

Detecting Stochastically Scheduled Activities in Video *

Massimiliano Albanese¹ Vincenzo Moscato² Antonio Picariello² V.S. Subrahmanian¹

Octavian Udrea¹

¹University of Maryland Institute for Advanced Computer Studies,
{albanese,vs,udrea@umiacs.umd.edu}

²Università di Napoli Federico II, Napoli, Italy,
{vmoscato,picus}@unina.it

Abstract

The ability to automatically detect activities in video is of increasing importance in applications such as bank security, airport tarmac security, baggage area security and building site surveillance. We present a stochastic activity model composed of atomic actions which are directly observable through image understanding primitives. We focus on answering two types of questions: (i) what are the minimal sub-videos in which a given action is identified with probability above a certain threshold and (ii) for a given video, can we decide which activity from a given set most likely occurred? We provide the MPS algorithm for the first problem, as well as two different algorithms (naiveMPA and MPA) to solve the second. Our experimental results on a dataset consisting of staged bank robbery videos (described in [Vu *et al.*, 2003]) show that our algorithms are both fast and provide high quality results when compared to human reviewers.

1 Introduction

There has been a tremendous amount of work in low-level computer vision video processing algorithms for detection of various types of elementary events or actions such as a person entering or leaving a room or unloading a package. However, there is much less work on how such low level detection algorithms can be utilized in high-level recognition of activities in video. High level activities can include transactions at a bank automatic teller machine (ATM), robberies at an ATM, unloading an aircraft, moving bags from a secure baggage zone at an airport to the truck that transports it to the plane, etc.

In this paper, we focus on such high level activities (and provide a simple ATM example to motivate our work). High level activities consist of a set of simple actions that are detectable using video image processing algorithms, together with some temporal requirements.

*Authors listed in alphabetical order. This work was partly supported by AFOSR grants FA95500610405 and FA95500510298, ARO grant DAAD190310202 and by the Joint Institute for Knowledge Discovery.

In this paper, we provide fast algorithms to answer two kinds of queries. The first type of query tries to find *minimal* (i.e. the shortest possible) clips of video that contain a given event with a probability exceeding a specified threshold — we call these *Threshold Activity Queries*. The second type of query takes a portion of a video (the part seen thus far) and a set of activities, and tries to find the activity in the set that most likely has occurred in the video (we call these *Activity Recognition Queries*).

The key contributions of this paper are the following. We first introduce (in Section 2) a simple stochastic automata representation of complex activities based on atomic actions which are recognizable by an image understanding primitive such as those in [Elgammal *et al.*, 2000; Bevilacqua, 2003; 2002; Cavallaro *et al.*, 2005; 2000; Makarov, 1996]. Examples of actions we have implemented in our system using computer vision algorithms include person and object identification, identification of people entering and exiting rooms and motion tracking. Our first major contribution (in Section 3) is the *MPS* algorithm to answer threshold queries. Our second major contribution (Section 4) are the *naiveMPA* and *MPA* algorithms to solve activity recognition queries. We evaluate these algorithms theoretically as well as experimentally (in Section 5) on a third party dataset consisting of staged bank robbery videos [Vu *et al.*, 2003] and provide evidence that the algorithms are both efficient and yield high-quality results.

2 Stochastic Activities

In this section, we provide a simple framework to represent high level activities on top of low level image processing primitives. This framework is a simple version of HMMs - we don't claim it is new. The novel part of this paper starts from the next section onwards.

We assume the existence of a finite set \mathcal{A} of “action” symbols; we assume that each such atomic action can be detected by an image understanding method¹. For the purposes of this paper, we will assume that action symbols are propositional, though there is no loss of generality in this assumption.

¹Although some of the image understanding methods cited in this paper return a numeric value between 0 and 1, this can be readily transformed into a method that returns either “yes” or “no” based on a fixed threshold.

Definition 1 (Stochastic activity) A stochastic activity is a labeled graph (V, E, ρ) where:

1. V is a finite set of action symbols;
2. E is a subset of $(V \times V)$;
3. ρ is a function that associates, with each v of out-degree 1 or more, a probability distribution on $\{(v, v') | (v, v') \in E\}$, i.e. for all $v \in V$, $\sum_{(v, v') \in E} \rho((v, v')) = 1$.
4. $\{v \in V | \nexists v' \in V \text{ s.t. } (v, v') \in E\} \neq \emptyset$. This states that there exists at least one end symbol in the activity definition.
5. $\{v \in V | \nexists v' \in V \text{ s.t. } (v', v) \in E\} \neq \emptyset$. This states that there exists at least one start symbol in the activity definition.

For $v \in V$, we denote by $p_b(v)$ the maximum product of probabilities on any paths between v and an end node.

Figure 1(a) shows a small example of a stochastic activity associated with transactions at an Automatic Teller Machine (ATM). In this figure, we have assumed that edges without a label have probability 1. For example, consider the node **withdraw-cash**. There are two edges starting at this node - one to **withdraw-card** labeled with a probability of 0.9 and one to **insert-checks** with probability 0.1. This means that there is a 90% probability of transitioning to the **withdraw-card** state after executing **withdraw-cash** and a 10% probability of transitioning to **insert-checks** after executing **withdraw-cash**. For the purposes of this paper, this example is somewhat simplified (e.g., we avoided talking about receipts generated by ATMs, etc.). Each of the actions in this stochastic activity can be easily detected by either an image processing algorithm (e.g. **detect-person** would check if a person is present in the image) or a sensor (e.g. to detect if **insert-card** holds). It is also important to note that in this framework, it is possible to execute **withdraw-cash**, then execute **insert-checks**, and then again execute **withdraw-cash** — informally speaking, the probability of this sequence occurring would be $0.1 \times 0.2 = 0.02$. Similarly, Figure 1(b) describes an attempted robbery at an ATM.

A labeling ℓ of a video v is a mapping that takes a video frame f as input, and returns a set of action symbols as output.

Example 1 The following is a labeling of a 500-frame video w.r.t. to the stochastic activity presented in Figure 1(a): (5, {detect - person}), (26, {present - card}), (45, {insert - card}), (213, {withdraw - cash}), (320, {insert - checks}), (431, {withdraw - card}), (496, {¬detect - person}).

We say that an action a' follows an action a in segment $[s, e] \subseteq v$ w.r.t. labeling ℓ and activity definition (V, E, ρ) iff: (i) $a \in \ell(s)$, $a' \in \ell(e)$ and $(a, a') \in E$; (ii) $\forall a'' \in \{b \in V | (a, b) \in E\} - \{a'\}$ and $\forall f$ s.t. $s \leq f \leq e$, $a'' \notin \ell(f)$.

Example 2 Consider the labeling in Example 1. In the subvideo [45, 320], action *withdraw - cash* follows action *insert - card*. However, action *insert - checks* does not follow action *insert - card*.

Intuitively, a labeling of video v specifies which actions are detected in a given video. We now define the probability with

which a given video segment $[s, e]$ satisfies an activity specification.

Definition 2 (Probabilistic satisfaction) Let $v = [s, e]$ be a video segment and let (V, E, ρ) be a stochastic activity. Let $\ell : v \rightarrow 2^V$ be a labeling of the video segment v . We say that v satisfies (V, E, ρ) with probability p iff there exists a sequence of frames $f_1 \leq \dots \leq f_n \in [s, e]$ and a sequence of activity symbols u_1, \dots, u_n such that:

- (i) $\forall i \in [1, n] u_i \in \ell(f_i)$.
- (ii) For all $i \in [1, n - 1]$, u_{i+1} follows u_i in $[f_i, f_{i+1}]$ w.r.t. ℓ and (V, E, ρ) .
- (iii) $\{w \in V | (w, u_1) \in E\} = \emptyset$ and $\{w \in V | (u_n, w) \in E\} = \emptyset$.
- (iv) $\prod_{i=1}^{n-1} \rho((u_i, u_{i+1})) \geq p$.

One problem with the above definition is that an activity consisting of very few action symbols will in most cases yield a higher probability than an activity with a hundred times as many actions. In cases when we want to decide which activity from a given set most likely occurs in a video, we need to normalize the probability w.r.t. to the activity definition. We say that a video $[s, e]$ satisfies an activity (V, E, ρ) w.r.t. labeling ℓ with relative probability p^* iff $[s, e]$ satisfies (V, E, ρ) with probability p and $p^* = \frac{p - p_{min}}{p_{max} - p_{min}}$, where p_{min}, p_{max} are the lowest and respectively highest probabilities labeling a path from a start node² to an end node³ in (V, E, ρ) .

Example 3 Consider the labeling in Example 1 and the stochastic activity in Figure 1. Then the video satisfies the action with probability 0.056. For this example, $p_{min} = 0$ (since we can have an arbitrarily large sequence of activities repeating the action symbols *withdraw - cash*, *insert - checks*) and $p_{max} = 0.63$. Then the video whose labeling is given in Example 1 satisfies the activity in Figure 1 with a relative probability of approximately 0.089.

Proposition 1 Let $v = [s, e]$ be a video segment, and let ℓ be its labeling. Let (V, E, ρ) be a stochastic activity such that v satisfies (V, E, ρ) with probability p . Then for all videos v' with labeling ℓ' such that $v \subseteq v'$ and $\forall f \in [s, e] \ell(f) = \ell'(f)$, v' satisfies (V, E, ρ) with probability p .

Definition 3 A stochastic activity query (SAQ) is a four-tuple $(v, \ell, (V, E, \rho), p)$ where v is a video, ℓ is a labeling of the video, (V, E, ρ) is a stochastic activity, and p is a threshold probability.

A sequence $[s, e]$ is an answer to the above SAQ iff $[s, e]$ satisfies (V, E, ρ) with probability p and there is no strict subset $[s', e'] \subset [s, e]$ which satisfies (V, E, ρ) with probability p .

Intuitively, we are looking for the smallest subvideos that satisfy an activity definition with probability above a given threshold p . We will now define the most probable subvideo and most probable activity problems more formally.

The Most Probable Subvideo (MPS) Problem. The most probable subvideo problem can now be defined as follows: given a stochastic activity query $(v, \ell, (V, E, \rho), p)$, find the set of all answers to the query.

²A node of indegree 0.

³A node of outdegree 0.

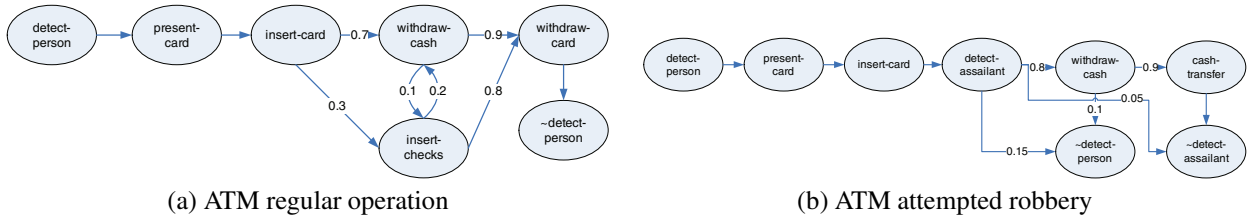


Figure 1: ATM stochastic activity examples

The Most Probable Activity (MPA) Problem. In many practical cases – airport security, ATM monitoring, etc., the system users are interested in monitoring a list of specific activities. The goal is to identify the activity (or activities) that most likely occur in a given portion of the surveillance video. The most likely activity problem can be defined as follows: given a video v , its labeling ℓ , a set of activity definitions $A = \{(V_1, E_1, \rho_1), \dots, (V_n, E_n, \rho_n)\}$, find the set $A' \subseteq A$ such that the following hold: (i) Let $p_m = \max\{p \in [0, 1] \mid \exists A_i = (V_i, E_i, \rho_i) \in A \text{ s.t. } v \text{ satisfies } A_i \text{ with relative probability } p\}$. Then $\forall (V_i, E_i, \rho_i) \in A'$, (V_i, E_i, ρ_i) satisfy v with relative probability p_m ; (ii) $\nexists (V_j, E_j, \rho_j) \in A - A'$, s.t. v satisfies (V_j, E_j, ρ_j) with relative probability p_m .

3 Threshold Queries

The most probable subvideo (MPS) algorithm tries to find all answers to a stochastic activity query $(v, \ell, (V, E, \rho), p)$ and is shown in Figure 2.

Algorithm MPS

Input: Query $(v, \ell, (V, E, \rho), p_t)$, where $v = [s, e]$.

Output: The set of subvideos representing the answer to the input query and the probabilities with which the activity is satisfied.

Notation: C is a set of triples (f, p, a) , where f is a frame in v , $p \in [0, 1]$ and $a \in V$ is an activity symbol.

```

1.  $R \leftarrow \emptyset$ ;
2.  $C \leftarrow \emptyset$ ;
3.  $V_s \leftarrow \{a \in V \mid \nexists a' \in V \text{ s.t. } (a', a) \in E\}$ ;
4.  $V_e \leftarrow \{a \in V \mid \nexists a' \in V \text{ s.t. } (a, a') \in E\}$ ;
5. for  $f_i = s$  to  $e$  do
6.   for every  $(f, p, a) \in C$  do
7.      $D \leftarrow \ell(f_i) \cap \{b \in V \mid (a, b) \in E\}$ ;
8.     if  $D \neq \emptyset$  then
9.        $C \leftarrow C - \{(f, p, a)\}$ ;
10.    for  $a' \in D$  do
11.      if  $p \cdot \rho(a, a') \geq p_t$  then
12.         $T \leftarrow \{(f', p', b) \in C \mid f' = f \wedge a' = b\}$ ;
13.         $C \leftarrow C - T \cup \{(f, \max(\max_{(f', p', a') \in T} (p'), p \cdot \rho(a, a')), a')\}$ ;
14.    endif
15.  endfor
16. endfor
17. for every  $a \in V_s \cap \ell(f_i)$  do
18.    $C \leftarrow C \cup \{(f_i, 1, a)\}$ ;
19. for every  $(f, p, a) \in C$  s.t.  $a \in V_e$  do
20.    $R \leftarrow R \cup \{(f, f_i, p)\}$ ;
21.    $C \leftarrow C - \{(f, p, a)\}$ ;
22. endfor
23. endfor
24. for  $([f_1, f_2], p) \in R$  do
25.   for  $([f'_1, f'_2], p')$  in  $R - \{([f_1, f_2], p)\}$  do
26.     if  $[f'_1, f'_2] \subset [f_1, f_2]$  then
27.        $R \leftarrow R - \{([f_1, f_2], p)\}$ ;
28.     else if  $([f'_1, f'_2] = [f_1, f_2]) \wedge (p' > p)$  then
29.        $R \leftarrow R - \{([f_1, f_2], p)\}$ ;
30.   endfor
31. endfor
32. return  $R$ ;

```

Figure 2: An algorithm for the MPS problem

The MPS algorithm maintains a set C of triples (f, p, a) that correspond to partially detected activities. f represents the frame when a start symbol (V, E, ρ) is detected; this is later used to determine the subvideos that should go in the answer. p represents the product of probabilities on the path from the start symbol up to the action symbol a – the last symbol detected up to the current frame. The algorithm works by changing the state represented by C – intuitively, consisting of all paths through the activity definition for which we may reach an end node with probability above the threshold. Triples are removed from C when it is evident that the current path will result in a probability lower than p_t (lines 12–13). When all frames of the video have been analyzed, the minimality condition in Definition 3 is enforced on lines 24–31 by removing redundant subvideos.

Consider the labeling in Example 1 and the activity definition in Figure 1(a) and assume that $p_t = 0.01$. At frame 5, the MPS algorithm will add the triple $(5, 1, \text{detect} - \text{person})$ to C (line 18). At frame 26, this triple will be replaced by $(5, 1, \text{present} - \text{card})$. Immediately after frame 213, we have $C = \{(5, 0.7, \text{withdraw} - \text{cash})\}$. The process continues until frame 496, after which the condition on line 19 becomes true and $(5, 0.056, \text{~detect} - \text{person})$ causes the subvideo $[5, 496]$ to be added to the answer R on line 20.

Theorem 1 (MPS correctness) *Algorithm MPS terminates and for an input query $(v, \ell, (V, E, \rho), p_t)$ it returns the set of answers R to the query – i.e.:*

- $\forall [s, e] \in R$, $[s, e]$ is an answer to $(v, \ell, (V, E, \rho), p_t)$ as per Definition 3.
- $\nexists [s, e]$ s.t. $[s, e]$ is an answer to $(v, \ell, (V, E, \rho), p_t)$ and $[s, e] \notin R$.

Theorem 2 (MPS complexity) *Algorithm MPS for input query $(v, \ell, (V, E, \rho), p_t)$ runs in time $\mathcal{O}(|v|^2 \cdot |V|)$.*

Note that MPS makes only one pass through the video, which allows it to process a video as it comes in frame by frame. MPS is therefore well suited for scenarios (such as surveillance) in which a video is analyzed while playing.

4 Activity recognition queries

In this section, we present two methods to answer activity recognition queries where:

- v is a video with labeling ℓ .
- $A = \{(V_1, E_1, \rho_1), \dots, (V_n, E_n, \rho_n)\}$ is a finite set of activities.

One way to find the set of activities (V_i, E_i, ρ_i) satisfied by v with the maximum relative probability among all activities in A is to use the MPS algorithm presented earlier as follows:

1. Start with $p_t = 1$.
2. If p_t has not changed since the previous iteration, return the last set of activities obtained at step 3.
3. Run *MPS* and record all activities satisfied with relative probability at least p_t .
4. If there is more than one activity in the previous set, then set p_t to the average between its current and previous value and go to step 2. Otherwise if only one activity is obtained at step 3, return it.
5. If there is no activity in the previously computed set, set p_t to half its previous value and go to step 2.

For reasons of space, we omit a detailed algorithmic description of *naiveMPA*. The algorithm's main issue is the binary search, which can run for many iterations, each taking $\mathcal{O}(n \cdot |v|^2 \cdot |V|)$ steps. Our experiments show that the algorithm is particularly slow when there is more than one activity to be returned.

Example 4 Consider the set of activities depicted in Figure 1 and the following labeling: (5, {detect – person}), (26, {present – card}), (45, {insert – card}), (213, {withdraw – cash}), (320, {insert – checks}), (344, {detect – assailant}), (431, {withdraw – card}), (446, {withdraw – cash}), (496, {–detect – person}), (500, {–detect – person}).

naiveMPA will determine that $T = \emptyset$ until $p_t = 0.0625$. At this point, subvideo [5, 496] satisfies the activity depicted in Figure 1(a) with relative probability $0.089 > p_t$, whereas the subvideo [30, 500] satisfies the activity in Figure 1(b) with relative probability of $0.044 < p_t$, hence the former will be returned.

The activity recognition problem can be more efficiently solved by maintaining state in a similar fashion to *MPS* for each of the activities in the set A . The algorithm is shown in Figure 3.

MPA maintains an array $C[1] \dots C[n]$ of sets, in which $C[i]$ holds the current state for activity (V_i, E_i, ρ_i) in a similar way to the C set used by *MPS*. $m[i]$ holds the maximum relative probability with which $A[i]$ has been satisfied thus far. Pruning occurs when a triple (f, p, a) in $C[i]$ denotes a path that cannot be completed with relative probability greater than $m[i]$ (line 13). As *MPS*, *MPA* makes only one pass through the video, thus permitting incremental computation.

Theorem 3 (MPA correctness) *Algorithm MPA terminates and returns the correct answer – i.e. the set of activities $\{A_i\}$ that is satisfied by v with the highest relative probability among all activities in A .*

Theorem 4 (MPA complexity) *Algorithm MPA runs in time $\mathcal{O}(|v|^2 \cdot \max_{i \in [1, n]} (|V_i|) \cdot n)$.*

5 Experiments and Implementation

Our prototype implementation of the *MPS*, *naiveMPA* and *MPA* consists of approximately 1000 lines of Java code. In addition, we implemented an image processing library of approximately 25,000 lines of Java code. The library operates on video frames, identifying blobs (corresponding to objects in the video), information about such objects (e.g., whether

Algorithm MPA

Input: Video v with labeling ℓ , set of activities A .

Output: $(V_i, E_i, \rho_i) \in A$ that is satisfied by v with the highest probability p and $\forall j \in [1, n], j \neq i, (V_j, E_j, \rho_j)$ is either not satisfied by v or is satisfied with probability $p' < p$.

Notes: C is a vector of sets, each $C[i]$ has the same structure as C for the *MPS* algorithm. By

notation, $f_i(p) = \frac{p - p_{min}^i}{p_{max}^i - p_{min}^i}$, where p_{max}^i, p_{min}^i are the maximum and respectively minimum probabilities for a path in (V_i, E_i, ρ_i) .

```

1. for  $i = 1$  to  $n$  do
2.    $m[i] \leftarrow 0$ ;
3.    $C[i] \leftarrow \emptyset$ ;
4. endfor
5. for every frame  $f_j$  in  $v$  do
6.   for every  $i = 1$  to  $n$  do
7.     for every  $(f, p, a) \in C[i]$  do
8.        $D \leftarrow \ell(f_j) \cap \{b \in V_i \mid (a, b) \in E_i\}$ ;
9.       if  $D \neq \emptyset$  then
10.         $C[i] \leftarrow C[i] - \{(f, p, a)\}$ ;
11.        for  $a' \in D$  do
12.           $p' \leftarrow p \cdot \rho_i(a, a')$ ;
13.          if  $f_i(p' \cdot p_b(a')) > \max(m[1], \dots, m[n])$  then
14.             $T \leftarrow \{(f', p', b) \in C \mid f' = f \wedge a' = b\}$ ;
15.             $C[i] \leftarrow C[i] - T \cup \{(f, \max(\max_{(f, p, b) \in T} (p), p'), a')\}$ ;
16.          endif
17.        endfor
18.      endfor
19.    for every  $a \in \{x \in V_i \mid \exists y \in V_i \text{ s.t. } (y, x) \in E_i\} \cap \ell(f_j)$  do
20.       $C[i] \leftarrow C[i] \cup \{(f_j, 1, a)\}$ ;
21.    for every  $(f, p, a) \in C[i]$  s.t.
22.       $a \in \{x \in V_i \mid \exists y \in V_i \text{ s.t. } (x, y) \in E_i\}$  do
23.         $C[i] \leftarrow C[i] - \{(f, p, a)\}$ ;
24.         $m[i] \leftarrow \max(m[i], f_i(p))$ ;
25.      endfor
26.    endfor
27. return  $\{(A_i, m[i]) \mid m[i] = \max(m[1], \dots, m[n])\}$ ;

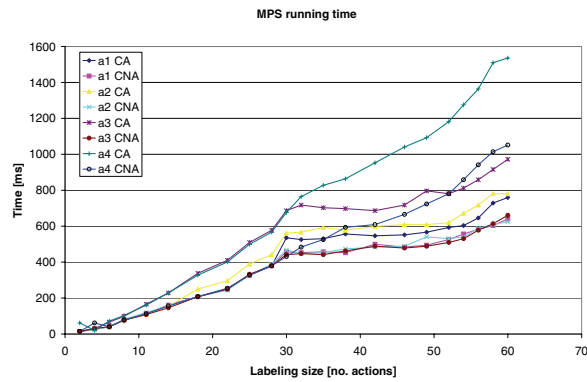
```

Figure 3: An algorithm for the activity recognition problem

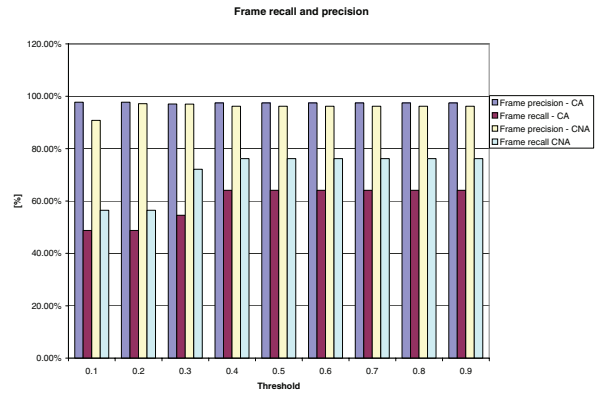
an portion of an image is a person) and implementing motion tracking primitives – this in turn leads to the ability to detect atomic actions (e.g., a person entering a room). The library implements algorithms described in [Elgammal *et al.*, 2000; Bevilacqua, 2003; 2002; Cavallaro *et al.*, 2005; 2000; Makarov, 1996]. Labeling data obtained from video processing was stored in a MySQL DBMS. All experiments were run on a Pentium Centrino at 1.5 Mhz, with 1.5GB of RAM, running Windows XP Professional.

Our experiments were performed over a dataset of 7 video sequences of approximately 15-30 seconds, depicting both regular bank operations and (staged) attempted robberies; the dataset is a superset of that described in [Vu *et al.*, 2003]. We developed five activity definitions with the following notation: (a1) depicts regular customer-employee interaction; (a2) describes a non-employee that enters the vault room (the “safe”); (a3) describes a bank robbery attempt; (a4) describes a successful bank robbery – in which the assailant(s) enter the bank, then enter the safe by holding an employee hostage and then escape; (a5) describes an employee accessing the safe on behalf of a customer.

In the first set of experiments we measured the running time of *MPS* for activity definitions (a1)-(a4). We implemented two version of the algorithm: *CA* refers to the original algorithm in Figure 2; we found that in many scenarios (such as activities at an ATM), we can improve efficiency by assuming activities cannot be interleaved – which lead to the version denoted by *CNA*. The running times for labeling



(a) MPS running time



(b) MPS frame precision and recall

Figure 4: MPS experimental results

size between 2 and 60 actions are shown in Figure 4(a)⁴. The activity definitions (a1)-(a4) are satisfied when the labeling size reaches 30, leading to a visible increase in running time; as expected, *CNA* is more efficient than the original *CA* version. We have also observed that running times are dependent on the complexity of an activity definition.

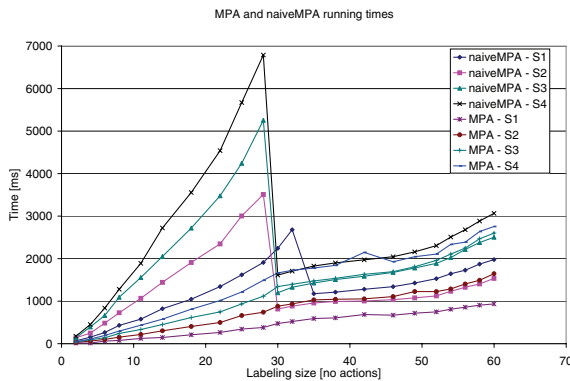


Figure 5: MPA running time

In the second set of experiments, we measure the precision and recall of *MPS* w.r.t. a set of 4 human annotators. The average among the annotations from the four subjects was considered the ground truth when measuring both precision and recall. The results are shown in Figure 4(b). We notice that precision is always very high and generally constant, while recall has a rather surprising variation pattern – namely, recall is monotonic with the threshold. The explanation for the apparently anomalous behavior comes from the fact that for low thresholds, the number of candidate subvideos is relatively high, hence the minimality condition causes *fewer frames* to appear in the answer. The monotonic behavior pattern stops when the threshold becomes high enough (.4 in this case).

⁴This does not include the running time of library methods that generate the labeling.

In the third set of experiments, we analyzed the performance and answer quality for *naiveMPA* and *MPA*. The following sets of activities were used: $S_1 = \{a5\}$, $S_2 = \{a3, a5\}$, $S_3 = \{a3, a4, a5\}$ and $S_4 = \{a2, a3, a4, a5\}$. The running times are shown in Figure 5. *naiveMPA* is especially inefficient until the labeling satisfies one of the activities – this is due to the large number of iterations taken until the threshold is close enough to 0. Running times for the two algorithms are comparable for labeling sizes greater than 30, although *naiveMPA* is always less efficient than *MPA*. The graph shows running time is generally proportional to the number of actions in the labeling.

6 Related work and Conclusions

Model-based anomaly recognition techniques provide an a priori definition of an anomalous activity and then detect new anomalous activities [Hongeng and Nevatia, 2001]. [Zhong *et al.*, 2004; Stauffer and Grimson, 2000] define models of regular activities and *bags of event n-grams* to detect abnormalities. Context-free grammars have also been studied [Ivanov and Bobick, 2000; Vu *et al.*, 2003; Nevatia *et al.*, 2003; Narayanan, 1997]. Oliver *et al.* [Oliver *et al.*, 2002] described an approach based on coupled HMMs for learning and recognizing human interactions. Dynamic Bayesian networks – which can be viewed as a generalization of HMMs –, can be used for tracking and recognizing multi-agent activities [Hamid *et al.*, 2003]. *Data-based* methods detect activities based on extracted features; Cuntoor and Chellappa [Cuntoor and Chellappa, 2006] proposed a characterization of events based on anti-eigenvalues, which are sensitive to changes. Parameswaran and Chellappa [Parameswaran and Chellappa, 2003] computed view invariant representations for human actions in both 2D and 3D.

Ontologies have been recently used in different contexts for video surveillance. Chen *et al* [Chen *et al.*, 2004] use ontologies for analyzing social interaction in nursing homes; Hakeem *et al* use ontologies for classification of meeting videos [Hakeem and Shah, 2004]. Georis *et al* [Georis *et al.*, 2004] use ontologies to recognize activities in a bank monitoring setting. As a result of the Video Event Challenge Workshops

held in 2003, ontologies have been defined for six domains of video surveillance, among which Visual Bank Monitoring and Airport-Tarmac Security; the workshop also led to the development of two languages. The Video Event Representation Language (VERL) [Hobbs *et al.*, 2004], provides an ontological representation of complex events in terms of simpler sub-events. The Video Event Markup Language (VEML) is used to annotate VERL events in videos.

Our work differs from the above approaches in the following respects:

1. Our stochastic activity model is not based on features extracted from data or on state-space representations, but on atomic actions that are recognizable by image understanding primitives.
2. To our knowledge, *MPA* is one of the very few algorithms that addresses the issue of detecting the most likely activity from a given set to have occurred in a certain video. In contrast, model-based representations usually train a specific model to recognize specific situations.
3. To our knowledge, we are the first to find the minimal video segments that contain an event with probability exceeding a threshold - this is important as a security guard wants to zero in immediately on the part of the video that contains the desired activity.

References

- [Bevilacqua, 2002] Alessandro Bevilacqua. *A system for detecting motion in outdoor environments for a visual surveillance application*. PhD thesis, University of Bologna, 2002.
- [Bevilacqua, 2003] Alessandro Bevilacqua. Effective shadow detection in traffic monitoring applications. In *WSCG*, 2003.
- [Cavallaro *et al.*, 2000] A. Cavallaro, F. Ziliani, R. Castagno, and T. Ebrahimi. Vehicle extraction based on focus of attention, multi feature segmentation and tracking. In *Proceedings of X European Signal Processing Conference (EUSIPCO)*, pages 2161–2164, Tampere, Finland, September 2000.
- [Cavallaro *et al.*, 2005] Andrea Cavallaro, Olivier Steiger, and Touradj Ebrahimi. Tracking video objects in cluttered background. *IEEE Trans. Circuits Syst. Video Techn.*, 15(4):575–584, 2005.
- [Chen *et al.*, 2004] D. Chen, J Yang, and H.D. Wactlar. Towards Automatic Analysis of Social Interaction Patterns in a Nursing Home Environment from Video. In *MIR 04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 283–290. ACM Press, 2004.
- [Cuntoor and Chellappa, 2006] N. P. Cuntoor and R. Chellappa. Key frame-based activity representation using antieigenvalues. In *Proceedings of the Asian Conference on Computer Vision*, 2006.
- [Elgammal *et al.*, 2000] Ahmed M. Elgammal, David Harwood, and Larry S. Davis. Non-parametric model for background subtraction. In *ECCV '00: Proceedings of the 6th European Conference on Computer Vision-Part II*, pages 751–767, London, UK, 2000. Springer-Verlag.
- [Georis *et al.*, 2004] B. Georis, M. Maziere, F. Bremond, and M. Thonnat. A Video Interpretation Platform Applied to Bank Agency Monitoring. In *IDSS'04 - 2nd Workshop on Intelligent Distributed Surveillance Systems*, FEB 23 2004.
- [Hakeem and Shah, 2004] A. Hakeem and M. Shah. Ontology and Taxonomy Collaborated Framework for Meeting Classification. In *ICPR (4)*, pages 219–222, 2004.
- [Hamid *et al.*, 2003] R. Hamid, Y. Huang, and I. Essa. Argmode - activity recognition using graphical models. In *Proc. IEEE Computer Vision and Pattern Recognition*, volume 4, pages 38–43, 2003.
- [Hobbs *et al.*, 2004] J. Hobbs, R. Nevatia, and B. Bolles. An Ontology for Video Event Representation. In *IEEE Workshop on Event Detection and Recognition*, 2004.
- [Hongeng and Nevatia, 2001] Somboon Hongeng and Ramakant Nevatia. Multi-agent event recognition. In *ICCV*, pages 84–93, 2001.
- [Ivanov and Bobick, 2000] Yuri A. Ivanov and Aaron F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):852–872, 2000.
- [Makarov, 1996] A. Makarov. Comparison of background extraction based intrusion detection algorithms. In *International Conference Image Processing ICIP96*, page section 16P3, 1996.
- [Narayanan, 1997] S. Narayanan. *KARMA: Knowledge-based Action Representations for Metaphor and Aspect*. PhD thesis, University of California, Berkeley, 1997.
- [Nevatia *et al.*, 2003] R. Nevatia, T. Zhao, and S. Hongeng. Hierarchical language-based representation of events in video streams. In *Proceedings of the Workshop on Event Mining (in conjunction with IEEE CVPR)*, 2003.
- [Oliver *et al.*, 2002] N. Oliver, E. Horvitz, and A. Garg. Layered representations for human activity recognition. In *Proc. IEEE International Conference on Multimedial Interfaces*, pages 3–7, 2002.
- [Parameswaran and Chellappa, 2003] V. Parameswaran and R. Chellappa. View invariants for human action recognition. In *Proceedings of IEEE Computer Vision and Pattern Recognition*, 2003.
- [Stauffer and Grimson, 2000] Chris Stauffer and W. Eric L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):747–757, 2000.
- [Vu *et al.*, 2003] Van-Thinh Vu, François Brémond, and Monique Thonnat. Automatic video interpretation: A novel algorithm for temporal scenario recognition. In *IJCAI*, pages 1295–1302, 2003.
- [Zhong *et al.*, 2004] Hua Zhong, Jianbo Shi, and Mirkó Vissontai. Detecting unusual activity in video. In *CVPR (2)*, pages 819–826, 2004.