Detecting abnormal activities in video sequences

Angelo Chianese
University of Naples
via Claudio, 21
80125, Naples, Italy
angchian@unina.it

Vincenzo Moscato
University of Naples
via Claudio, 21
80125, Naples, Italy
vmoscato@unina.it

Antonio Picariello
University of Naples
via Claudio, 21
80125, Naples, Italy
picus@unina.it

ABSTRACT

Automatically detecting suspicious human activities in restricted environments such as airports, parking lots and banks represents an open issue for the last generation surveillance systems. In this paper, we present an approach that allows to detect anomalies in a video sequence without any need of describing *a priori* "abnormal" activities. In particular, we first introduce a normal activities model based on the concept of elementary actions observable by means of image understanding procedures. We then provide an algorithm based on the use of *decision trees* that can quickly detect an abnormal situation as variation of currently processed activity with respect to normal patterns contained in the system knowledge base. Our preliminary experimental results on a dataset consisting of staged bank robbery videos show that our algorithm provides very encouraging results when compared to human reviewers.

1. INTRODUCTION

The task of designing algorithms that can analyze human activities in video sequences has been an active field of research during the last decade. Nevertheless, we are still far from a systematic solution to the problem. The analysis of activities performed by humans in restricted settings is of great importance in applications like automated surveillance systems. There has been significant interest in this area where the challenge is to be able to build a visual surveillance system that can automatically detect abnormal activities.

The practical applications of such a system could include airport monitoring, monitoring of activities in secure installations, surveillance in parking lots, etc. In such a system, for example a bank monitoring system, it is necessary to have a high degree of accuracy in detecting abnormal activities given the high stakes involved.

In this paper, we present an approach that allows to detect "anomalies" in a video sequence, without any need of describing *a priori* abnormal activities; thus avoiding the drawback of such techniques that is to know all the abnormal activities for the observed environments.

In particular, we first introduce a normal activities model based on the concepts of *complex activities* that are sequences of *atomic actions* directly observable through image understanding primitives and showing a precise semantic. We then provide a novel algorithm that can quickly detect an abnormal activity as soon as an unexpected "variation" in currently processed data occurs, with respect to "normal patterns" contained in the system knowledge base.

Our experimental results on a dataset consisting of staged bank robbery videos show that our algorithm provides quite good results when compared to human reviewers.

The paper is organized as follows: in section 2 we provide an overview of existing techniques for activity detection; section 3 outlines the system architecture; section 4 and 5 are dedicated to illustrate the image processing and reasoning modules of the system; section 6 contains the preliminary experimental results; some conclusions are discussed in section 7.

2. RELATED WORKS

The study of human activity has a wide literature. On one hand, there are model-based approaches where researchers have tried to describe activities at a higher level, and on the other hand, there are efforts to build algorithms that can learn patterns from training data and can detect activities using such patterns.

For the first kind of approaches, Neumann and Novak [12] proposed a hierarchical representation of event models, with each model being a template that can be matched with scene data. Natural language descriptions of activities can be mapped onto this hierarchical model.

Nagel [10] proposed an early approach to obtaining conceptual descriptions from image sequences, which could then be used for representing and recognizing activities. Dousson, Gaborit and Ghallab [5] have presented models and algorithms for situation analysis from video data.

Hidden Markov Models (HMMs) [3, 2], variations such as Coupled Hidden Markov Models (CHMMs) [4], and more recently, probabilistic graphs [1] have been used for activity recognition. Shet, Harwood and Davis [14] write Prolog programs to recognize activities in video. Naphade et al. [11] propose a graphical framework for detection of semantic concepts like sites, objects and simple events.

For the second type of techniques, Ivanov and Bobick [9] use *Stochastic Context Free Grammars* ($SCFG$) which is a probabilistic version of *context free grammars* (CFG) to de-

Ambi-sys '08, February 11-14, 2008, Quebec, Canada.

scribe normal activities. Another interesting approach is the *n-grams* technique proposed by Hamid et al. [7]. In the training step a video is divided in a set of n-grams that can be seen as events sequences of a given length. The n-grams are represented by means of apposite histograms and the detection step is based on the matching between the histograms of current events sequence and the n-grams histograms.

Eventually, ontologies have been recently used in different contexts for video surveillance. For example, Georis et al. [6] use ontologies to recognize activities in a bank monitoring setting, while Hobbs [8] introduces the *Video Event Representation Language* (*VERL*) for providing an ontological representation of complex events in terms of simpler sub-events.

3. SYSTEM ARCHITECTURE

Our video-surveillance system is able to detect suspicious events inside video sequences in an automatic way, that are acquired from fixed tv-cameras, without any human help. A logical view of our system architecture is at glance shown in figure 1. We have subdivided our system into 4 logical layers.

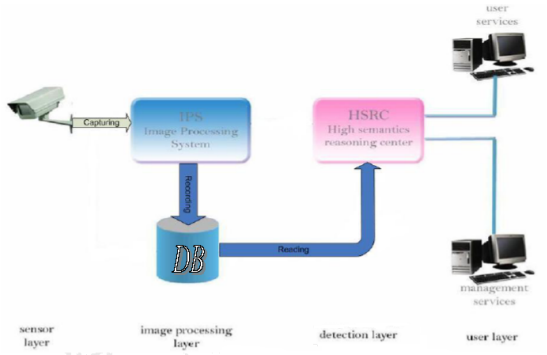

Figure 1: System Architecture

1. *Sensor Layer*: at this level we assume the presence of a certain number of sensors (in our work a fixed camera) that acquire a set of information (video frames in our case) to be processed.
2. *Image Processing Layer*: at this level the acquired video frames are processed by an *Image Processing System* (IPS), a software module capable of tracking, by means of appropriate image processing algorithms, from the current video frames some useful low-level information (e.g., objects presence in a frame, objects spatial position, objects motion, objects occlusions, etc...) necessary to the next high level analysis. Low level information are then converted in an apposite format that constitute the "video-labeling" stored into a dedicate database.
3. *Detection Layer*: At this level the *High Semantics Reasoning Center* (HSRC) software module, by analyzing the described low-level information, tries to detect the occurrences of an abnormal activity in a video, on the base of the knowledge of normal patterns, obtained in a training step.
4. *User Layer*: At this level we find the services offered to the users by means of apposite graphical user interfaces. The services can be classified into two types.
   - *Management Services* allow to train the system for the detection by apposite video training data.
   - *User Services* allow to configure and set some system working parameters, video visualization and analysis way and so on.

4. THE IMAGE PROCESSING SYSTEM

In this section we will provide the description of Image Processing System. We first describe the software module used for video frames analysis and then illustrate the image semantic management.

4.1 The Reading People Tracker

The IPS has been realized using the *Reading People Tracker* (RPT) software module provided with GPL license at the url www.siebel-research.de.

Such a system, by means of apposite algorithm of *object detection* and *object motion*, is able to track persons, vehicles, generic objects, etc. (*blobs*) inside a given scene. The general architecture of such a system is show in figure 2.

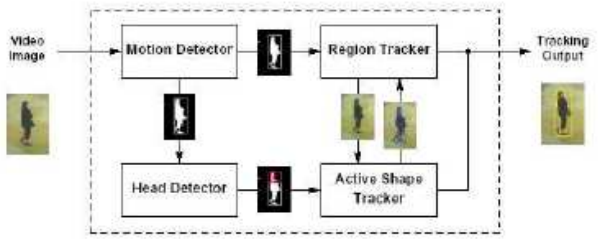

Figure 2: IPS Functional Architecture

We have the following components.

- *Motion Detector*: the outputs of this component are image regions containing "moving blobs" obtained by background substraction techniques.
- *Region Tracker*: the blobs inside the detected regions by Motion Detector are tracked by such a component. The tracking includes objects splitting and merging.
- *Head Detector*: used for detecting blob heads.
- *Active Shape Tracker*: such a module uses particular 2-D shape models to determine the kind of blobs (e.g., person, group of persons, vehicle, generic objects).

We have introduced a *Shadow Detector* module [13] in order to detect possible shadows improving the tracking in presence of objects splitting and merging (see figure 3).

4.2 Image Semantic Management

Some classes have been added in the RPT in order to, from one hand, convert image analysis in terms of elementary objects actions that constitute, as described in section 5.1, our *video labeling*, from the other one, to store such a labeling into an Oracle RDBMS.

Figure 3: Shadow Detection in IPS

In the current prototypal version of the system we are able to:

- verify if the object O_j is present in the frame f_i;
- verify if the object O_j is present in the frame f_i in a particular image region Z_k;
- verify with a probability p if the object O_j and O_k are in an occlusion situation in the frame f_i;
- verify with a probability p if the objects O_k and O_j are splitting in the frame f_i;
- verify with a probability p if the objects O_k and O_j are merging in the frame f_i.

The figure 4 shows an example of image processing output and labeling generation for a split between two objects.

$split(O_1, O_3, F1210)$

Figure 4: Split Predicate

The reasoning module will work on the top of such information trying of detecting abnormal activity in a video sequence.

5. THE REASONING MODULE

The main aim of this section is to describe the basic principles and the implemented solution at the basis of the proposed approach for detection of suspicious activities inside a given video sequence. In particular, our objective is to "capture" *abnormal* activities that can be seen as sudden and unexpected discontinuities in the *normal* activities "paths".

To these purposes, in a *training step* the system has to learn in an automatic way normal behaviors in video dataflow and store them in an apposite *Knowledge Base*.

In a successive *detection step* anomalies in video sequences are identified by considering variations of current processed video data with respect to the stored normal data sequences.

Figure 5 summarizes the architecture of the reasoning module: the video data sequences are coded in terms of *labeling* and processed, in the training phase, by the *Training Module* and, in the detection phase, by the *Detection Module* that communicates with an *Alerter Module* to generate alarms with different alert levels.

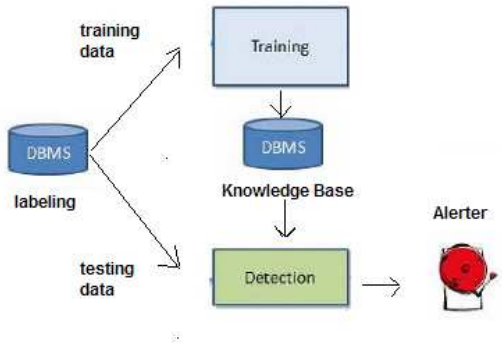

Figure 5: HSRC Architecture

In the following, we first provide some definitions useful to describe our activity model; we then describe the training and detection algorithms.

5.1 The Activity Model

As already described in section 3, the current implementation of the system is based on an apposite image processing engine capable of detecting *elementary actions* by directly "observing" video data. The output of such a module is a sequence of information that represent the temporal flow of objects actions in the video sequence.

DEFINITION 1 (ELEMENTARY ACTIONS). *An* elementary action *is the tuple:*

$$\rho = \langle \mu, \mathcal{X}, \tau, \pi \rangle \tag{1}$$

where:

1. $\mu \in \mathcal{V}$ *is the name of atomic action in according to a given dictionary* $\mathcal{V} = \{v_1, v_2, ..., v_m\}$*;*
2. $\mathcal{X} = \{x_1, x_2, ..., x_n\}$ *is set of parameters that complete semantic information associated to a given action (e.g. identifiers of involved video objects);*
3. τ *is the temporal instant in which the action occurred;*
4. π *is a measure of confidence in the actions detection.*

The available system for labeling generation produces the following model instance.

- The dictionary is constituted by $m = 5$ symbols. In particular:

$$V = \{objIn, objInZone, objOccl, objMerge, objSplit\} \tag{2}$$

- The maximum number of action parameters is in our model $n = 2$. We distinguish actions that involve one object (i.e. *objIn*) and actions that involve two objects (i.e. *objMerge*, *objSplit*, *objCccl*, *objInZone*).
- We assume as temporal instant τ the frame number in which an action occurred (the temporal features of an action are represented by the frame number).
- $0 \leq \pi \leq 1$

DEFINITION 2 (VIDEO LABELING). *A* video labeling *is an ordered sequence of elementary actions:*

$$\lambda = \rho_1, \rho_2, ..., \rho_l \tag{3}$$

EXAMPLE 1 (VIDEO LABELING). *In the following an example of video labeling produced by the image processing engine is shown.*

$\langle objIn, O1, null, 2007, 1.0\rangle$
$\langle objIn, O2, null, 2007, 1.0\rangle$
$\langle objOccl, O1, O2, 2008, 0.76\rangle$
$\langle objOccl, O1, O2, 2009, 0.70\rangle$
$\langle objIn, O3, null, 2009, 1.0\rangle$
$\langle objIn, O1, null, 2010, 1.0\rangle$
$\langle objIn, O2, null, 2010, 1.0\rangle$
$\langle objInZone, O3, Z1, 2010, 1.0\rangle$

DEFINITION 3 (COMPLEX ACTIVITY). *A* complex activity *is a video labeling ordered subsequence of variable length:*

$$\epsilon_i = \{\rho_k, ..., \rho_{k+\mathcal{L}-1}\} \tag{4}$$

where $\mathcal{L}_{min} \leq \mathcal{L} \leq \mathcal{L}_{max}$ is the subsequence length.

For better characterize a given activity we consider three kinds of *aggregation functions*:

1. a *parameters aggregation function* $\omega_1 : \mathcal{X}^{\mathcal{L}} \rightarrow \Re$ - it returns an aggregation measure (i.e. the count) of objects involved in the activity;
2. a *temporal aggregation function* $\omega_2 : \{\tau_1, ..., \tau_{\mathcal{L}}\} \rightarrow \Re$ - it return the temporal interval (i.e. the number of frame) in which an activity occurred;
3. a *probability aggregation function* $\omega_3 : \{\pi_1, ..., \pi_{\mathcal{L}}\} \rightarrow \Re$ - it returns a joint measure (i.e. the minimum probability) of the confidence probabilities related to a given activity.

EXAMPLE 2 (AGGREGATION FUNCTIONS). *Let us consider the following activity ϵ_i with $\mathcal{L} = 2$.*

$\langle objIn, O1, null, 2007, 1.0\rangle$
$\langle objIn, O2, null, 2008, 1.0\rangle$
$\langle objOccl, O1, O2, 2009, 0.76\rangle$
$\langle objIn, O1, null, 2010, 1.0\rangle$
$\langle objIn, O2, null, 2010, 1.0\rangle$

We have the following values for aggregation functions: $\omega_1(\epsilon_i) = 2$, $\omega_2(\epsilon_i) = 4$, $\omega_3(\epsilon_i) = 0.76$

DEFINITION 4 (NORMAL OR ABNORMAL ACTIVITY). *A* normal activity *is any actions sequence observed in the training step, an* abnormal activity *is an activity that has not been observed in the training process.*

In order to generate a *Knowledge Base* containing all normal activities for a given video environment, it is useful to group in apposite *classes* all the activities with the same length and semantic and to represent them using opportune features.

DEFINITION 5 (ACTIVITY CLASS). *An* activity class *is a set of activities having the same length and presenting the same ordered sequence of actions names.*

$$\Sigma = \epsilon_1, ..., \epsilon_{l_c} \tag{5}$$

A class can be represented by a features matrix containing for each component activity the values of aggregation functions.

DEFINITION 6 (FEATURES MATRIX). *A* Features matrix *is a matrix:*

$$\mathcal{M} = \{m_{j,i}\} = \{\omega_j(\epsilon_i)\} \tag{6}$$

where $\epsilon_i \in \Sigma$ is the i-th element of the class and ω_j is the output of j-th aggregation function.

EXAMPLE 3 (AGGREGATION FUNCTIONS). *In the following an example of class and its feature matrix is reported.*

$\Sigma = \{\epsilon_1, \epsilon_2, \epsilon_3\}$,
$\epsilon_1 = \{\langle objIn, O1, null, 2007, 1.0\rangle,$
$\langle objIn, O2, null, 2007, 1.0\rangle,$
$\langle objOccl, O1, O2, 2010, 0.80\rangle\}$
$\epsilon_2 = \{\langle objIn, O3, null, 3000, 1.0\rangle,$
$\langle objIn, O4, null, 3007, 1.0\rangle,$
$\langle objOccl, O3, O4, 3010, 0.70\rangle\}$
$\epsilon_3 = \{\langle objIn, O1, null, 5000, 1.0\rangle,$
$\langle objIn, O2, null, 5000, 1.0\rangle,$
$\langle objOccl, O1, O6, 5006, 0.65\rangle\}$

	ϵ_1	ϵ_2	ϵ_3
ω_1	*2*	*4*	*0.80*
ω_2	*2*	*11*	*0.70*
ω_3	*3*	*7*	*0.65*

Starting from the features matrixes, a generic class can be represented in a more compact way by means of the following structure:

$$\Theta = \langle \mathcal{N}, \mathcal{P}, \mathcal{T}, \mathcal{C} \rangle \tag{7}$$

where:

- $\mathcal{N}$ is a subset of the codomain of injective function $\eta : \mathcal{V}^{\mathcal{L}} \rightarrow \mathcal{S}$, that associates for each names-tuple $\rho_{i_1}.\mu, ..., \rho_{i_{\mathcal{L}}}.\mu$ a set of symbols. An adopted function is:

$$\eta : \{objIn, objInZone, objSplit, objMerge, objOccl\} \rightarrow \{I, Z, S, M, O\} \tag{8}$$

- $\mathcal{P} = \{\kappa_1, ..., \kappa_u\}$ is a set of properties matching the class parameters aggregation values; for example, $\kappa_h = \mathcal{O}_{min} \leq \omega_1(\mathcal{M}) \leq \mathcal{O}_{min}$, where $\mathcal{O}_{min}$ and $\mathcal{O}_{max}$ are respectively the minimum and maximum number of involved objects for each component activity.

- $\mathcal{T} = \{t_1, ..., t_y\}$ is a set of properties matching the class temporal aggregation values satisfy; for example, $t_h = \Delta\mathcal{F}_{min} \leq \omega_2(\mathcal{M}) \leq \Delta\mathcal{F}_{min}$, where $\Delta\mathcal{F}_{min}$ and $\Delta\mathcal{F}_{max}$ are respectively the minimum and maximum frame number in which each component activity occurs.
- $\mathcal{C}$ represents the number of component activities that satisfy a set $\mathcal{A}$ of constraints on probability aggregation values: $\mathcal{C} = \{\epsilon_i \in \Sigma : \mathcal{A}(\omega_3(\epsilon_i)) = true\}$. A typical constraint is: $\omega_3(\epsilon_i) \geq \iota$, being ι a given threshold.

EXAMPLE 4 (CLASS REPRESENTATION). *The class of example 3 is characterized by the following structure using a threshold* $\iota = 0.70$.

$$\Theta = \langle \{IIO\}, 2 \leq \omega_1(\mathcal{M}) \leq 3, 4 \leq \omega_2(\mathcal{M}) \leq 11, 2 \rangle \quad (9)$$

5.2 Training Step

The training module allows the generation of the *System Knowledge Base* containing information about all activity classes observed in the training set.

The video labeling related to *normal data* is submitted to the training module that, by analyzing the video events of length $\mathcal{L}$ with $\mathcal{L}_{min} \leq \mathcal{L} \leq \mathcal{L}_{max}$, discovers the activity classes and update their representative features (i.e. elements of set Θ). All the discovered information in the training step are then organized in an apposite *decision tree* that allows a rapid access to the data. An insert of a new normal activity in the training set causes a decision tree updating.

Each tree-node contains the information of a generic activity class and it is created only at the moment in which the first activity occurrence is detected, thus avoiding to generate the entire tree at the beginning; while the detection of other occurrences causes the update of class information.

At the end of the training, a given node is not instanced if one of these conditions is satisfied:

- in the training phase any activity with the related predicate-names sequence has been found;
- any related activity do not satisfied the set of probability constraints.

The following example shows the procedure at the basis of creation of the decision tree. We assume: (i) $\mathcal{V} = \{in, split\}$, (ii) $\mathcal{L} \in [2, 3]$.

EXAMPLE 5. *Let us considering the following video labeling:*

$\langle objIn, O1, -, 1, 1.0 \rangle$
$\langle objSplit, O1, O2, 2, 0.85 \rangle$
$\langle objIn, O1, -, 2, 1.0 \rangle$
$\langle objIn, O2, -, 2, 1.0 \rangle$
$\langle objIn, O1, -, 3, 1.0 \rangle$
$\langle objIn, O2, -, 3, 1.0 \rangle$
$\langle objIn, O2, -, 4, 1.0 \rangle$

In the tables 1 and 2 the information about generated classes (of length 2 and 3 respectively) are reported.

It is possible to note as all video actions are analyzed generating the decision tree (shown in figure 6): the algorithm for tree construction is schematized in the following.

$\mathcal{N}$	ϵ_1	ϵ_2	ϵ_3	$(\mathcal{O}_{min}, \mathcal{O}_{max})$	$(\Delta\mathcal{F}_{min}, \Delta\mathcal{F}_{max})$	$\mathcal{C}$
(I,S)	2	2	0.85	(2,2)	(2,2)	1
(S,I)	2	1	0.85	(2,2)	(1,1)	1
(I,I)	2	1	1	(1,2)	(1,2)	4
(I,I)	2	2	1	-	-	
(I,I)	2	1	1	-	-	
(I,I)	1	2	1	-	-	

Table 1: Class information for activities of length 3

$\mathcal{N}$	ϵ_1	ϵ_2	ϵ_3	$(\mathcal{O}_{min}, \mathcal{O}_{max})$	$(\Delta\mathcal{F}_{min}, \Delta\mathcal{F}_{max})$	$\mathcal{C}$
(I,S,I)	2	2	0.85	(2,2)	(2,2)	1
(S,I,I)	2	1	0.85	(2,2)	(1,1)	1
(I,I,I)	2	2	1	(2,2)	(2,2)	3
(I,I,I)	2	2	1	-	-	
(I,I,I)	2	2	1	-	-	

Table 2: Class information for activities of length 3

5.3 Detection Step

At the end of the training phase, the system is able to analyze a generic video labeling and to determine the abnormal activities by means of the class tree structure. In particular for each complex activity of video labeling, the tree is scanned to retrieve the corresponded class. If the tree-search outcome is negative the activity is labeled as abnormal. In the case of positive search, the event aggregation values are examined in order to verify if they satisfy class constraints, and only in this situation, the activity is classified as normal. The algorithm 2 shows the detection step. We suppose the presence of functions:

- $\nu(\epsilon_i)$ returns the label for the activity ϵ_i;
- node tree-scan(ϵ_i): verifies if there exists a node in the class-tree corresponding to activity ϵ_i;
- bool check-constraints(ϵ_i,Σ): verifies if ϵ_i satisfies the constraints of class Σ.

In order to reduce the number of false alarms in the detection task, we have introduced an *anomaly indicator* able to indicate for each abnormal activity the alert degree. It is based on the following consideration:

- the first occurrence of abnormal activities does not have to generate an high level of alert;
- the alert level depends on activities length (lengthy events generate high levels of alert);

Algorithm 1 Tree Building

```
for (each ε_i ∈ λ) do
  if (∃ class Σ : ε_i ∈ Σ) then
    update node
  else
    create new node
    link new node to the tree
  end if
end for
```

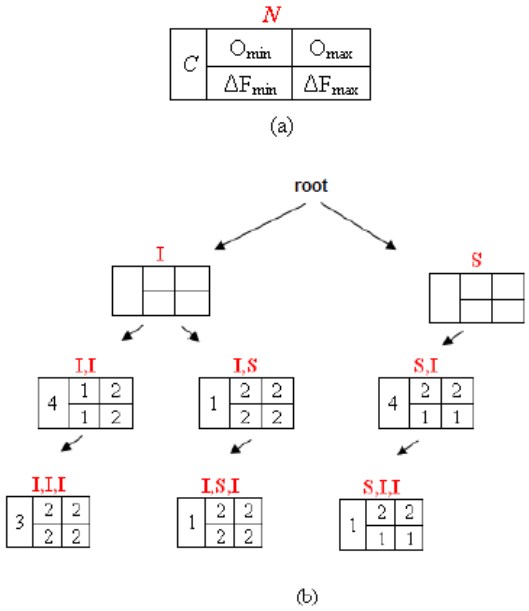

Figure 6: Training Tree: a) class-node; b) a tree example

Algorithm 2 Detection

```
for (each ϵ_i ∈ λ : L(ϵ_i) ∈ [L_min, L_max]) do
  ν(ϵ_i) =abnormal
  Σ=tree-scan(ϵ_i)
  if (Σ ≠ NULL) then
    if check-constraints(ϵ_i,Σ) then
      ν(ϵ_i) =normal
    end if
  end if
end for
```

- the alert level depends on activities confidence (events with a low probability generate low levels of alert);
- the alert level of the current activity depends on the past activities.

To these reasons an apposite *temporal window* $\mathcal{W}$ has been defined with the aim of storing the $\sigma(\cdot\mathcal{L}_{max} - \mathcal{L}_{min} + 1)$ past events w.r.t current activities and the related labeling (σ is a variable used to expand/reduce the temporal window). In particular, we have:

$$\mathcal{W} = \Gamma \cup \Omega \tag{10}$$

where Γ and Ω are the activities labeled as normal and abnormal respectively in the temporal window.

The anomaly indicator is the calculated as follows:

$$\mathcal{IA} = \frac{\sum_{\epsilon_i \in \Gamma} \log_b \mathcal{C}_i \cdot \omega_2(\epsilon_i) \cdot \omega_3(\epsilon_i)}{\sum_{\epsilon_i \in \Gamma} \log_b \mathcal{C}_i \cdot \omega_2(\epsilon_i) \cdot \omega_3(\epsilon_i) + \sum_{\epsilon_i \in \Omega} \omega_2(\epsilon_i) \cdot \omega_3(\epsilon_i)} \tag{11}$$

where $\mathcal{C}$ represent the number of activities of a given class and usually $b = 10$.

We distinguish on the base of such an indicator values three levels of alert: *low* or *green* ($\mathcal{IA} \rightarrow 1$), *medium* or *yellow* and *high* or *red* ($\mathcal{IA} \rightarrow 0$).

6. EXPERIMENTAL RESULTS

Our experiments were performed on a dataset consisting of 7 videos of 15 - 20 seconds in length, some depicting staged bank attacks and some depicting day-to-day bank operations; the dataset was thoroughly documented in [15]. Figure 7 contains a few frames from a video sequence depicting a staged bank attack and figure contains a few frames from a video sequence depicting regular bank operations.

Figure 7: Example video sequence of a simulated bank attack

Figure 8: Example video sequence of regular customer interaction at a bank

The data set has been divided in two parts: (i) a part of sub-videos containing normal activities are used as training set for the training step, (ii) the remanent part in combination with other video sequences contained bank attacks are used as testing set for the detection step.

Differently from other reasoning algorithms that return a set of frames and use classical precision and recall metrics [1], our algorithm returns an alarm indicator. For this reason, to evaluate the effectiveness of our approach w.r.t. an human observer, we have used a particular recall measure $\mathcal{R}$ that has been calculated by the following experimental protocol.

1. $\mathcal{R}$=0.
2. The system and human observer associate an alert level for each frame f in the testing set. In particular:
   - *red* for frames belonging to abnormal sequences w.r.t. the testing set;
   - *yellow* for frames belonging to medium alert sequences w.r.t. the testing set;
   - *green* for frames belonging to normal sequences w.r.t. the testing set.
3. For each frame f in the testing set:
   - $\mathcal{R} = \mathcal{R} + 1$ if human and system judgment agree;

- $\mathcal{R} = \mathcal{R} + 0.5$ if the system judgment is *yellow* (*red* or *green*) and the human judgment is *red* or *green* (*yellow*);
- $\mathcal{R} = \mathcal{R} + 0$ if the system judgment is *red* (*green*) and the human judgment is *green* (*red*);

4. $\mathcal{R} = \mathcal{R}/N_f$, where N_f is the number of frame in he testing set.

We have used as testing set a video sequences of 500 frames with the following activities:

- 1-103: Customer enters in the bank;
- 104-128: Two thieves enter in the bank;
- 129-500: Bank robbery (employer is forced to pick up money from the safe zone).

Figure 9 and table 3 report the comparison between human and system in terms of recall for different combination of algorithm parameters.

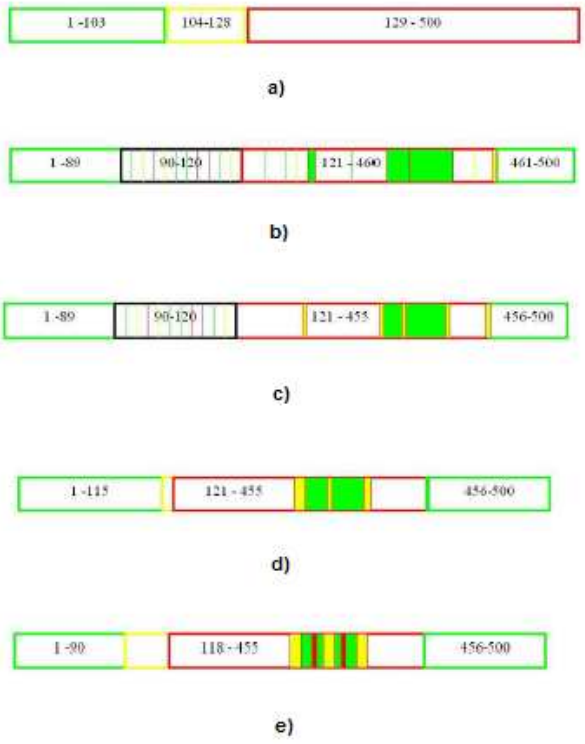

Figure 9: a) Human judgment; b,c,d,e) System judgments for different parameters configurations

We can observe that the system labels incorrectly as normal the situations in which: (i) the thieves enter in the bank, (ii) the thief and employer enter in the safe zone and are not visible to camera, (iii) the thieves leave the bank. This situation is caused by incapability of the used image processing engine to the distinguish a thief from a customer or employer.

Eventually, the obtained recall value can be compared with the technique proposed in [1] that uses the same dataset.

Case	System Parameters	$\mathcal{R}$
(b)	$\mathcal{L}_{min} = 2, \mathcal{L}_{max} = 3, \sigma = 3, \iota = 0.8$	62%
(c)	$\mathcal{L}_{min} = 4, \mathcal{L}_{max} = 6, \sigma = 3, \iota = 0.8$	68%
(d)	$\mathcal{L}_{min} = 4, \mathcal{L}_{max} = 6, \sigma = 15, \iota = 0.8$	76%
(e)	$\mathcal{L}_{min} = 6, \mathcal{L}_{max} = 10, \sigma = 15, \iota = 0.8$	84%

Table 3: System results for different parameters configuration

Such an algorithms obtains for a ground truth of four human observers an average frame recall lower than 80%.

7. CONCLUSIONS AND FUTURE WORKS

In this paper, we have presented an approach that allows to detect anomalies in a video sequences without any need of describing a priori abnormal activities. In particular, we have introduced a normal activities model based on the concept of atomic actions and complex activities and provided a novel algorithm capable of quickly detecting an abnormal activity as variation of current processed data with respect to normal patterns contained in the system knowledge base. Our experimental results on a dataset consisting of staged bank robbery videos have shown that our algorithm provides quite good results when compared to human reviewers. Future efforts will be devoted to enlarge our experimentation to more standard data set, in order to compare our approach with other ones presented in the literature, such as HMM anc CFS.

8. REFERENCES

[1] M. Albanese, V. Moscato, A. Picariello, V. S. Subrahmanian, and O. Udrea, Detecting Stochastically Scheduled Activities in Video, *International Joint Conference on Artificial Intelligence*, pages 1802-1807, 2007.

[2] F. Bremond and M. Thonnat, Analysis of Human Activities Described by Image Sequences, *Int. Florida AI Research Symp.*, 1997.

[3] H. Buxton and S. Gong, Visual Surveillance in a Dynamic and Uncertain World, *Artif. Intell.*, 78(1-2):431-459, 1995.

[4] Q. Cai and J. Aggarwal, Tracking Human Motion Using Multiple Cameras, *13th International Conference on Pattern Recognition*, 1996.

[5] C. Dousson, P. Gaborit, and M. Ghallab, Situation Recognition: Representation and Algorithms, *IJCAI*, pp. 166-174, 1993.

[6] B. Georis, M. Maziere, F. Bremond, and M. Thonnat, A Video Interpretation Platform Applied to Bank Agency Monitoring, *2nd Workshop on Intelligent Distributed Surveillance Systems (IDSS 04)*, 2004.

[7] R.Hamid, A.Johnson, S.Batta, A. Bobick, C.Isbell, G.Coleman, Detection and Explanation of Anomalous Activities:Representing Activities as Bagas of Event n-Grams, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005.

[8] J. Hobbs, R. Nevatia, and B. Bolles, An Ontology for Video Event Representation, *IEEE Workshop on Event Detection and Recognition*, 2004.

[9] Y. Ivanov and A. Bobick, Recognition of Visual Activities and Interactions by Stochastic Parsing, *IEEE Trans. Pattern Anal. Mach. Intell.*, 22 (8):852-872, 2000.

[10] H. Nagel, From Image Sequences Towards Conceptual Descriptions, *Image Vision Comput.*, 6(2):59-74, 1988.

[11] M. R. Naphade and T. S. Huang, A Probabilistic Framework for Semantic Indexing and Retrieval in Video, *IEEE International Conference on Multimedia and Expo*, 2000.

[12] B. Neumann and H. Novak, Event Models for Recognition and Natural Language Description of Events in Real-World Image Sequences, *IJCAI*, 724-726, 1983.

[13] A. Prati, I. Mikic, M. M. Trivedi, and R. Cucchiara, Detecting Moving Shadows: Formulation, Algorithms and Evaluation, *IEEE Trans. on PAMI*,25(7): 918-924, 2003.

[14] D. H. V. D. Shet and L. S. Davis, Vidmap: Video monitoring of activity with prolog, *IEEE International Conference on Advanced Video and Signal based Surveillance (AVSS)*, Como (Italy), 2005.

[15] F. Vu, V. T.and Bremond, and M. Thonnat, Automatic Video Interpretation: A Novel Algorithm for Temporal Scenario Recognition, *Eighteenth International Joint Conference on Artificial Intelligence*, 2003.