth clusters. The quantitative result of this comparison is given in table 1. For each video sequence and for each ground-truth event only the best overlap across all $\alpha - cut$ values is reported. Remark that specific activities involving one mobile require precise definition obtained with activity clusters of higher resolution while loading operations involving the interaction of several mobiles are defined with mid-resolution activity clusters ($\alpha - cut$ values of 0.75 or 0.85). For instance, figure 1 presents a frontal loading activity cluster obtained for an $\alpha - cut$ value of 0.75. In general, all events are recognized correctly in all video sequences. When the percentage of overlap decreases or even goes to zero, it is mainly due to low-level object occlusion problems, which do not allow extracting all mobile trajectories and disturbs then the analysis of all possible mobile interactions.

Our results can partially be compared to those obtained with a supervised approach to learn apron activity models with Inductive Logic Programming (Dubba et al. [4]). As previously indicated, Dubba et al. have worked on the same apron monitoring video dataset from the Toulouse airport in France. Dubba et al. have concentrated on supervised learning of four apron activities: Aircraft arrival;

Table 2. Results comparison between our results and those presented in Dubba et al. at ECAI 2010 [4]

	Dubba et al.	Our Approach	
Reference Event	TPR	TPR	Mean temporal overlap with GT
Rear Loading / Unloading	80 %	100 %	53 %
Aircraft arrival	100 %	100 %	80 %
Aircraft departure	57 %	100 %	80 %
Jet Bridge positionning	57 %	100 %	59 %
Frontal Loading / Unloading	--	100 %	58 %
GPU vehicle arrival	--	80 %	86 %
Handler deposits chocks	--	100 %	66 %
Push-back positioning	--	60 %	85 %

Aircraft departure; Rear Loading/unloading; Jet Bridge Positionning. Dubba et al. obtained a global True Positive Rate (TPR) of 74%. In our work (from Table 1), we have 80% global True Positive Rate for the recognition of eight apron activities: Aircraft arrival; Aircraft departure; Rear Loading/unloading; Jet Bridge Positionning; Frontal Loading/Unloading; GPU vehicle arrival; Handler deposits chocks; Push-back positioning. Dubba et al. approach works as a hit or miss recognition system and in their paper there is no information on what is the temporal overlap between the recognised activities and the ground-truth activities. In our case such temporal overlap has a global value of 73%. The event-by-event comparison between the two approaches is detailed in table 2.

8 Conclusions

We have presented in this paper, a novel approach to extract activity patterns from video. The technique is unsupervised and is based on the use of fuzzy relations to model Spatial and temporal properties from detected mobile objects. Fuzzy relations are aggregated employing typical soft-computing algebra. A clustering algorithm based on the transitive closure calculation of the fuzzy relations allows finding spatio-temporal patterns of activity. Our current results are encouraging as the final patterns of activity are given with coherent spatial and temporal information, which is understandable for the end-user. When comparing our results with explicit ground-truth given by a domain expert, we were able to identify the events in general with a temporal overlap of at least, or near, 50%. Events with small temporal overlap in some video sequences is because of low-level detection problems. The comparison with a supervised method on the same data indicates that our approach is able to extract the interesting activities signalled in the ground-truth with a higher True Positive Rate. More importantly, our approach is completely unsupervised. In our future work we will try to work on improving our technique to determine the meaningfulness (or abnormality) of single activity patterns. We also plan to work on the semantical description of the activity clusters.

References

1. Anjum, N., Cavallaro, A.: Single camera calibration for trajectory-based behavior analysis. In: AVSS 2007, IEEE Conference on Advanced Video and Signal Based Surveillance, pp. 147–152 (2007)
2. Bashir, F., Khokhar, A., Schonfeld, D.: Object Trajectory-Based Activity Classification and Recognition Using Hidden Markov Models. IEEE Transactions on Image Processing 16, 1912–1919 (2007)
3. Doulamis, A.: A fuzzy video content representation for video summarization and content-based retrieval. Signal Processing 80(6), 1049–1067 (2000)
4. Dubba, K.S.R., Cohn, A.G., Hogg, D.C.: Event model learning from complex videos using ilp. In: Proceeding of ECAI 2010, the 19th European Conference on Artificial Intelligence, pp. 93–98 (2010)
5. Foresti, G., Micheloni, C., Snidaro, L.: Event classification for automatic visual-based surveillance of parking lots. In: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, vol. 3, pp. 314–317. IEEE, Los Alamitos (2004)
6. Lee, S.W., Mase, K.: Activity and Location Recognition Using Wearable Sensors. IEEE Pervasive Computing 1(03), 24–32 (2002)
7. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proc. of 7th International Joint Conference on Artificial Intelligence (IJCAI), pp. 674–679 (1981)
8. Lv, F., Song, X., Wu, B., Singh, V., Nevatia, R.: Left luggage detection using bayesian inference. In: Proceedings of the 9th IEEE International Workshop (2006)
9. Piciarelli, C., Foresti, G., Snidaro, L.: Trajectory clustering and its applications for video surveillance. In: Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, AVSS 2005, vol. 18, pp. 40–45. IEEE, Los Alamitos (2005)
10. Porikli, F.: Learning object trajectory patterns by spectral clustering. In: 2004 IEEE International Conference on Multimedia and Expo (ICME), vol. 2, pp. 1171–1174. IEEE, Los Alamitos (2004)
11. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 593–600 (1994)
12. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Mach. Intell. 22(8), 747–757 (2000)
13. Wilson, A.D., Bobick, A.F.: Hidden Markov models for modeling and recognizing gesture under variation. International Journal of Pattern Recognition and Artificial Intelligence 15, 123–160 (2001)
14. Xiang, T., Gong, S.: Video behaviour profiling and abnormality detection without manual labelling. In: Tenth IEEE International Conference on Computer Vision, ICCV 2005, vol. 2 (2005)
15. Zadeh, L.: Similarity relations and fuzzy ordering. Information sciences 3, 159–176 (1971)
16. Zivkovic, Z.: Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern Recognition Letters (2006)