AUDIO-VIDEO EVENT RECOGNITION SYSTEM FOR PUBLIC TRANSPORT SECURITY

Van-Thinh Vu
Francois Bremond
Gabriele Davini
Monique Thonnat

INRIA
2004 route des Lucioles
06902 Sophia Antipolis
France

Quoc-Cuong Pham
Nicolas Allezard
Patrick Sayd

CEA/LIST
91191 Gif-sur-Yvette
France

Jean-Luc Rouas
Sébastien Ambellouis
Amaury Flancquart

INRETS/LEOST
20 rue Elisée Reclus BP 317
59666 Villeneuve d'Ascq
France

Keywords: audio-video surveillance, audio-video event, behavior analysis, event recognition.

Abstract

This paper presents an audio-video surveillance system for the automatic surveillance in public transport vehicle. The system comprises six modules including in particular three novel ones: (i) **Face Detection and Tracking**, (ii) **Audio Event Detection** and (iii) **Audio-Video Scenario Recognition**. The Face Detection and Tracking module is responsible for detecting and tracking faces of people in front of cameras. The Audio Event Detection module detects abnormal audio events which are precursor for detecting scenarios which have been predefined by end-users. The Audio-Video Scenario Recognition module performs high level interpretation of the observed objects by combining audio and video events based on spatio-temporal reasoning. The performance of the system is evaluated for a series of pre-defined audio, video and audio-video events specified using an audio-video event ontology.

1 Introduction

The French project SAMSIT (Système d'Analyse de Médias pour une Sécurité Intelligente dans les Transports publics) aims at conceiving solutions for the automatic surveillance in public transport vehicle (e.g. trains and metros) by analyzing human behaviors based on audio-video stream interpretation. Its goal is to take into account the characteristics of mobile spaces and the limitation of bandwidth of available communication systems for designing efficient embedded surveillance systems. The design of audio-video surveillance systems dedicated to mobile spaces implies developing and adapting audio-video processing algorithms and equipment for handling new types of environment. These algorithms have to be endowed with the ability of performing efficient pre-processing to transfer only pertinent information to higher level processes running at a distant operational headquarter for handling alert messages.

Many video understanding systems have already been developed in the computer vision community. Haritaoglu et al. [7] use shape analysis and tracking to locate people and their parts (e.g., head, feet) in image sequences. Oliver et al. [12] use Bayesian analysis to identify human interactions using trajectories obtained from a monocular image. Johnson and Hogg [9] have defined an efficient people tracker based on B-spline corresponding to people shape models. Nevertheless, few video understanding systems have been able to successfully combine audio-video interpretations in real world applications due to a large variety of video understanding issues. First, typical video processing challenges come from shadows, illumination changes, over-segmentations or miss-detections. Second, the tracking process remains a major issue since the loss of a tracked object prevents the analysis of its behavior. In addition, only few systems provide a true semantic video understanding.

Figure 1 shows a near real-time intelligent audio-video surveillance system based on a generic platform called SAMSIT. Such a system is composed of a knowledge base containing a priori knowledge and six main modules: (1) Object Detection and Tracking, (2) Face Detection and Tracking, (3) Temporal Multi-Camera Analysis, (4) Primitive Audio Event Detection, (5) Primitive Video Event Detection and (6) Audio-Video Event Recognition. Among these six modules, three of them are novel (i.e. Face Detection and Tracking, Audio Event Detection and Audio-Video Event Recognition) and are described in the following sections.

2 The Knowledge Base

The knowledge base contains all the information needed by the SAMSIT platform for recognizing efficiently behaviors of interest predefined by end-users (e.g. security operators of train companies). This knowledge base needs to be specified for each video surveillance system constructed from the SAMSIT platform. There are typically three types of a priori knowledge contained in this knowledge base: the **3D context** of the observed scene, the **calibration matrices** of cameras and the **models of scenarios of interest**.

The ***3D context*** of the observed scene describes the geometry and the semantics associated to the empty scene. These information includes:

+ the geometric descriptions (e.g. position, surface or volume) of zones of interest (e.g. a forbidden zone, an entrance zone) and of static objects of the empty scene (e.g. walls, doors, equipment).
+ the semantic properties associated to contextual objects including both their physical properties (e.g. transparency, reflexion) and functionnalities (e.g. seat, door).

The ***calibration matrices*** of the cameras allow the SAMSIT platform to calculate for all detected mobile objects their 3D positions in the real world from their 2D positions in the images. Combining the calculated 3D positions with the geometric information defined in the 3D context, the SAMSIT platform can determine how detected mobile objects interact with the observed scene.

The ***models of scenarios of interest*** are the scenarios specified by the end-users. The models of scenarios are defined using a description language designed in a generic framework [15] which has been conceived for other types of video understanding applications. These scenario models are then used by the SAMSIT platform for interpreting audio-video streams.

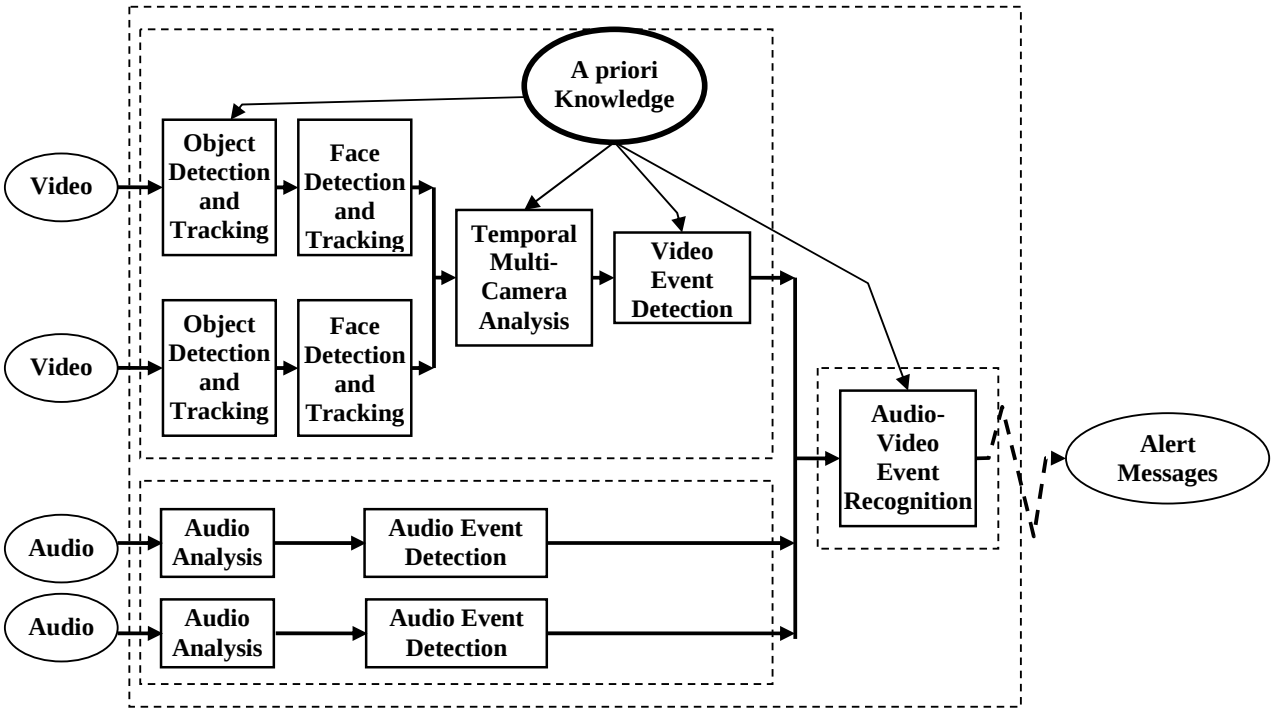

Figure 1: An intelligent audio-video surveillance system based on the SAMSIT platform: by using a priori knowledge and processing audio-video streams, the system triggers the alerts corresponding to the recognized scenarios.

3 Object Detection and Tracking

This first task consists in detecting and tracking mobile objects. The goal of the **Object Detector** is to detect for each frame the moving regions in the scene and classify them into a list of mobile objects with labels corresponding to their type based on their 3D size and their shape, such as PERSON. This task can be divided into three sub-tasks: ***detection of mobile objects***, ***extraction of features*** and ***classification of mobile objects***. A list of mobile objects is obtained at each frame. Each mobile object is described by 3D numerical parameters (center of gravity, position, height, width,...) and by a semantic class (PERSON, OCCLUDED PERSON, GROUP, CROWD, METRO TRAIN, SCENE OBJECT, NOISE or UNKNOWN). Several hard issues arise in this module. Surprisingly, the vibration of the train did not add too much noise in the detection. However, the strong changes in lighting conditions (e.g. train entering a tunnel) prevent SAMSIT system to correctly detect people inside the train at several occasions.

The goal of the **frame to frame tracker** (F2F Tracker) is to link from one frame to the next frame the list of mobile objects computed by the object detector. The output of the frame to frame tracker is a graph of mobile objects. This graph provides all the possible trajectories that a mobile object may have. The link between a new mobile object and an old one is computed depending on three criteria: the similitude between their semantic class, their 2D (in the image) and 3D (in the real world) distance.

4 Face Detection and Tracking

The face detection and tracking module is intended to complement the first module and provide inputs to the Video Event Detection module. Most of the time, faces can be seen in the field of view of one of the cameras in the sensors network, while the other body parts are often occluded,

especially by the seats. To accommodate the wide variety of face appearances, complex backgrounds, possible occlusions, multiple scales, and unpredictable motion, we developed a face detector based on a statistical modelling of face features. The detector was then combined to a particle filter to increase the temporal consistency of the detection.

4.1 Face Descriptors

The face features are captured by a set of local descriptors based on histograms of the gradient orientation, and weighted by the local gradient magnitude, computed in small rectangular areas inside the region of interest (Figure 2). These descriptors have already proven to be efficient for shape recognition tasks such as hand gesture recognition [5] and more recently for human detection [4]. They are in particular less sensitive to lighting conditions than intensity based methods.

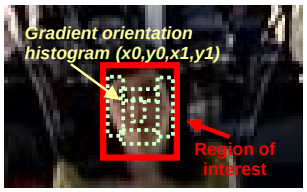

Figure 2: Local gradient orientation histograms.

4.2 Learning Faces

The face descriptors are used as classifiers. The most discriminative descriptors are searched by training with the *Adaboost* algorithm [6] on a large database containing face and non-face samples. In the training algorithm, a decision stump is associated to each histogram component, and the classifiers are cascaded as in [16]. Each stage of the cascade was trained to perform 99.9% of positive detection and 50% of false alarm.

4.3 Detection Step

The detection algorithm scans the entire image in a sliding window, at different scales and evaluates the classification cascade in the window. The cascade structure enables to drastically speed up the process, as it progressively rejects sub-images at each stage (Figure 3). Only a small number of feature evaluations are required on average.

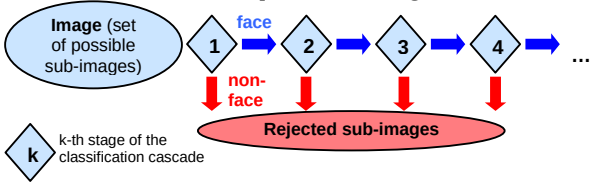

Figure 3: Detection based on a cascade of classifiers.

4.4 Tracking Faces with a Particle Filter

In order to temporally associate the detection results, a tracking module filter based on a color particle filter was implemented. The Particle filter, also known as the Condensation algorithm [8], is a powerful probabilistic method to estimate complex multimodal density distributions. Our particle filter is similar to [13], it computes the likelihood from the Bhattacharyya distance between color histograms in the HSV color space, and estimates state vectors in 5 dimensions (two for the position, two for the scale, and one for the angle to the horizontal direction).

4.5 Results

The database used in the learning process was composed of 3959 faces under various pose and lighting conditions. At each stage of the training process, 10000 negative samples were randomly selected from classification errors at the previous stage. The resulting cascade of classifiers contains 10 stages. The false positive rate after training was 1.1e-9**.** On a test sequence of 500 frames, we obtained a false positive rate equal to 1.7e-7 and a detection rate superior to 94%. Moreover, the false positive detections can be partly filtered out with the tracking algorithm. The face detection and tracking results are illustrated in Figure 4 and Figure 5.

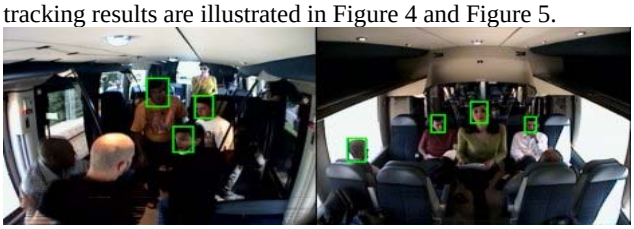

Figure 4: Face detection results.

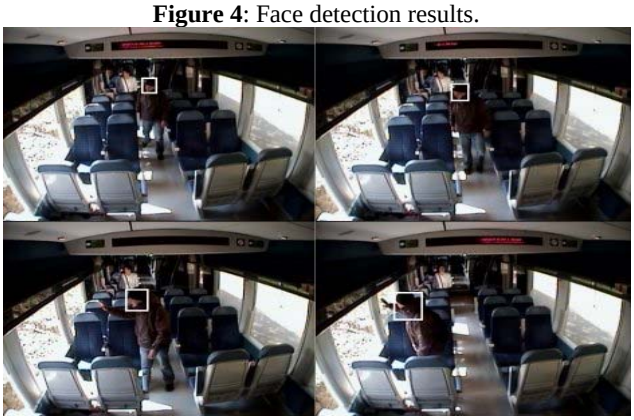

Figure 5: Face tracking results.

5 Temporal Multi-Camera Analysis

This module is composed of three steps: mobile object and face combination, multi-camera fusion and long-term tracking.

The first step aims at fusing the information coming from several cameras including both mobile object detection [section 3] and face detection results [section 4]. The first step of this task consists in computing the correspondences between mobile objects and faces. When a face is detected without corresponding mobile object in cases where people are motionless (e.g. sitting) for a long time, a mobile object is built based on the face information. In the same way, when a mobile object is detected without corresponding face in cases, for instance, when the person is not facing the camera, the

mobile object is still considered as valid. Therefore, after this process, most of the people present in the train are detected.

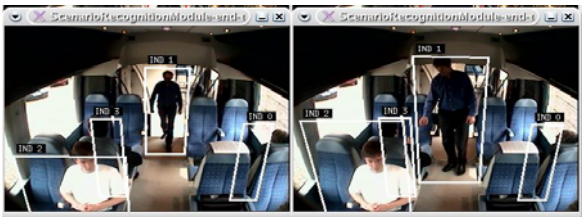

Figure 6: the tracked individuals keep the same identifier during all frames.

To discard erroneously detected mobile objects which correspond to noise, the video understanding process relies either on the combination of multi-camera information or on the verification of the temporal coherency on a temporal window. In order to take advantage of all the calibrated cameras viewing the same scene (cameras with overlapping field of views), we combine all the graphs of mobile objects computed by the previous tasks for each camera into a global one that we called the Combined Graph. As a result, the features (the 3D positions and the dimensions) of the mobile objects computed in the Combined Graph give a better estimation of the positions and the dimensions of the real persons evolving in the scene.

Another process consists in tracking individuals on a long period of time. This tracker performs a temporal analysis on the Combined Graph. This Long-term Tracker computes and selects the trajectories of mobile objects which can correspond to a real person thanks to an explicit model of person trajectory.

Figure 6 shows four individuals correctly tracked (i.e. they have the same identifier) during the whole video sequence.

6 Primitive Audio Event Detection

In this section, we present an intelligent microphone that aims at detecting and indentfying sounds inside the vehicle. We use a classical learning/classification method based on a Gaussian Mixture Model. We focus on the following sound classes: shouts, tag and noise events. Each of these sounds is a key element of the scenarios the audio-video system has to detect. The sound analysis system is divided in two main modules: the front-end processing module for activity zones detection and the classification module. The first one extracts relevant audio samples from the signal before the classification module estimates the most probable class to which they belong. Both segmentation and classification steps are described in the next sections. In the last part, cross-validation experiments are presented to evaluate the performance of the method.

6.1 Segmentation in Activity Zones

This task is based on 3 steps:

1. an automatic audio segmentation, which splits an audio signal in several quasi-stationary consecutive zones,
2. an activity detection algorithm, which aims to skip silence and low-level noise zones, out of interest and
3. a merging step, to gather successive activity segments.

It is issued from the "Forward-Backward Divergence" (DFB) algorithm [2]. The audio signal is described by an autoregressive gaussian model and the method consists in detecting the changes in the autoregressive models through the prediction errors computed on two analysis windows. The distance between the two models is obtained by computing the mutual entropy of the two corresponding conditional laws. Three kinds of segment are obtained, quasi-stationary segments, transient segments and short segments. Their lengths vary between 20 and 100 ms. Each segment is classify as silence or activity according to its energy. To avoid over-segmentation, reduce the computing time and the false detection rate, we merge quasi-adjacent segments. Two activity segments are quasi adjacent if they are separated by a non activity segment which duration is under 300 ms. The classification step is applied to each activity zones obtained after the merge task.

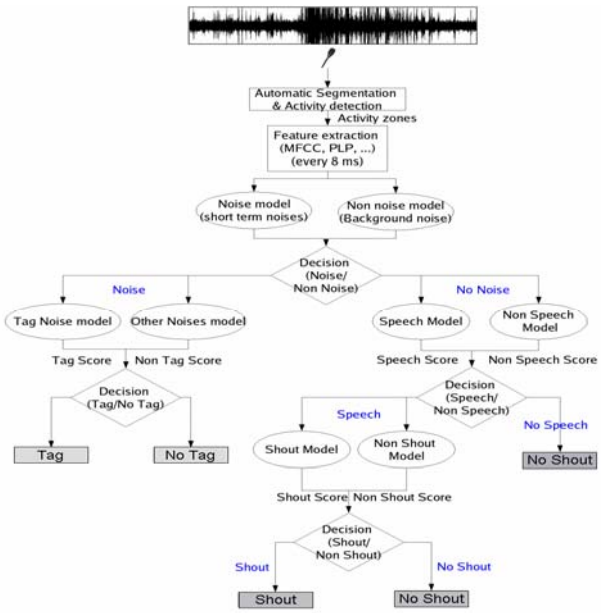

Figure 7: hierarchical classification tree.

6.2 Modeling and Classification Framework

The classification process is achieved according to the hierarchical tree described in Figure 7. Each model that appears as a component of the tree is computed during a training step. It is based on a GMM of a set of acoustical parameters extracted from a training data corpus. We have decided to extract the Mel Frequency Cepstral Coefficients (MFCC) to which we have added a term of energy and all first and second derivatives. Finally, the dimension of the features vector is 39.

GMM method supposes that the different classes which are represented in the feature space can be modeled with a weighted sum of Gaussian distributions. The parameters of the Gaussian mixture are estimated using the EM (Expectation-Maximisation) algorithm initialised with the LBG algorithm ([14] & [10]).

During the classification phase, all the activity segments detected in the test utterance are gathered and parameterised with the same features vector. The likelihood of each vector

according to each model is given by $P(Y|C_i) = \prod_{j=1}^{N} P(y_j|C_i)$ where $Y = \{y_1, y_2, ..., y_N\}$ is a set of activity segments and $P(y_j|C_i)$ denotes the likelihood of each segment relatively to the class C_i. $P(y_j|C_i)$ is approximated under the Winner Takes All (WTA) assumption [11].

6.3 Results of Cross-Validation

Scenes were recorded using simultaneously 4 microphones in regional train and in real condition. Actors were asked to play scenarios representative of the public french train operator's needs: fight scene, violent robbery scene, mobile snatching and spray paint scene. Each scenario is played several times. For each kind of scenario, the actors played a scene which is not a critical situation but has similar acoustic properties. The scene is called the "normal" situation. All files of the corpus have been labelled for learning and evaluation steps.

Cross-validation aims at estimating how well the model we have learned from some training data is going to perform on future unknown data. We have choosen the Leave-one-out Method. This method involves in three steps. Firstly, the model is trained on all the training data except for one. Secondly, the learned model is evaluated on the remaining data. Both steps are repeated such that each data is used once as the validation data. The evaluation process we achieved focuses not only on the good or bad detection of events, but also on the precision on the time scale of the detection. The results are then expressed as correctly identified durations and misidentified durations. This procedure is repeated for all the files of the corpus and for different number of Gaussian laws.

Results for shouts detection are displayed in Figure 7. In the graphics, correct identifications correspond to the white parts of bars, wrong classifications are represented by the grey part of bars and false alarms are displayed as a curve.

This graphic shows that the least false error rate is obtained using 1024 Gaussian laws in the mixture. For shout class, the duration of the false alarms (9.4 seconds) is relatively low compared to the total duration of the corpus (approximately 2540 seconds) and the total duration of shouts to identify (approximately 140 seconds). Shout detection rate does not seem very good. However, even if "all" shouts or spray paints are not always identified on the Scene cases, this GMM method performs well in terms of number of "identified events". A critical situation can be composed of several shouts or spray paints and the detection of a part of them can be sufficient to detect a critical event and to set off an alarm.

7 Primitive Video Event Detection

The primitive video event detection module aims at recognizing all primitive states corresponding to a stable temporal property of a mobile object (e.g. a person is close to a door). These primitive states are directly linked to visual features of mobile objects. The main difficulty of this module is to manage events involving a large number of mobile objects. Classical constraint resolution techniques get into a combinatorial explosion while coping with this problem. A dedicated constraint resolution technique is developed to reduce the processing time for obtaining a real-time recognition of this type of events [15].

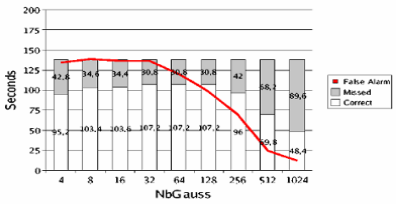

Figure 8: cross validation results for shout class.

8 Audio-Video Event Recognition

The audio-video event recognition aims at recognizing **complex temporal events** that combine both ***audio*** and ***video events***. Those events are defined in the knowledge base [section 2] and correspond to the terminal events (i.e. goals) of the application. There are two main issues arising in this module: the **representation** and the **real-time recognition** of complex audio-video events.

```
CompositeEvent( vandalism_against_window,
 PhysicalObjects( (vandal : Person))
 Components(
  (vandalism_against_window_VIDEO :
   CompositeEvent
    vandal_close_to_window(vandal))
  (vandalism_against_window_AUDIO :
   CompositeEvent
    tag_detected_close_to_person(vandal)))
 Constraints(
  (i_ of (Interval of
    vandalism_against_window_VIDEO) <=
     vandal's AEs's First's AEStartFrame)
  (_i of Interval of
    vandalism_against_window_VIDEO >=
     vandal's AEs's First's AEEndFrame))
 Alarm( AText("Vandalism against window")
        AType("VERYURGENT")
        APos2D(vandal->Pos2D)
        APos3D(vandal->Position)))
```

Figure 9: an audio-video event that is used in train surveillance.

8.1 Audio-Video Event Representation

The audio-video event representation corresponds to the modeling of all the knowledge used by the system to detect audio-video events occurring in the scene. To allow security operators to easily define and modify the event models, the description of the knowledge is declarative and intuitive (in natural terms). The proposed model of an audio-video event *E* is composed of five parts:

+ a set of Physical Object variables corresponding to the physical objects involved in *E*: any contextual object including static object (e.g. equipment, zone of interest) and mobile object (e.g. person, vehicle, train). The vehicle type can be of different subtypes to represent different vehicles types (e.g. train, car).
+ a set of temporal variables corresponding to the components (i.e. sub-events) of *E*.
+ a set of forbidden variables corresponding to the components that are not allowed to occur during *E*.
+ a set of constraints (including symbolic, logical, spatial and temporal constraints including Allen's interval algebra operators [1]) involving these variables.

+ a set of decisions corresponding to the tasks predefined by experts that need to be executed when *E* is detected (e.g. activating an alarm or displaying a message).

8.2 Audio-Video Event Recognition

The audio-video event recognition algorithm recognizes which events are occurring using primitive video events detected by the primitive video event detection module [section 0] and the audio events detected by the audio event detection module [section 6]. There are two main difficulties arising in this process: (i) the synchronization between audio and video events and (ii) the real-time recognition.
In order to facilitate event recognition, event templates (i.e. event models) are generated for each event, the last component of which corresponds to a recognized primitive event [section 0]. The event template contains the list of physical objects involved in the primitive state. These physical objects partially instantiate the event template. To recognize an event composed of two (or one) sub-events, given the event template partially instantiated, the recognition algorithm selects (if needed) a set of physical objects matching the remaining physical object variables of the event model. The algorithm then looks back in the past for any previously recognized state/event that matches the first component of the event model. If these two recognized components verify the event model constraints, the event is said to be recognized. In order to facilitate complex event recognition, after each event recognition, event templates are generated for all composite events, the last component of which corresponds to this recognized event. The recognition of complex event usually requires a search in a large space composed of all the possible combinations of components and objects. To avoid this combinatorial explosion, all complex events are simplified into events composed of at most 2 components through a stage of compilation in a preprocessing phase. Then the recognition of complex events is performed in a similar way to the recognition of events composed of two sub-events. The video event recognition algorithm is based on the method of Vu et al [15].

9 Results

The SAMSIT system has been tested on two sets of recorded audio-video streams. These sets have been realized on a train (TER 75700) equipped with four cameras facing each other and four microphones installed along the corridor. The first set has been done while the train was stopped whereas the second set has been recorded with the train in motion. Even if all scenes were acted by professionals, half of the scenes did not correspond to the end-user specifications (e.g. graffiti on seats rather on windows). Among the five scenarios specified by end-users four have been recognized in at least one video: vandalism against window, group fighting, theft and beggar. The scenario that was not recognized (group agitation) has been acted in difficult environment conditions: crowded, partially visible or strong light changing conditions. Among the four recognized scenarios, the scenario "vandalism against window" was the most successfully recognized. On the seven available videos illustrating this scenario, three were correctly recognized, two did not correspond to the specified scenarios, two were not showing the individual in action (i.e. the individual was out of view). The use of audio events was particularly useful in cases of the scenario "vandalism against window" where SAMSIT was able to detect out the bomb used to paint the window.

10 Conclusion

This paper has described an audio-video surveillance platform able to automatically recognize high level human behaviors involving individuals using both audio and video information. Different methods have been developed to compute specific types of behaviors under different configurations. These methods have been coherently integrated in the proposed framework. Despite hard visual conditions, SAMSIT system was able to recognize successfully several scenarios. There is still much work to be done to obtain a reliable system. On top of enhancing audio-video algorithms, a future direction consists in improving the lightning conditions by for instant increasing artificial lightning in the train.

References

[1] J. F. Allen. **Maintaining Knowledge about Temporal Intervals**. In Communications of the ACM, 26(11), 1983.

[2] R. André-Obrecht. **A New Statistical Approach for Automatic Speech Segmentation**. IEEE Transactions on Acoustic, Speech and Signal Processing, 36(1), 1988.

[3] F. Cupillard, F. Bremond and M. Thonnat. **Behaviour Recognition for Individuals, Groups of people and Crowd**. In IEE Proc. of the IDSS Symposium - Intelligent Distributed Surveillance Systems, London, 26 February 2003.

[4] N. Dalal and B. Triggs. **Histograms of Oriented Gradients for Human Detection**. International Conference on Computer Vision and Pattern Recognition, 2,886-893, June 2005.

[5] W. T. Freeman and M. Roth. **Orientation Histograms for Hand Gesture Recognition**. IEEE Intl. Workshop on Automatic Face and Gesture Recognition. Zurich. June 1995.

[6] J. H. Friedman, T. Hastie, and R. Tibshirani. **Additive logistic regression: a statistical view of boosting**. Dept. of Statistics, Stanford University Technical Report.1998.

[7] I. Haritaoglu, D. Harwood and L. Davis. **W4: Real-time surveillance of people and their activities**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8), 2000.

[8] M. Isard and A. Blake. **Condensation - conditional density propagation for visual tracking**. International Journal of Computer Vision, 29(1):5 - 28, 1998.

[9] N. Johnson and D. Hogg. **Learning the distribution of object trajectories for event recognition**. Image and Vision Computing, 14(8):609-615, 1996.

[10] Y. Linde, A. Buzo and R.M. Gray. **An algorithm for Vector Quantizer design**. IEEE Transactions on Communications, 28(1).

[11] S. Nowlan. **Soft competitive adaptation: neural network learning algorithm based on fitting statistical mixtures**. Ph.D. dissertation, School of Computer Science, CMU, 1991.

[12] N. Oliver, B. Rosario and A. Pentland. **A bayesian computer vision system for modelling human interactions**. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8).

[13] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. **Color-based probabilistic tracking**. In European Conference on Computer Vision, ECCV'2002, Copenhaguen, Denmark, June 2002.

[14] D.A. Reynolds and R.C. Rose. **Robust Text-Independent Speaker Identification Using Gaussian Mixtures Speaker Models**. IEEE Transactions on Speech and Audio Processing, 3(1), 1995.

[15] V. T. Vu, F. Bremond and M. Thonnat. **Automatic Video Interpretation: A Novel Algorithm for Temporal Scenario Recognition**. The Eighteenth International Joint Conference on Artificial Intelligence (IJCAI'03), Acapulco, 9-15 August 2003.

[16] P. Viola and M. Jones. **Robust Real-time Object Detection**. Second Int. Workshop on statistical and computational theories of vision - modeling, learning, computing, and sampling, Vancouver 2001.