Finding "Unexplained" Activities in Video

Massimiliano Albanese[1], **Cristian Molinaro**[1], **Fabio Persia**[2],
Antonio Picariello[2], **V. S. Subrahmanian**[1]
[1]University of Maryland Institute for Advanced Computer Studies,
{albanese,molinaro,vs@umiacs.umd.edu}
[2]Università di Napoli Federico II, Napoli, Italy,
{fabio.persia,picus}@unina.it

Abstract

Consider a video surveillance application that monitors some location. The application knows a set of activity models (that are either normal or abnormal or both), but in addition, the application wants to find video segments that are unexplained by any of the known activity models — these unexplained video segments may correspond to activities for which no previous activity model existed. In this paper, we formally define what it means for a given video segment to be unexplained (totally or partially) w.r.t. a given set of activity models and a probability threshold. We develop two algorithms – FindTUA and FindPUA – to identify *Totally* and *Partially Unexplained Activities* respectively, and show that both algorithms use important pruning methods. We report on experiments with a prototype implementation showing that the algorithms both run efficiently and are accurate.

1 Introduction

Video surveillance is omnipresent. For instance, airport baggage areas are continuously monitored for suspicious activities. In crime-ridden neighborhoods, police often monitor streets and parking lots using video surveillance. In Israel, highways are monitored by a central authority for suspicious activities. However, all these applications search for *known* activities – activities that have been identified in advance as being either "normal" or "abnormal". For instance, in the highway application, security officers may look both for normal behavior (e.g. driving along the highway in a certain speed range unless traffic is slow) as well as "suspicious" behavior (e.g. stopping the car near a bridge, taking a package out and leaving it on the side of the road before driving away).

Most past work on activity detection uses a priori definitions of normal/abnormal activities and explicitly searches for activity occurrences [Hongeng and Nevatia, 2001]. Hidden Markov Models have been applied to problems ranging from gesture recognition [Wilson and Bobick, 1999] to complex activity recognition [Vaswani *et al.*, 2005]. [Oliver *et al.*, 2002] describes an approach based on coupled HMMs (CHMMs) for learning and recognizing human interactions. Dynamic Bayesian Networks (DBNs) have been used to capture causal relationships between observations and hidden states by [Hamid *et al.*, 2003] who used them to detect complex, multi-agent activities. [Albanese *et al.*, 2007] developed a stochastic automaton based language to detect activities in video, while [Cuntoor *et al.*, 2008] presented an HMM-based algorithm. Alternatively, [Zhong *et al.*, 2004] learn models of normal behavior and detect anomalies by finding deviations from normal behaviors. An approach for both single and multiple actor activity recognition has been proposed in [Hongeng *et al.*, 2004], where Bayesian networks and probabilistic finite state automata are used to describe single actor activities, and the interaction of multiple actors is modeled by an event graph.

In contrast, in this paper, we assume we are given some set $\mathcal{A}$ of activity definitions expressed as stochastic automata with temporal constraints — we extend the stochastic automata of [Albanese *et al.*, 2007] as this seems to be a common denominator underlying the HMM and DBN frameworks. $\mathcal{A}$ can consist of either "normal" activities or "suspicious" activities or both. In this paper, we try to find video sequences that are not "explained" by any of the activities in $\mathcal{A}$. As an example of why detecting activities that are neither 'normal', nor 'abnormal' is a very important problem in many applications, consider a video surveillance application in an airport. While we do want to find instances of known suspicious activities, there may be many other dangerous activities for which no model exists yet. Such previously "unknown" activities can be flagged as unexplained activities in our framework.

In order to achieve this, we define a possible-worlds based model and define the probability that a sequence of video is totally (or partially) unexplained. Based on this, users can specify a probability threshold and look for all sequences that are totally (or partially) unexplained with a probability exceeding the threshold. We then define two algorithms – the FindTUA algorithm finds all subsequences of a video that are totally unexplained, while the FindPUA algorithm finds all subsequences of the video that are partially unexplained. We develop a prototype implementation and report on experiments showing that the algorithms are accurate and efficient.

The paper is organized as follows. Section 2 sets up the basic definitions of the activity model extending [Albanese *et al.*, 2007] in a straightforward way. Section 3 defines the

probability that a video sequence is totally (or partially) unexplained. Section 4 derives a set of theorems that enable fast search of totally and partially unexplained video sequences. Section 5 presents the FindTUA and FindPUA algorithms. Section 6 describes our experiments.

2 Basic Activity Model

This section presents a slight variant of the stochastic activity model of [Albanese *et al.*, 2007] as a starting point. The novel contributions of this paper start in the next section. We assume the existence of a finite set $\mathcal{S}$ of *action symbols*, corresponding to atomic actions that can be detected by image understanding methods.

Definition 2.1 (Stochastic activity) *A* stochastic activity *is a labeled directed graph $A = (V, E, \delta, \rho)$ where*

- *V is a finite set of nodes labeled with action symbols from $\mathcal{S}$;*
- *$E \subseteq V \times V$ is a set of edges;*
- *$\delta : E \rightarrow \mathbb{N}^+$ is a function that associates, with each edge $e = (v_i, v_j)$, an upper bound on the time that can elapse between v_i and v_j;*
- *ρ is a function that associates, with each node $v \in V$ having out-degree 1 or more, a probability distribution on $\{\langle v, v' \rangle \mid \langle v, v' \rangle \in E\}$, i.e., $\sum_{\langle v,v' \rangle \in E} \rho(\langle v, v' \rangle) = 1$;*
- *$\{v \in V \mid \nexists\, v' \in V \text{ s.t. } \langle v', v \rangle \in E\} \neq \emptyset$, i.e., there exists at least one* start node *in the activity definition;*
- *$\{v \in V \mid \nexists\, v' \in V \text{ s.t. } \langle v, v' \rangle \in E\} \neq \emptyset$, i.e., there exists at least one* end node *in the activity definition.*

Figure 1 shows an example of stochastic activity modeling deposits at an Automatic Teller Machine (ATM). Each edge e is labeled with $(\delta(e), \rho(e))$. For instance, the two edges starting at node insertCard mean that there is a 50% probability of going to node insertChecks and a 50% probability of going to node insertCash from node insertCard. In addition, it is required that insertChecks and insertCash follow insertCard within 2 and 1 time units, respectively. In general, actions can be easily detected by either an image processing algorithm (e.g. detectPerson would check if a person is present in the image) or a sensor (e.g. to detect if insertCard holds).

The basic difference between this model and the one of [Albanese *et al.*, 2007] is the addition of the function δ which allows us to express constraints on the maximum "temporal distance" between two actions for them to be considered part of the same activity.

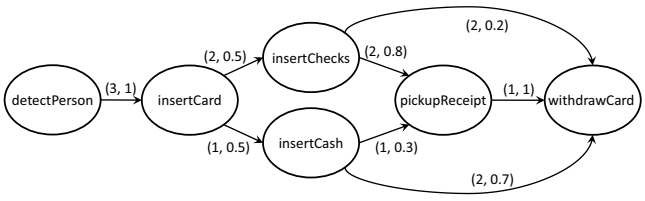


Figure 1: Example of stochastic activity: ATM deposit

Definition 2.2 (Stochastic activity instance) *An* instance *of a stochastic activity (V, E, δ, ρ) is a sequence $\langle s_1, \ldots, s_m \rangle$ of nodes in V such that*

- *$\langle s_i, s_{i+1} \rangle \in E$ for $1 \leq i < m$;*
- *$\{s \mid \langle s, s_1 \rangle \in E\} = \emptyset$, i.e., s_1 is a start node; and*
- *$\{s \mid \langle s_m, s \rangle \in E\} = \emptyset$, i.e., s_m is an end node.*

The probability of the instance is $\prod_{i=1}^{m-1} \rho(\langle s_i, s_{i+1} \rangle)$.

Thus, an instance of a stochastic activity A is a path in A from a start node to an end node. In Figure 1, ⟨detectPerson, insertCard, insertCash, withdrawCard⟩ is an instance with probability 0.35.

A *video* is a finite sequence of frames. Each frame f has an associated timestamp, denoted $f.ts$; without loss of generality, we assume timestamps to be positive integers. A *labeling* ℓ of a video v is a mapping $\ell : v \rightarrow 2^{\mathcal{S}}$ that takes a video frame $f \in v$ as input, and returns a set of action symbols $\ell(f) \subseteq \mathcal{S}$ as output. Intuitively, a labeling can be computed via an appropriate suite of image processing algorithms and specifies what actions are detected in each frame of a video.

Example 2.1 *Consider a video $v = \langle f_1, f_2, f_3, f_4, f_5 \rangle$, with $f_i.ts = i$ for $1 \leq i \leq 5$. A possible labeling ℓ of v is: $\ell(f_1) = \{$detectPerson$\}$, $\ell(f_2) = \{$insertCard$\}$, $\ell(f_3) = \{$insertCash$\}$, $\ell(f_4) = \{$withdrawCash$\}$, $\ell(f_5) = \{$withdrawCard$\}$.*

Throughout the paper, we use the following terminology and notation for sequences. Let $S_1 = \langle a_1, \ldots, a_n \rangle$ and $S_2 = \langle b_1, \ldots, b_m \rangle$ be two sequences. We say that S_2 is a *subsequence* of S_1 iff there exist $1 \leq j_1 < j_2 < \ldots < j_m \leq n$ s.t. $b_i = a_{j_i}$ for $1 \leq i \leq m$. If $j_i = j_{i+1} - 1$ for $1 \leq i < m$, then S_2 is a *contiguous* subsequence of S_1. We write $S_1 \cap S_2 \neq \emptyset$ iff S_1 and S_2 have a common element and write $e \in S_1$ iff e is an element appearing in S_1. The *concatenation* of S_1 and S_2, i.e., the sequence $\langle a_1, \ldots, a_n, b_1, \ldots, b_m \rangle$, is denoted by $S_1 \cdot S_2$. Finally, we use $|S_1|$ to denote the length of S_1, that is, the number of elements in S_1.

Definition 2.3 (Activity occurrence) *Let v be a video, ℓ a labeling of v, and $A = (V, E, \delta, \rho)$ a stochastic activity. An* occurrence *o of A in v w.r.t. ℓ is a sequence $\langle (f_1, s_1), \ldots, (f_m, s_m) \rangle$ such that*

- *$\langle f_1, \ldots, f_m \rangle$ is a subsequence of v,*
- *$\langle s_1, \ldots, s_m \rangle$ is an instance of A,*
- *$s_i \in \ell(f_i)$, for $1 \leq i \leq m$, and* [1]
- *$f_{i+1}.ts - f_i.ts \leq \delta(\langle s_i, s_{i+1} \rangle)$, for $1 \leq i < m$.*

The probability of o, denoted $p(o)$, is the probability of the instance $\langle s_1, \ldots, s_m \rangle$.

When concurrently monitoring multiple activities, shorter activity instances generally tend to have higher probability. To remedy this, we normalize occurrence probabilities by introducing the relative probability $p^*(o)$ of an occurrence o of activity A as $p^*(o) = \frac{p(o)}{p_{max}}$, where p_{max} is the highest probability of any instance of A.

[1] With a slight abuse of notation, we use s_i to refer to both node s_i and the action symbol labeling it.

Example 2.2 *Consider the video and the labeling of Example 2.1. An occurrence of the activity of Figure 1 is* $o = \langle (f_1, \textit{detectPerson}), (f_2, \textit{insertCard}), (f_3, \textit{insertCash}), (f_5, \textit{withdrawCard}) \rangle$*, and* $p^*(o) = 0.875$.

We use $\mathcal{O}(v, \ell)$ to denote the set of all activity occurrences in v w.r.t. ℓ. Whenever v and ℓ are clear from the context, we write $\mathcal{O}$ instead of $\mathcal{O}(v, \ell)$.

3 Unexplained Activity Probability Model

In this section we define the probability that a video sequence is unexplained, given a set $\mathcal{A}$ of known activities. We start by noting that the definition of probability of an activity occurring in a video can implicitly involve conflicts. For instance, consider the activity occurrence o in Example 2.2 and consider a second activity occurrence o' such that $(f_1, \textsf{detectPerson}) \in o'$. In this case, there is an implicit conflict because $(f_1, \textsf{detectPerson})$ belongs to both occurrences, but in fact, detectPerson can only belong to one activity occurrence, i.e. though o and o' may both have a non-zero probability of occurrence, the probability that these two activity occurrences coexist is 0. Formally, we say two activity occurrences o, o' *conflict*, denoted $o \nsim o'$, iff $o \cap o' \neq \emptyset$. We now use this to define possible worlds.

Definition 3.1 (Possible world) *A* possible world *for a video v and a labeling ℓ is a subset w of $\mathcal{O}$ s.t. $\nexists o_i, o_j \in w, o_i \nsim o_j$.*

Thus, a possible world is a set of activity occurrences which do not conflict with one another, i.e., an action symbol in a frame cannot belong to two distinct activity occurrences in the same world. We use $\mathcal{W}(v, \ell)$ to denote the set of all possible worlds for a video v and a labeling ℓ; whenever v and ℓ are clear from the context, we simply write $\mathcal{W}$.

Example 3.1 *Consider a video with two conflicting occurrences o_1, o_2. There are 3 possible worlds: $w_0 = \emptyset$, $w_1 = \{o_1\}$, and $w_2 = \{o_2\}$. Note that $\{o_1, o_2\}$ is not a world as $o_1 \nsim o_2$. Each world represents a way of explaining what is observed. The first world corresponds to the case where nothing is explained, the second and third worlds correspond to the scenarios where we use one of the two possible occurrences to explain the observed action symbols.*

Note that any subset of $\mathcal{O}$ not containing conflicting occurrences is a legitimate possible world — possible worlds are not required to be maximal w.r.t. $\subseteq$. In the above example, the empty set is a possible world even though there are two other possible worlds $w_1 = \{o_1\}$ and $w_2 = \{o_2\}$ which are supersets of it. The reason is that o_1 and o_2 are uncertain, so the scenario where neither o_1 nor o_2 occurs is a legitimate one. We further illustrate this point below.

Example 3.2 *Suppose we have a video where a single occurrence o has $p^*(o) = 0.6$. In this case, it is natural to say that there are two possible worlds $w_0 = \emptyset$ and $w_1 = \{o\}$ and expect the probabilities of w_0 and w_1 to be 0.4 and 0.6, respectively. By restricting ourselves to maximal possible worlds only, we would have only one possible world, w_1, whose probability is 1, which is wrong. Nevertheless, if $p^*(o) = 1$, w_1 is the only possible scenario. This can be achieved by assigning 0 and 1 to the probabilities of w_0 and w_1, respectively.*

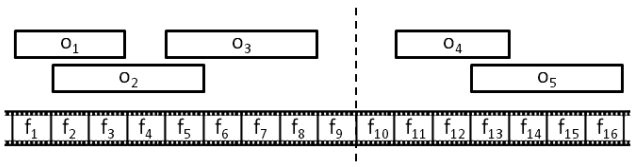


Figure 2: Conflict-Based Partitioning of a video

We use $\overset{*}{\nsim}$ to denote the transitive closure of $\nsim$. Clearly, $\overset{*}{\nsim}$ is an equivalence relation and determines a partition of $\mathcal{O}$ into equivalence classes $\mathcal{O}_1, \ldots, \mathcal{O}_m$.

Example 3.3 *Suppose we have a video $v = \langle f_1, \ldots, f_{16} \rangle$ and a labeling ℓ such that five occurrences o_1, o_2, o_3, o_4, o_5 are detected as depicted in Figure 2, that is, $o_1 \nsim o_2$, $o_2 \nsim o_3$, and $o_4 \nsim o_5$. There are two equivalence classes determined by $\overset{*}{\nsim}$, namely $\mathcal{O}_1 = \{o_1, o_2, o_3\}$ and $\mathcal{O}_2 = \{o_4, o_5\}$.*

The equivalence classes determined by $\overset{*}{\nsim}$ lead to a conflict-based partitioning of a video.

Definition 3.2 (Conflict-Based Partitioning) *Let v be a video, ℓ a labeling, and $\mathcal{O}_1, \ldots, \mathcal{O}_m$ the equivalence classes determined by $\overset{*}{\nsim}$. A* Conflict-Based Partitioning *(CBP) of v (w.r.t. ℓ) is a sequence $\langle (v_1, \ell_1), \ldots, (v_m, \ell_m) \rangle$ such that:*

- *$v_1 \cdot \ldots \cdot v_m = v$;*
- *ℓ_i is the restriction of ℓ to v_i, i.e., it is a labeling of v_i s.t. $\forall f \in v_i, \ell_i(f) = \ell(f)$, for $1 \leq i \leq m$; and*
- *$\mathcal{O}(v_i, \ell_i) = \mathcal{O}_i$, for $1 \leq i \leq m$.*

The v_i's are called segments.

Example 3.4 *A CBP of the video in Example 3.3 is $\langle (v_1, \ell_1), (v_2, \ell_2) \rangle$, where $v_1 = \langle f_1, \ldots, f_9 \rangle$, $v_2 = \langle f_{10}, \ldots, f_{16} \rangle$, ℓ_1 and ℓ_2 are the restrictions of ℓ to v_1 and v_2, respectively. Another partitioning of the same video is $v_1 = \langle f_1, \ldots, f_{10} \rangle$ and $v_2 = \langle f_{11}, \ldots, f_{16} \rangle$.*

Thus, activity occurrences determine a set of possible worlds (intuitively, different ways of explaining what is in the video). We wish to find a probability distribution over all possible worlds that (i) is consistent with the relative probabilities of the occurrences, and (ii) takes conflicts into account. We assume the user specifies a function $Weight : \mathcal{A} \rightarrow \mathbb{R}^+$ which assigns a weight to each activity and prioritizes the importance of the activity for his needs. The weight of an occurrence o of activity A is the weight of A. We use $C(o)$ to denote the set of occurrences conflicting with o, i.e., $C(o) = \{o' \mid o' \in \mathcal{O} \wedge o' \nsim o\}$. Note that $C(o)$ includes o itself and $C(o) = \{o\}$ when o does not conflict with any other occurrence. Finally, we assume that activity occurrences belonging to different segments are independent events. Suppose p_i denotes the (unknown) probability of world w_i. As we know the probability of occurrences, and as each occurrence occurs in certain worlds, we can induce a set of nonlinear constraints that are used to learn the values of the p_i's.

Definition 3.3 *Let v be a video, ℓ a labeling, and $\mathcal{O}_1, \ldots, \mathcal{O}_m$ the equivalence classes determined by $\overset{*}{\nsim}$. We*

define the non-linear constraints $NLC(v, \ell)$ as follows:

$$\begin{cases} p_i \geq 0, \quad \forall w_i \in \mathcal{W} \\ \sum_{w_i \in \mathcal{W}} p_i = 1 \\ \sum_{w_i \in \mathcal{W} \text{ s.t. } o \in w_i} p_i = p^*(o) \cdot \frac{Weight(o)}{\sum_{o_j \in C(o)} Weight(o_j)}, \forall o \in \mathcal{O} \\ p_j = \prod_{k=1}^{m} \sum_{w_i \in \mathcal{W} \text{ s.t. } w_i \cap \mathcal{O}_k = w_j \cap \mathcal{O}_k} p_i \qquad \forall w_j \in \mathcal{W} \end{cases}$$

The first two types of constraints enforce a probability distribution over the set of possible worlds. The third type of constraint ensures that the probability of occurrence o – which is the sum of the probabilities of the worlds containing o – is equal to its relative probability $p^*(o)$ weighted by $\frac{Weight(o)}{\sum_{o_j \in C(o)} Weight(o_j)}$, the latter being the weight of o divided by the sum of the weights of the occurrences conflicting with o. Note that: (i) the value on the right-hand side of the third type of constraint decreases as the amount of conflict increases, (ii) if an occurrence o is not conflicting with any other occurrence, then its probability $\sum_{w_i \in \mathcal{W} \text{ s.t. } o \in w_i} p_i$ is equal to $p^*(o)$, namely the probability returned by the stochastic automaton. The last kind of constraint reflects the independence between segments.

In general $NLC(v, \ell)$ might admit multiple solutions.

Example 3.5 *Consider a single-segment video consisting of frames $f_1, \ldots, f_9$ shown in Figure 2. Suppose three occurrences o_1, o_2, o_3 have been detected with relative probability 0.3, 0.6, and 0.5 respectively. Suppose the weights of o_1, o_2, o_3 are 1, 2, 3, respectively. Five worlds are possible in this case: $w_0 = \emptyset$, $w_1 = \{o_1\}$, $w_2 = \{o_2\}$, $w_3 = \{o_3\}$, and $w_4 = \{o_1, o_3\}$. Then, $NLC(v, \ell)$ is as follows (we omit the constraints expressing independence among segments because, in this case, they are trivial):*

$$\begin{aligned} &p_0 + p_1 + p_2 + p_3 + p_4 = 1 \\ &p_1 + p_4 = 0.3 \cdot \tfrac{1}{3} \\ &p_2 = 0.6 \cdot \tfrac{1}{3} \\ &p_3 + p_4 = 0.5 \cdot \tfrac{3}{5} \end{aligned}$$

which has multiple solutions. One solution is $p_0 = 0.4$, $p_1 = 0.1$, $p_2 = 0.2$, $p_3 = 0.3$, $p_4 = 0$. Another solution is $p_0 = 0.5$, $p_1 = 0$, $p_2 = 0.2$, $p_3 = 0.2$, $p_4 = 0.1$.

In the rest of the paper, we assume that $NLC(v, \ell)$ is solvable. We say that a sequence $S = \langle (f_1, s_1), \ldots, (f_n, s_n) \rangle$ *occurs in* a video v w.r.t. a labeling ℓ iff $\langle f_1, \ldots, f_n \rangle$ is a contiguous subsequence of v and $s_i \in \ell(f_i)$ for $1 \leq i \leq n$. We give two semantics for S to be unexplained in a world w:

1. S is *totally unexplained* in w, denoted $w \not\models_T S$, iff $\forall (f_i, s_i) \in S, \nexists o \in w, (f_i, s_i) \in o$;
2. S is *partially unexplained* in w, denoted $w \not\models_P S$, iff $\exists (f_i, s_i) \in S, \nexists o \in w, (f_i, s_i) \in o$.

Intuitively, S is totally (resp. partially) unexplained in w iff w does not explain every (resp. at least one) symbol of S. We now define the probability that a sequence occurring in a video is totally or partially unexplained.

Definition 3.4 *Let v be a video, ℓ a labeling, and S a sequence occurring in v w.r.t. ℓ. The probability interval that S is totally unexplained in v w.r.t. ℓ is $\mathcal{I}_T(S) = [l, u]$, where:*

$$\begin{aligned} l = \textbf{minimize} & \textstyle\sum_{w_i \in \mathcal{W} \text{ s.t. } w_i \not\models_T S} p_i \\ \textbf{subject to } & NLC(v, \ell) \\ u = \textbf{maximize} & \textstyle\sum_{w_i \in \mathcal{W} \text{ s.t. } w_i \not\models_T S} p_i \\ \textbf{subject to } & NLC(v, \ell) \end{aligned}$$

Likewise, the probability interval that S is partially unexplained in v w.r.t. ℓ is $\mathcal{I}_P(S) = [l', u']$, where:

$$\begin{aligned} l' = \textbf{minimize} & \textstyle\sum_{w_i \in \mathcal{W} \text{ s.t. } w_i \not\models_P S} p_i \\ \textbf{subject to } & NLC(v, \ell) \\ u' = \textbf{maximize} & \textstyle\sum_{w_i \in \mathcal{W} \text{ s.t. } w_i \not\models_P S} p_i \\ \textbf{subject to } & NLC(v, \ell) \end{aligned}$$

Thus, given a solution of $NLC(v, \ell)$, the probability that a sequence S occurring in v is totally (resp. partially) unexplained is the sum of the probabilities of the worlds in which S is totally (resp. partially) unexplained. As $NLC(v, \ell)$ may have multiple solutions, we find the tightest interval $[l, u]$ (resp. $[l', u']$) s.t. this probability is in $[l, u]$ (resp. $[l', u']$) for any solution. Different criteria can be used to infer a value from an interval $[l, u]$, e.g. the MIN l, the MAX u, the average (i.e., $(l + u)/2$), etc. Clearly, the only requirement is that this value has to be in $[l, u]$. In the rest of the paper we assume that one of the above criteria has been chosen — $\mathcal{P}_T(S)$ (resp. $\mathcal{P}_P(S)$) denotes the probability that S is totally (resp. partially) unexplained.

Given two sequences S_1 and S_2 occurring in a video, it can be easily verified that if S_1 is a subsequence of S_2, then $\mathcal{P}_T(S_1) \geq \mathcal{P}_T(S_2)$ and $\mathcal{P}_P(S_1) \leq \mathcal{P}_P(S_2)$.

Definition 3.5 (Unexplained activity occurrence) *Let v be a video, ℓ a labeling, $\tau \in [0, 1]$ a probability threshold, and $L \in \mathbb{N}^+$ a length threshold. Then,*

- *a* totally unexplained activity occurrence *is a sequence S occurring in v s.t. (i) $\mathcal{P}_T(S) \geq \tau$, (ii) $|S| \geq L$, and (iii) S is maximal, i.e., there does not exist a sequence $S' \neq S$ occurring in v s.t. S is a subsequence of S', $\mathcal{P}_T(S') \geq \tau$, and $|S'| \geq L$.*
- *a* partially unexplained activity occurrence *is a sequence S occurring in v s.t. (i) $\mathcal{P}_P(S) \geq \tau$, (ii) $|S| \geq L$, and (iii) S is minimal, i.e., there does not exist a sequence $S' \neq S$ occurring in v s.t. S' is a subsequence of S, $\mathcal{P}_P(S') \geq \tau$, and $|S'| \geq L$.*

In the definition above, L is the minimum length a sequence must be for it to be considered a possible unexplained activity occurrence. Totally unexplained occurrences S have to be maximal because once we find S, then any sub-sequence of it has a probability of being (totally) unexplained greater than or equal to the probability of S. On the other hand, partially unexplained occurrences S' have to be minimal because once we find S', then any super-sequence of it has a probability of being (partially) unexplained greater than or equal to the one of S.

Intuitively, an unexplained activity occurrence is a sequence of action symbols that are observed in the video and poorly explained by the known activity models. Such sequences might correspond to unknown variants of known activities or to entirely new – and unknown – activities.

An *Unexplained Activity Problem* (UAP) instance is a 4-tuple $\langle v, \ell, \tau, L\rangle$, where v is a video, ℓ is a labeling of v, $\tau \in [0,1]$ is a probability threshold, and $L \in \mathbb{N}^+$ is a length threshold. The desired result is all totally/partially unexplained activity occurrences.

4 Properties of UAPs

In this section, we derive properties of the above model that can be leveraged (in the next section) to devise efficient algorithms to find unexplained activities. Specifically, we first show an interesting property concerning the resolution of $NLC(v,\ell)$ (some subsequent results rely on it); then, in the following two subsections, we consider specific properties for totally and partially unexplained activities.

For a given video v and labeling ℓ, we now show that if $\langle (v_1,\ell_1),\ldots,(v_m,\ell_m)\rangle$ is a CBP, then we can find the solutions of the *non-linear* constraints $NLC(v,\ell)$ by solving m *smaller sets of linear constraints.*[2] We define $LC(v,\ell)$ as the set of linear constraints of $NLC(v,\ell)$ (thus, we include all the constraints of Definition 3.3 except for the last kind). Henceforth, we use $\mathcal{W}$ to denote $\mathcal{W}(v,l)$ and $\mathcal{W}_i$ to denote $\mathcal{W}(v_i,\ell_i)$, $1 \le i \le m$. A solution of $NLC(v,\ell)$ is a mapping $\mathcal{P} : \mathcal{W} \to [0,1]$ which satisfies $NLC(v,\ell)$. Likewise, a solution of $LC(v_i,\ell_i)$ is a mapping $\mathcal{P}_i : \mathcal{W}_i \to [0,1]$ which satisfies $LC(v_i,\ell_i)$. It is important to note that $\mathcal{W} = \{w_1 \cup \ldots \cup w_m \mid w_i \in \mathcal{W}_i, 1 \le i \le m\}$.

Theorem 1 *Let v be a video, ℓ a labeling, and $\langle (v_1,\ell_1),\ldots,(v_m,\ell_m)\rangle$ a CBP. $\mathcal{P}$ is a solution of $NLC(v,\ell)$ iff $\forall i \in [1,m]$ there exists a solution $\mathcal{P}_i$ of $LC(v_i,\ell_i)$ s.t. $\mathcal{P}(\bigcup_{i=1}^{m} w_i) = \prod_{i=1}^{m} \mathcal{P}_i(w_i)$ for every $w_1 \in \mathcal{W}_1,\ldots,w_m \in \mathcal{W}_m$.*

Consider a video v and a labeling ℓ, and let $\langle (v_1,\ell_1),\ldots,(v_m,\ell_m)\rangle$ be a CBP. Given a sequence $S = \langle (f_1,s_1),\ldots,(f_q,s_q)\rangle$ occurring in v, we say that $v_i, v_{i+1},\ldots,v_{i+n}$ $(1 \le i \le i+n \le m)$ are the sub-videos containing S iff $f_1 \in v_i$ and $f_q \in v_{i+n}$. In other words, S spans the sub-videos $v_i, v_{i+1},\ldots,v_{i+n}$: it starts at some point in sub-video v_i (as v_i contains the first frame of S) and ends at some point in sub-video v_{i+n} (as v_{i+n} contains the last frame of S). In addition, we use S_k to denote the *projection* of S on the k-th sub-video v_k $(i \le k \le i+n)$, that is, the subsequence of S containing all the pairs $(f,s) \in S$ with $f \in v_k$.

4.1 Totally unexplained activities

The following theorem says that we can compute $\mathcal{I}_T(S)$ by solving LC (which are linear constraints) for each sub-video containing S (instead of solving a non-linear set of constraints for the whole video).

Theorem 2 *Consider a video v and a labeling ℓ. Let $\langle (v_1,\ell_1),\ldots,(v_m,\ell_m)\rangle$ be a CBP and $\langle v_i,\ldots,v_{i+n}\rangle$ be the sub-videos containing a sequence S occurring in v. For $i \le k \le i+n$, let*

$$l_k = \textbf{minimize} \sum_{w_h \in \mathcal{W}_k \ s.t.\ w_h \not\models_T S_k} p_h \quad \textbf{subject to } LC(v_k,\ell_k)$$
$$u_k = \textbf{maximize} \sum_{w_h \in \mathcal{W}_k \ s.t.\ w_h \not\models_T S_k} p_h \quad \textbf{subject to } LC(v_k,\ell_k)$$

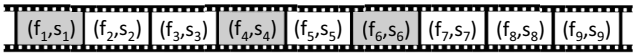
If $\mathcal{I}_T(S) = [l,u]$, then $l = \prod_{k=i}^{i+n} l_k$ and $u = \prod_{k=i}^{i+n} u_k$.

The following theorem provides a sufficient condition for a pair (f,s) not to be included in any sequence S occurring in v and having $\mathcal{P}_T(S) \ge \tau$.

Theorem 3 *Let $\langle v,\ell,\tau,L\rangle$ be a UAP instance. Given (f,s) s.t. $f \in v$ and $s \in \ell(f)$, let $\varepsilon = \sum_{o \in \mathcal{O} \ s.t.\ (f,s) \in o} p^*(o) \cdot \frac{Weight(o)}{\sum_{o_j \in C(o)} Weight(o_j)}$. If $\varepsilon > 1-\tau$, then there does not exist a sequence S occurring in v s.t. $(f,s) \in S$ and $\mathcal{P}_T(S) \ge \tau$.*

If the condition stated in the theorem above holds for a pair (f,s), then we say that (f,s) is *sufficiently explained. It is important to note that to check whether a pair (f,s) is sufficiently explained, we do not need to solve any set of linear or non-linear constraints, since ε is computed by simply summing the (weighted) probabilities of the occurrences containing (f,s).* Thus, this result yields a further efficiency. A frame f is *sufficiently explained* iff (f,s) is sufficiently explained for every $s \in \ell(f)$. If (f,s) is sufficiently explained, then it can be disregarded for the purpose of identifying unexplained activity occurrences, and, in addition, this may allow us to disregard entire parts of videos as shown in the example below.

Example 4.1 *Consider a UAP instance $\langle v,\ell,\tau,L\rangle$ where $v = \langle f_1,\ldots,f_9\rangle$ and ℓ is s.t. $\ell(f_i) = \{s_i\}$ for $1 \le i \le 9$, as depicted in the figure below.*



Suppose that $L = 3$ and that (f_1,s_1), (f_4,s_4), (f_6,s_6) are sufficiently explained, that is, because of Theorem 3, we can conclude that there is no sequence S occurring in v with $\mathcal{P}_T(S) \ge \tau$ and containing any of them. Even though we have been able to apply the theorem to a few (f_i,s_i) pairs, we can conclude that no unexplained activity occurrence can be found before f_7, because $L = 3$.

Given a UAP instance $I = \langle v,\ell,\tau,L\rangle$ and a contiguous subsequence v' of v, v' is *relevant* iff (i) $|v'| \ge L$, (ii) $\forall f \in v'$, f is not sufficiently explained, and (iii) v' is maximal (i.e., there does not exist $v'' \ne v'$ s.t. v' is a subsequence of v'' and v'' satisfies (i) and (ii)). We use $relevant(I)$ to denote the set of relevant sub-videos.

Theorem 3 entails that relevant sub-videos can be individually considered when looking for totally unexplained activities because there is no totally unexplained activity spanning two different relevant sub-videos.

[2] This therefore yields two benefits: first it allows us to solve a smaller set of constraints, and second, it allows us to solve linear constraints which are usually easier to solve than nonlinear ones.

4.2 Partially unexplained activities

The following theorem states that we can compute $\mathcal{I}_P(S)$ by solving NLC for the sub-video consisting of the segments containing S (instead of solving NLC for the whole video).

Theorem 4 *Consider a video v and a labeling ℓ. Let $\langle (v_1, \ell_1), \ldots, (v_m, \ell_m) \rangle$ be a CBP and $\langle v_i, \ldots, v_{i+n} \rangle$ be the sub-videos containing a sequence S occurring in v. Let $v^* = v_i \cdot \ldots \cdot v_{i+n}$ and ℓ^* be a labeling for v^* s.t., for every $f \in v^*$, $\ell^*(f) = \ell(f)$. $\mathcal{I}_P(S)$ computed w.r.t. v and ℓ is equal to $\mathcal{I}_P(S)$ computed w.r.t. v^* and ℓ^*.*

5 Algorithms

We now present algorithms to find totally and partially unexplained activities. Due to lack of space, we assume $|\ell(f)| = 1$ for every frame f in a video (this makes the algorithms much more concise – generalization to the case of multiple action symbols per frame is straightforward[3]). Given a video $v = \{f_1, \ldots, f_n\}$, we use $v(i, j)$ $(1 \leq i \leq j \leq n)$ to denote the sequence $S = \langle (f_i, s_i), \ldots, (f_j, s_j) \rangle$, where s_k is the only element in $\ell(f_k)$, $i \leq k \leq j$.

The FindTUA algorithm computes all totally unexplained activities in a video. Leveraging Theorem 3, FindTUA only considers relevant subsequences of v. When the algorithm finds a sequence $v'(start, end)$ of length at least L having a probability of being unexplained greater than or equal to τ (line 5), then the algorithm makes it maximal by adding frames on the right. Instead of adding one frame at a time, $v'(start, end)$ is extended of L frames at a time until its probability drops below τ (lines 7–10); then, the exact maximum length of the unexplained activity is found (line 12, this is accomplished by performing a binary search between s and e). Note that $\mathcal{P}_T$ is computed by applying Theorem 2.

Algorithm 1 FindTUA

Input: UAP instance $I = \langle v, \ell, \tau, L \rangle$
Output: Set of totally unexplained activities

```
1: Sol = ∅
2: for all v' ∈ relevant(I) do
3:     start = 1; end = L
4:     repeat
5:         if P_T(v'(start, end)) ≥ τ then
6:             end' = end
7:             while end < |v'| do
8:                 end = min{end + L, |v'|}
9:                 if P_T(v'(start, end)) < τ then
10:                    break
11:            s = max{end − L, end'}; e = end
12:            end = max{mid | s ≤ mid ≤ e ∧ P_T(v'(start, mid)) ≥ τ}
13:            S = v'(start, end); Add S to Sol;
14:            start = start + 1; end = start + |S| − 1
15:        else
16:            start = start + 1; end = max{end, start + L − 1}
17:    until end > |v'|
18: return Sol
```

Theorem 5 *Algorithm FindTUA returns all the totally unexplained activities of the input instance.*

The FindPUA algorithm below computes all partially unexplained activities. To find an unexplained activity, it starts with a sequence of a certain length (at least L) and adds frames on the right of the sequence until its probability of being unexplained is greater than or equal to τ. As in the case of FindTUA, this is done not by adding one frame at a time, but adding L frames at a time (lines 5–8) and then determining the exact minimal length (line 11, this is accomplished by performing a binary search between s and e). The sequence is then shortened on the left making it minimal (line 15, this is again done by performing a binary search instead of proceeding one frame at a time). Note that $\mathcal{P}_P$ is computed by applying Theorem 4.

Algorithm 2 FindPUA

Input: UAP instance $I = \langle v, \ell, \tau, L \rangle$
Output: Set of partially unexplained activities

```
1: Sol = ∅; start = 1; end = L
2: while end ≤ |v| do
3:     if P_P(v(start, end)) < τ then
4:         end' = end
5:         while end < |v| do
6:             end = min{end + L, |v|}
7:             if P_P(v(start, end)) ≥ τ then
8:                 break
9:         if P_P(v(start, end)) ≥ τ then
10:            s = max{end' + 1, end − L + 1}; e = end
11:            end = min{mid | s ≤ mid ≤ e ∧ P_P(v(start, mid)) ≥ τ}
12:        else
13:            return Sol
14:    s' = start; e' = end − L + 1
15:    start = max{mid | s' ≤ mid ≤ e' ∧ P_P(v(mid, end)) ≥ τ}
16:    S = v(start, end); Add S to Sol
17:    start = start + 1; end = start + |S| − 1
18: return Sol
```

Theorem 6 *Algorithm FindPUA returns all the partially unexplained activities of the input instance.*

6 Experimental Results

Our prototype implementation of the proposed framework consists of:

- an *image processing library*, which performs low-level processing of video frames, including object tracking and classification;
- a *video labeler*, which maps frames to action symbols, based on the output of the image processing stage;
- an *activity recognition algorithm*, based on [Albanese *et al.*, 2007], which identifies all possible occurrences of known activities;
- a *UAP engine*, which implements algorithms FindTUA and FindPUA in 7500 lines of Java code.

We generated a video by concatenating multiple videos from the ITEA CANDELA dataset, a publicly available dataset depicting a number of staged package exchanges and object drop-offs and pick-ups (`http://www.multitel.be/~va/candela/abandon.html`). We evaluated precision and recall against a ground truth provided by human annotators. Annotators were informed about known activities by providing them with a graphical representation of the activity models (for an example, see Figure 1). They were asked to watch the video and identify video segments where totally (resp. partially) unexplained activities occurred.

[3] Indeed, it suffices to consider the different sequences given by the different action symbols.

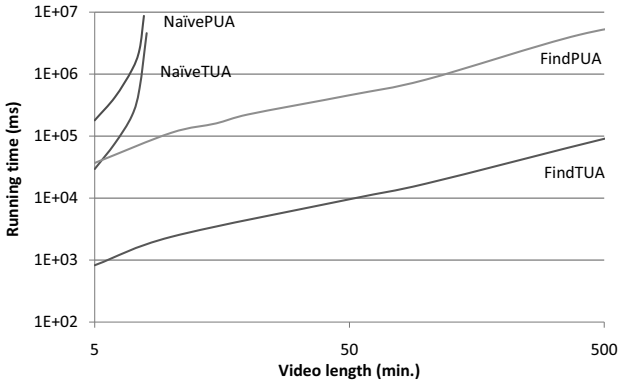


Figure 3: Processing times

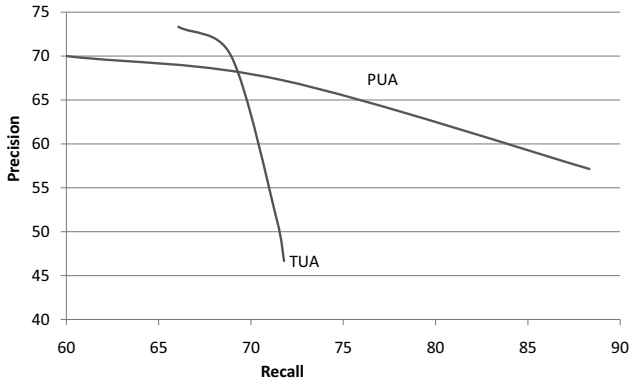


Figure 4: Precision and recall

Running time. Figure 3 shows the processing time of FindTUA and FindPUA as a function of the length of the video. Note that both axes are on a logarithmic scale. It is clear that both algorithms run in time linear in the length of the video, and significantly outperform naïve algorithms that do not use the optimizations of Theorems 2, 3, and 4.

Precision/recall. Given a set $\mathcal{A}$ of activity definitions, let $\{S_i^a\}_{i\in[1,m]}$ denote the set of unexplained sequences returned by our algorithms, and let $\{S_j^h\}_{j\in[1,n]}$ denote the set of sequences flagged as *unexplained* by human annotators. We evaluate precision and recall as

$$P = \frac{|\{S_i^a | \exists S_j^h \; s.t. \; S_i^a \approx_p S_j^h\}|}{m}$$
$$R = \frac{|\{S_j^h | \exists S_i^a \; s.t. \; S_i^a \approx_p S_j^h\}|}{n}$$

where $S_i^a \approx_p S_j^h$ means that S_i^a and S_j^h overlap by a percentage no smaller than p.

The precision/recall graph for algorithms FindTUA and FindPUA, when $p = 75\%$, is reported in Figure 4. The value of the probability threshold τ that maximizes the F-measure for FindTUA and FindPUA is 0.6. For $\tau = 0.6$, FindTUA achieves $P = 70\%$ and $R = 69\%$, whereas FindPUA achieves $P = 67\%$ and $R = 72\%$.

7 Conclusions

Suppose $\mathcal{A}$ is a given set of known activity models and v is a video. This paper presents a possible worlds framework to find all subsequences of v that cannot be explained by an activity in $\mathcal{A}$ with a probability exceeding a user-specified threshold. We develop the FindTUA and FindPUA algorithms to find totally and partially unexplained sequences respectively.

Finally, we present a prototype implementation of the proposed framework, and show, through experiments, that algorithms FindTUA and FindPUA work well in practice on a real world video data set.

References

[Albanese *et al.*, 2007] M. Albanese, V. Moscato, A. Picariello, V. S. Subrahmanian, and O. Udrea. Detecting stochastically scheduled activities in video. In *Proc. of IJCAI'07*, pages 1802–1807, 2007.

[Cuntoor *et al.*, 2008] N. P. Cuntoor, B. Yegnanarayana, and R. Chellappa. Activity modeling using event probability sequences. *IEEE Trans. Image Processing*, 17(4):594–607, April 2008.

[Hamid *et al.*, 2003] R. Hamid, Y. Huang, and I. Essa. Argmode - activity recognition using graphical models. In *Proc. IEEE CVPR'03*, volume 4, pages 38–43, 2003.

[Hongeng and Nevatia, 2001] Somboon Hongeng and Ramakant Nevatia. Multi-agent event recognition. In *Proc. of IEEE ICCV'01*, volume 2, pages 84–93, 2001.

[Hongeng *et al.*, 2004] Somboon Hongeng, Ramakant Nevatia, and François Brémond. Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, 96(2):129–162, 2004.

[Oliver *et al.*, 2002] N. Oliver, E. Horvitz, and A. Garg. Layered representations for human activity recognition. In *Proc. of IEEE ICMI'02*, pages 3–7, 2002.

[Vaswani *et al.*, 2005] N. Vaswani, A. K. Roy-Chowdhury, and R. Chellappa. Shape activity: A continuous-state hmm for moving/deforming shapes with application to abnormal activity detection. *IEEE Trans. Image Processing*, 14(10):1603–1616, October 2005.

[Wilson and Bobick, 1999] A. D. Wilson and A. F. Bobick. Parametric hidden Markov models for gesture recognition. *IEEE Trans. Pattern Anal. Machine Intell.*, 21(9):884–900, September 1999.

[Zhong *et al.*, 2004] Hua Zhong, Jianbo Shi, and Mirkó Visontai. Detecting unusual activity in video. In *Proc. of IEEE CVPR'04*, volume 2, pages 819–826, 2004.