Propagating Certainty in Petri Nets for Activity Recognition
Gal Lavee, Michael Rudzsky, and Ehud Rivlin, Member, IEEE

Abstract—This paper considers the problem of recognizing activities as they occur in surveillance video. Activities are high-level nonatomic semantic concepts which may have complex temporal structure. Activities are not easily identifiable using image features, but rather by the recognition of their composing events. Unfortunately, these composing events may only be observed up to a particular certainty. This paper describes particle filter Petri Net (PFPN), an activity recognition process that combines uncertain event observations to determine the likelihood that a particular activity is taking place in a video sequence. Our paper is based on previous study in which activities are specified as Petri Nets. The stochastic PFPN framework proposed in this paper improves over existing deterministic approaches to activity recognition by enabling the certainty reasoning required for coping with inherent ambiguity in both low-level video processing and activity definition. Furthermore, the PFPN approach reduces the dependence on a duration model and enables the creation of holistic activity models. Often when activity recognition frameworks are proposed they are strongly paired with a particular methodology for low-level video processing and event recognition. Each proposed approach is then applied to a nonstandard dataset. In our experiments, we provide an empirical comparison of our approach with leading activity recognition approaches across several datasets, using a constant event recognition as input. Our results illustrate the tradeoff between deterministic and stochastic activity recognition approaches. Furthermore, our experiments suggest that the holistic PFPN approach is more robust for activity recognition in the surveillance video domain than competing approaches.

Index Terms—Activity recognition, event recognition, particle filter, Petri net, video surveillance.

I. INTRODUCTION

The ability of humans to recognize activities in video sequences greatly exceeds that of the most successful automatic video activity recognition approaches. One reason for this may be that human brains are able to combine semantic knowledge about the structure of an activity with reasoning under ambiguity. That is, our knowledge about the kind of occurrences possible in the scene influences our decision when we observe some ambiguous information.

Manuscript received August 31, 2011; revised January 21, 2012, April 2, 2012; accepted May 8, 2012. Date of publication June 9, 2012; date of current version February 1, 2013. This paper was recommended by Associate Editor T. Zhang.

The authors are with the Faculty of Computer Science, The Technion Israel Institute of Technology, Haifa 32000, Israel (e-mail: gallavee@cs.technion.ac.il, rudzsky@cs.technion.ac.il, evah@cs.technion.ac.il).

Digital Object Identifier 10.1109/TCSVT.2012.2203742

An activity [8], [9], [21] is defined as a set of events with partial temporal interval ordering. An event is defined as an atomic occurrence restricted to a continuous temporal interval. The problem of activity recognition is determining whether a particular activity occurs in a video sequence.

Event recognition, the labeling of event occurrences, is itself a much studied and nontrivial problem (see Section II). Due to the difficulties in event recognition, event observations are often associated with a certainty measure that quantifies the event recognizer’s confidence. Certainty measures for each event observation are often ad hoc, heterogeneous, and outside the control of the activity recognition algorithm design. Thus, a general activity recognition approach should be agnostic to how the event observation certainty is calculated.

As a motivating example, let us consider an automatic bank surveillance system. We are interested in issuing an automatic alert on the occurrence of such hypothetical activities as Bank Attack (see Fig. 1): a customer and teller are present in the bank, the teller enters the safe, the customer enters the safe, and the customer leaves. The customer and teller may appear at any time but must both be present before the first safe access. The teller may enter the safe before the customer or vice versa, but they may not enter at the same time. Both the teller and the customer must have been in the safe before the customer departs.

Clearly, modeling the complex temporal relationships which define this activity is a major challenge of constructing this type of system. Another primary challenge is combining the observation certainties of participating events to arrive at a defined certainty measure for the activity recognition.

In order to issue timely alerts, the system is also required to maintain a dynamic state estimation as new observations become available.

The main contribution of this paper is an activity modeling approach that utilizes the Petri Net (PN) formalism [3], [11], [23], [29], which is robust to modeling complex temporal relations. To perform recognition, this activity specification is translated into the Bayesian recursive framework (BRF). Within this framework, observation event certainties may be treated as features in the measurement model. This construction allows application of the particle filter algorithm, enabling reasoning on uncertain event observations independent of how observation certainty measures are computed. Furthermore, the recursive framework maintains a system state estimation at each frame of the video, allowing dynamic alerts when activities are recognized.
The variable nature of video content has resulted in a large number of approaches, each analyzing video data with slightly different semantic content. Furthermore, each approach is often paired with specific approaches to low-level video processing and event recognition. The lack of a direct comparison of activity recognition approaches in the literature that is independent of these other factors (2 is a notable exception) makes it difficult to reach conclusions on the merits of a particular approach. Thus, another contribution of this paper is the empirical evaluation of several leading approaches for activity recognition on the same datasets, where low-level video processing and event recognition are kept constant across all experiments.

This paper is organized as follows. Section II gives an overview of related studies in activity recognition. Sections III and IV review some background on PN and particle filter ideas. Section V introduces our approach to activity recognition. Section VI describes our experiments and provides discussion of their results. Finally, we conclude the paper in Section VII.

II. RELATED WORK IN ACTIVITY RECOGNITION

We distinguish activity recognition (sometimes called scenario recognition [35]) from the related field of event recognition, the labeling of event occurrences in video sequences. Event recognition approaches output a certainty measure quantifying the confidence in each event recognition. This measure must take into account the uncertainties inherent in the video data. We informally group these uncertainties into two categories: observation uncertainty and semantic uncertainty. Observation uncertainty refers to the limitations of sensors (such as cameras) to provide exact measurements, as well as to the noise element associated with many systems. Semantic uncertainty refers to the ambiguity inherent in human definitions. For example, given a location of two objects, their “closeness” is an ambiguous semantic concept.

Each event recognition approach must make a decision on how to combine the various types of uncertainty in order to calculate the measure of certainty. This decision is often made in an ad hoc manner. In many cases, a separate certainty evaluation is applied for each type of event. The event recognition literature contains many approaches to the calculation of measures of certainty, including Bayesian networks [33], scene statistics [29], [30], and fuzzy membership functions [13], [14]. Alternatively, events may be detected in a binary fashion [3], or made binary by thresholding [35]. In other approaches to event recognition, the semantics of an activity domain can be formalized to enable the system to better localize event occurrences [10], [19], [25], [34].

We further distinguish paper in activity recognition from the popular research domain of action recognition [7], [32]. Activities, unlike actions, have a large variation in appearance and are not sufficiently characterized by image features such as color, texture, and edge orientation. Additionally, approaches to action recognition often rely on the availability of a large corpus of training data, which may be used to construct classifiers for unlabeled examples. The domain of activity recognition in surveillance video generally does not have this luxury. Interesting activities are usually rare, and few, if any, training examples are available. This constraint also limits the usefulness of probabilistic state space models such as hidden Markov models (HMM), which generally require a number of examples on the order of the state space. For these reasons, the activity recognition community has turned to using formal specifications of domain knowledge to model activities in video.

Since there is no standard way to specify activities, each paper is obliged to define a formal representation of an activity. Thus, there are two main components of existing approaches to activity recognition: 1) a modeling formalism—allowing formal specification of activity models by a human domain expert; and 2) a recognition algorithm—an approach to processing video input in combination with the formal activity specification to output an activity label. Unfortunately, in most cases these two components have strong coupling and it is difficult (although theoretically possible) to evaluate the merits of a particular formalism independently of the recognition algorithm that is attached to it.

Many well-studied formalisms have been applied to activity model specification. Several approaches have specified activities using logical formulas [2], [10], [19], [25], [34], [35]. Each of these papers typically defines several predicates and relational operators to enable reasoning about temporal relationships between events. Formal grammars have also been used to describe activity composition [17], [31]. The PN [3], [20], [23], [29] are a useful formalism for modeling activities which have multiple streams of events with partial ordering constraints between them (see Section III).

Other formalisms have been designed with the problem of activity recognition in mind. The situation graph tree (SGT) formalism (used in [13] and [14]) is robust in representing generalization and specialization hierarchies. The Propagation Net (P-net) formalism [33] allows modeling of multiple “streams” of events within the activity by constructing a graph representing the temporal structure of the activity. A parameterized duration model is also associated with each node and edge in the graph. A related formalism, activity dynamic Bayesian network (ADBn), which adds an element of hierarchy between activities, is proposed in [24].

Deterministic algorithms for activity recognition rely on the resolution of ambiguity at lower levels of processing. An algorithm called Store Totally Recognized Scenarios (STRS) [35], the leading deterministic recognition algorithm, has been applied to several real-world domains. This approach is used to recognize activities specified using temporal constraint logic. Upon observation of a new event, this algorithm stores the event and determines if it completes a (sub)activity. If so, the completed activity is stored and checked against higher level activities to determine if they were completed.

A common deterministic algorithm which uses a PN activity specification [3], [11], [15], [23], [29] proceeds as follows: each event observation moves a token through a PN model of the activity toward the place node indicating the recognition of the activity.

Stochastic algorithms for activity recognition generally maintain a probability distribution over the state space of
the activity. In most cases, the state space is described in terms of a set of random variables. The joint distribution over these variables is often simplified with assumed independence relationships (e.g., the Markov assumption) which break this distribution into a product of factors. These assumptions reduce the complexity of storage and enable efficient computation of the joint and appropriate marginal probabilities.

D-Condensation is a stochastic algorithm, based on sampling, applied to activities specified as propagation nets [33]. This algorithm has been shown to be effective in such applications as quality control of glucose monitoring. To allow simplification of the joint distribution it is assumed that nodes (events) in the net become active (when the event occurs) in a restricted order. Furthermore, this approach assumes a separate distribution over the duration of each event and the time between events is known in advance. The algorithm for recognizing activities formalized as ADBNs [24], uses similar assumptions, but assumes only a duration model for the time between events. [2] proposes a generic framework for probabilistic logical entailment.

III. PETRI NETS

The PN are specified as a directed bipartite graph (see Fig. 1). Graphically, place nodes are represented as circles and transition nodes are represented as rectangles. Each place node holds zero or more tokens in each state and transition nodes specify the movement of tokens between place nodes when a state change occurs. Those place nodes connected by directed arcs into the transition node are the input place nodes of the transition. Similarly, those place nodes connected by directed arcs out of the transition are called the output place nodes of the transition. A transition node is enabled if all input place nodes connected to it hold tokens. A special type of arc called an inhibitor arc requires that an input place node must not hold a token in order for the transition to become enabled. Enabled transition nodes may fire, altering the distribution of tokens in the PN. When an enabled transition node fires, the tokens held in the input place nodes are deleted and new tokens are placed in each of the output place nodes. Conditional transition nodes can have an enabling rule applied to them which imposes additional conditions on the enabling of the transition. A PN model marking is defined as the instantaneous configuration of the tokens held in the various place nodes in the PN graph. For further details on the PN formalism interested readers are referred to [18], [26].

The prevalent approach to modeling video activities with PN is connecting fragments corresponding to the composing events in such a way that enforces the temporal and logical relationships between them. Using the PN formalism, we are able to model all temporal interval relations as defined in [4]. Place nodes are “waypoints” which indicate the progress throughout the activity. Special place nodes indicate the start and end of the activity. The “start” place node will hold a token when the recognition of the activity is still in its initial state. The “end” place node will hold a token when the activity recognition has been completed. Otherwise, one or more of the intermediate place nodes will hold a token (e.g., the token configuration pictured in Fig. 1 indicates the activity state: the teller is present, but the visitor has yet to appear). Each transition node in the figure has an associated event label. For a transition to fire, this event must be observed while the transition is enabled. For further details on the construction of PN activity models, readers are referred to the literature [3], [18], [23], [29].

Consider the PN definition of an activity depicted in Fig. 1 (For more examples of activity models, the reader is referred to [22]). This is part of the Bank Attack activity used in our experiments which is closely modeled after an activity PN proposed in [3]. The PN activity model in the figure may be formalized as a tuple \( \langle P, T, C, \text{events}, h, S, F \rangle \) where \( P = \{P_1, P_2, \ldots, P_{10}\} \) is the set of places, \( T = \{T_1, T_2, \ldots, T_{17}\} \) is the set of transitions, \( C \) is the set of connecting arcs, \( \text{events} = \{\text{Visitor Appears}, \text{Teller Appears}, \text{Teller in Safe}, \ldots\} \) is the set of events that are relevant to the activity, and \( h : T \rightarrow \text{events} \) is the labeling function mapping transitions to an event label implied by the figure. For example, \( h(T_3) = \text{Visitor in Safe} \). \( S = \{P_1\} \) is the place node representing the “start” of the activity, and \( F = \{P_{10}\} \) is the set of place nodes representing the recognition of the activity. For a more in depth discussion of modeling activities with PN see [23].

Many of the activities in the surveillance domain that we are interested in recognizing share several events. The main differences between these activities lies in the configuration of these events. For instance, Visitor Disappears is the final event in the Bank Attack activity (described earlier) as well as other activities including the Normal activity (describing a normal bank customer interaction). This example illustrates the importance of modeling the possibility of events occurring outside the context of the activity. We refer to an approach that allows modeling of events in both the activity and nonactivity context as a holistic approach to activity modeling. Such an
approach is especially important in a stochastic setting where event observations are used to reinforce the state estimation. In this setting, an event occurring outside the activity context, which is not included in the model may cause an inappropriate redistribution of probability mass, often leading to false positive recognitions. Fortunately, the PN formalism affords us the capacity to model activities holistically, without major extensions, by adding additional transition nodes appropriately.

IV. PARTICLE FILTER

In a Bayesian approach to analyzing dynamic systems, the goal is to estimate the posterior distribution $P(x_{t+1}|y_{1:t})$ over the system state at time $t$, denoted by $x_{t}$. Taking into account all observations up to the current time, denoted by $y_{1:t}$. The Bayesian recursive filter (BRF) is an approach appropriate for online problems in which the posterior estimation of the previous state is used to arrive at an updated estimation with each new observation. Particle filters (PF), also known as sequential Monte Carlo methods, are techniques for probability density estimation within the BRF framework based on sampling. The PF approach maintains a set of $N$ hypotheses of the current state, called particles, denoted as $X = \{x^{(1)}_{t}, x^{(2)}_{t}, \ldots, x^{(N)}_{t}\}$. Each particle $x^{(i)}_{t}$ is associated with a weight $w^{(i)}_{t}$. The weights are normalized to sum of one, and hence describe a probability distribution over the states. This distribution is used to approximate the posterior distribution as follows:

$$P(x_{t}|y_{1:t}) \approx \sum_{i=1}^{N} \delta(x_{t}-x^{(i)}_{t})$$

where $\delta$ denotes the Dirac delta function.

The BRF framework consists of two major components. The dynamic model, denoted as $P(x_{t}|x_{t-1})$, describes the evolution of states over time. The measurement model, denoted as $P(y_{t}|x_{t})$, specifies the relationship between state and observation variables. After initialization, a BRF approach proceeds in two stages, prediction and correction. In the prediction stage, the previous estimation of the posterior is used to arrive at an estimation of possible future states. The PF algorithm samples each new particle, $x^{(i)}_{t}$, from $P(x_{t}|x^{(i)}_{t-1})$. This distribution, called the proposal (or importance) distribution, is often used as an approximation to the posterior. In the correction stage, the most recent observation, $y_{t}$, is used to adjust the prediction, increasing (decreasing) the probability of states consistent (inconsistent) with the observation. In the PF algorithm each particle weight, $w_{t}$, is multiplied by $P(y_{t}|x^{(i)}_{t})$. The weights are then renormalized to estimate the posterior. Particle filter algorithms are used in many application fields including visual tracking of objects [16]. For more on particle filter approaches the reader is referred to [5] and [12].

V. OUR APPROACH

Our approach to activity recognition takes two inputs: 1) an activity definition, specified as a PN, and 2) a video sequence specified as a list of time-stamped events with associated certainties.
the set of all reachable markings
the set of all markings where the activity is recognized
the set of all relevant events
the set of all relevant event combinations
the state of the system at time \( t \)
the initial Petri Net marking of the system
the underlying Petri Net marking at time \( t \)
the event combination occurring at time \( t \)
the observation at time \( t \)
the confidence in the observation of the \( i \)th event in \( O \) at time \( t \)

2) State Space: The Bayesian paradigm requires a dynamic model in which the current state is fully determined by the previous state(s). Therefore, in order to model PN dynamics in this paradigm, we must consider the (possibly empty) set of events that occur at each frame as a component of the state. Thus, \( x_t = (\text{state}_t, \text{event}_t) \), the state variable at frame \( t \) of the video sequence, is a two component tuple. The first component, \( \text{state}_t \), denotes the marking of the PN at time \( t \). The second component, \( \text{event}_t \), denotes the event(s) (if any) that occur at time \( t \).

3) Observation Space: At each frame, we will observe a certainty value for each relevant event in the set \( O \). The 4th component of \( y_t \), denoted as \( y_{t(i)} \), represents the certainty in the observation of \( O(i) \), the \( i \)th event in \( O \).

4) Conditional Independence: In addition to defining our state and observation variables, we also assume some conditional independence among the variables to reflect the nature of the underlying PN. First, our state space is factorized using the (first order) Markov assumption, which assumes that the state at any frame \( t \) is independent of the state at all previous frames, given the state at frame \( t-1 \). Thus, it is only necessary to maintain an approximation of the dependency between the states at time slices \( t \) and \( t-1 \). Additionally, we assume that the events that occur in a particular frame can only be those corresponding to enabled transitions in the underlying PN. Furthermore, given the events that occur at time \( t \), the observation variable is independent of the current marking. More concretely, \( \text{event}_t \) is independent of \( \text{state}_{t-1} \) and \( \text{state}_{t-1} \) given \( \text{state}_t \). Also \( y_t \) is independent of \( \text{state}_t \) given \( \text{event}_t \). These conditional independence relations are illustrated by the simple graphical model shown in Fig. 2.

5) Dynamic Model: The dynamic model, \( P(x_t|x_{t-1}) \), is decomposed as follows:

\[
P(x_t|x_{t-1}) = P(x_t|x_{t-1}, y_{t(i)}) P(y_{t(i)}|x_{t-1})
\]

where the second equality is due to the conditional independence of \( x_{t-1} \) from \( x_{t-2} \) and \( x_{t-3} \) given \( x_{t-1} \) (see Fig. 2). Thus, it suffices to derive \( P(x_t|x_{t-1}, y_{t(i)}) \) from the PN structure. We derive an uninformative \( P(x_t|x_{t-1}, y_{t(i)}) \), giving equal probability to all enabled transitions from the structure of the PN as follows:

\[
\hat{P}(x_{t(i)}|x_{t(i)}) = \begin{cases} 1, & \text{if all transitions in } x_{t(i)} \text{ are enabled in marking } x_{t(i)} \\ 0, & \text{otherwise} \end{cases}
\]

where \( x_{t(i)} = \emptyset \) (the empty set) is considered to be enabled in all markings. We proceed to normalize the above

\[
P(x_{t(i)}|x_{t(i)}) = \frac{\hat{P}(x_{t(i)}|x_{t(i)})}{\sum_{t(i)} \hat{P}(x_{t(i)}|x_{t(i)})}
\]

Similarly, \( P(x_{t(i)}|x_{t-2}, x_{t-1}) \) is derived from the PN as follows:

\[
P(x_{t(i)}|x_{t-2}, x_{t-1}) = \begin{cases} 1, & \text{if } x_{t(i)} \leftarrow x_{t-1} | \{x_{t-1}|x_{t-1}\} \\ 0, & \text{otherwise} \end{cases}
\]

where \( x_{t(i)} \leftarrow x_{t-1} | \{x_{t-1}|x_{t-1}\} \) indicates there is a path from state \( x_{t-1} \) to state \( x_{t(i)} \) via transitions labeled with the event(s) \( x_{t-1} \). We then proceed to normalize the above

\[
P(x_{t(i)}|x_{t-2}, x_{t-1}) = \frac{\hat{P}(x_{t(i)}|x_{t-2}, x_{t-1})}{\sum_{t(i)} \hat{P}(x_{t(i)}|x_{t-2}, x_{t-1})}
\]

6) Measurement Model: In order to derive the measurement model, \( P(y_t|x_t) \), we first assume that all components of the observation vector are independent of one another. This assumption yields the following equation:

\[
P(y_{t(i)}|x_t) = \prod_{i=1}^{n} P(y_{t(i)}|x_t)
\]

where \( y_{t(i)} \) denotes the \( i \)th component of the observation vector. Furthermore, using the assumption that \( y_t \) is conditionally independent of \( x_{t(i)} \) given \( x_{t-1} \) (see Fig. 2) we get

\[
P(y_t|x_t) = \prod_{i=1}^{n} P(y_{t(i)}|x_{t(i)})
\]

It therefore suffices to derive \( P(y_{t(i)}|x_{t(i)}) \) from the PN structure to obtain the measurement model.

The construction of the observation likelihood, \( P(y_{t(i)}|x_{t(i)}) \), models the notion that if a particular event occurs, the observed certainty in this event should be high. The sigmoid function (centered on 0.5) is a natural fit for modeling this intuition. Conversely, if an event does not occur, the observed certainty in this event should be low. The complement to the sigmoid function models this alternate situation. Recall, the \( i \)th component of \( y_t \), denoted as \( y_{t(i)} \), represents certainty in the \( i \)th event in set \( O \), denoted by \( O(i) \). Since, \( x_{t(i)} \in M \subseteq 2^O \) (the
In order to arrive at an activity classification of an unlabeled video sequence, we constructed an experimental framework with several components (see Fig. 3). The raw image data (input 1) are fed to Module 1 (Object Detection and Tracking) to generate a list of objects and their corresponding locations at each frame. In our experiments, we used the ground truth locations on the synthetic bank dataset, a particle filter tracker [16] on the ETISEO dataset, and a tracker based on boosting [6] for the Technion dataset. We used a semiautomatic

As previously mentioned, the particle weights provide an approximation of the posterior probability distribution, $P(x_t|y_1:t)$. Thus, summing the weights of those particles where $x_{(i)} \in R$, allows determining the probability that the activity is recognized.

**VI. EXPERIMENTS**

In our experiments, we endeavored to provide a rigorous empirical comparison of our activity recognition approach to other leading approaches. In order to make the comparison fair, we used the same video processing and event recognition system as input to all approaches. We evaluated the different approaches over three datasets.

**A. System Description**

In order to arrive at an activity classification of an unlabeled video sequence, we constructed an experimental framework with several components (see Fig. 3). The raw image data (input 1) are fed to Module 1 (Object Detection and Tracking) to generate a list of objects and their corresponding locations at each frame. In our experiments, we used the ground truth locations on the synthetic bank dataset, a particle filter tracker [16] on the ETISEO dataset, and a tracker based on boosting [6] for the Technion dataset. We used a semiautomatic

As previously mentioned, the particle weights provide an approximation of the posterior probability distribution, $P(x_t|y_1:t)$. Thus, summing the weights of those particles where $x_{(i)} \in R$, allows determining the probability that the activity is recognized.

**VI. EXPERIMENTS**

In our experiments, we endeavored to provide a rigorous empirical comparison of our activity recognition approach to other leading approaches. In order to make the comparison fair, we used the same video processing and event recognition system as input to all approaches. We evaluated the different approaches over three datasets.

**A. System Description**

In order to arrive at an activity classification of an unlabeled video sequence, we constructed an experimental framework with several components (see Fig. 3). The raw image data (input 1) are fed to Module 1 (Object Detection and Tracking) to generate a list of objects and their corresponding locations at each frame. In our experiments, we used the ground truth locations on the synthetic bank dataset, a particle filter tracker [16] on the ETISEO dataset, and a tracker based on boosting [6] for the Technion dataset. We used a semiautomatic
approach to tracking, where if an object track had been lost by the tracker, the tracker was again reinitialized using the
size event recognition and do not include repeated nontrivial

true negatives, false positives, and false negatives. Since each
ground truth allows us to compute the number of true positives,
for this activity over the length of the video sequence.

The object locations are input into Module 2 (Event Recog-
nition). Specification of scene information (Input 2), such as
definitions and camera calibration information, is also
input into Module 2. We chose to implement a simple rule-
based event recognition similar to the one adopted in [35]. In
our experiments, we recognize several kinds of events includ-
ing: Appearance, Disappearance, In Zone, Moving, Stopped,
Near (2 objects), Far (2 objects). We make use of the
scene specification to recognize these events and associate the
appropriate certainty value. Fig. 4 shows a keyframe from our
datasets annotated with the output of Modules 1 and 2.

The list of events and their corresponding certainty values
that occur at each frame output by Module 2 are input
into Module 3 (Activity Recognition). The activity model
specifications (which may be PN, propagation nets (P-Net),
temporal constraint logic models) (input 3) are also input into
Module 3. This module outputs a list of recognized activities
in the video sequence. In our experiments, we implemented
two different types of activity recognition approaches. The
STRS approach is implemented based on [33] and represents
deterministic approaches to activity recognition. In this
approach, event inputs were thresholded and only those whose
certainties were above a threshold were passed into the activity
recognition module. The P-Net approach is implemented based
on [33] and represents competing probabilistic approaches
to activity recognition. Note that the P-Net approach relies
on an available duration model for each of the events in
the activity. In general, such a model is not available and
an uninformative duration model was applied. The exception
to this was the synthetic bank dataset, which allowed us to
provide a distribution over each event duration. The particle
filter Petri Net (PFPN) is our approach and is implemented
using the methods explained in detail in section V. We used
$N = 2000$ particles in our experiments. The value of this
parameter was determined empirically to provide the best
tradeoff between performance and running time.

B. Metrics

In order to give performance results that allow for com-
parison of the various activity recognition approaches, we
must determine when an activity has been recognized. For
those approaches that output a certainty associated with each
recognition (e.g., P-Net and PFPN) we simply threshold this
value. In all of our experiments, an activity is considered to be
recognized if its associated certainty is above some threshold
$\theta$. Since at each frame, we may have a different activity recog-
nition certainty, we simply threshold the maximum certainty
for this activity over the length of the video sequence.

Comparing the activity recognition output to the available
ground truth allows us to compute the number of true positives,
true negatives, false positives, and false negatives. Since each
clip has only a few activities occurring in it, the number of
positive examples is significantly smaller than the number of
negative examples. This is also the case in real-surveillance
systems. For this reason, an approach may achieve a fairly
high-accuracy score by classifying all examples as negative.
Thus, when evaluating each approach, it is important to take
into account the tradeoff between true and false positive
recognitions. To this end, we have chosen to present the results
as an receiving operating characteristic (ROC) curve, which
plots the true positive rate (also known as recall) against the
false positive rate. These metrics are given by the formulas

\[
\text{true positive rate} = \frac{tp}{tp + fn} \quad (12)
\]

\[
\text{false positive rate} = \frac{fp}{fp + tn} \quad (13)
\]

Clearly, as we decrease our recognition threshold, $\theta$, from
1.0 to 0 we will achieve a higher number of both true and
false positives, causing both earlier rates to rise. An ideal
threshold selection would achieve a high-true positive rate and
a low-false positive rate. Generally, a tradeoff exists between
these rates and we must choose a threshold to favor one or the
other. The ROC curve presentation illustrates this tradeoff
for a number of values of $\theta$, for each of the algorithms we
have evaluated in this paper. We have also computed the area
under the curve (AUC) which quantifies this tradeoff as a real
number.

It should be clarified that the STRS algorithm is not a
stochastic algorithm, thus its decision is binary and cannot
be thresholded. To compile the curve for this algorithm, we
instead varied the threshold for event recognition, that is, the
certainty threshold above which an event will be considered
recognized.

C. Datasets

Unfortunately, a surveillance video dataset with annotated
activities is time consuming to compile. Recent public datasets
pertaining to surveillance video, such as VIRAT [28], empha-
size event recognition and do not include repeated nontrivial
activities such as the ones we consider in this paper. Existing activity datasets have only a few examples of each activity. Thus, in addition to using a publicly available dataset, we captured and annotated our own dataset. In order to allow evaluation of our approach on a large number of examples, we created synthetic animated video clips. These clips can be created at considerably less cost than real data and serve to further illustrate the effectiveness of our approach. Example activity clips from the datasets used in our experiments can be viewed at [1].

1) Synthetic Bank Dataset: In the first set of experiments, we considered 500 short synthetic video clips lasting from 274-526 frames (9-17 s) each. Based on [3] we considered the activities: 1) Bank Attack; 2) Attempted Bank Attack; 3) Normal Customer Interaction; 4) Cashier accesses safe; and 5) Outsider enters safe. Most of the clips contain one or more of these activities. An example sequence from this dataset is shown in Fig. 5.

Fig. 5. Example sequence from Synthetic Bank dataset. A bank attack activity is taking place. Sequence should be interpreted left to right, top to bottom.

2) ETISEO Building Entrance Dataset: In order to evaluate our approaches on an existing, publicly available set of video sequences, we chose the ETISEO building entrance dataset [27]. This dataset is a publicly available dataset of real videos, which includes multiple camera views of the scene and includes several nontrivial activities. The ground truth of object locations across the different camera locations is available for download. The ground truth activity labeling was manually annotated by the authors. This dataset contains six sequences from up to four camera angles (though not all sequences contain data for all cameras angles). We defined six activities that can take place in this domain: 1) Arrive By Car; 2) Arrive On Foot; 3) Depart By Car; 4) Depart on Foot; 5) Meet and Walk Together; and 6) Meet and Walk Apart. Note that different scene objects may be involved in one or more activities. The various camera angles provide an additional challenge to the event recognition module: how to integrate event recognitions from different camera angles. In our experiments, we chose a simple approach of merging events from all camera angles into a single list before inputting this list into the activity recognition module. In the case where the same event is detected in multiple camera angles, we choose the event with the highest certainty value. The sequences in this dataset ranged from 924 to 1649 frames in (30-54 s) in length. Each sequence contained one or more of the activities. An example sequence from this dataset is shown in Fig. 6 (additional sequences can be seen in [22]).
Fig. 7. Example sequence from Technion Parking Lot dataset. An arrive by car activity is occurring simultaneously with a car theft activity taking place. Sequence should be interpreted left to right, top to bottom.

Fig. 8. Experiment 1: ROC curve for the Synthetic Bank dataset.

Fig. 9. Experiment 2: ROC curve for the ETISEO Building Entrance dataset.

Fig. 10. Experiment 3a: ROC curve for the Technion Parking Lot dataset (real tracking).

D. Discussion

In examining Figs. 8–11 as well as Table II, we can make several observations regarding our experiments. The first observation is that the lowest false positive rates are achieved by the deterministic STRS approach. In the bank dataset, this approach achieves the highest AUC, and achieves a comparable true positive rate with all other approaches. However, in the other experiments, where the observation and semantic uncertainty grows, the relative performance of this approach degrades in comparison to the stochastic approaches. The reason behind this phenomenon is that deterministic
approaches such as STRS must observe every event component of an activity in order for the activity to be recognized. If one of these event components is observed with low certainty (below the threshold), or not observed at all, the deterministic approach will not be able to recognize the activity and will record a false negative. Stochastic approaches that are able to reason on uncertain input, such as PFPN, have a mechanism to “fill in the gaps,” which allows the recognition of an activity, in spite of this missing information. Informally, this is done by predicting all the events that may happen, and then validating these predictions with future observations. The downside of this approach is that occasionally event “hallucinations” will occur inappropriately, causing false recognitions of activities.

In other words, the stochastic approaches tend to err on the side of false positives. Clearly, this result implies a tradeoff. If a particular application requires a near zero false positive rate, a deterministic approach to activity recognition is a better fit for this application. In the inverse case, where a false negative is more costly than a false positive (e.g., sending a policeman to the bank as opposed to letting the bank be robbed), a stochastic approach such as PFPN is preferable.

Among the stochastic methods examined in our experiments, PFPN achieves a lower false positive rate while achieving higher or roughly equivalent true positive rate. The PFPN approach also achieves a higher AUC score than the P-Net model in all but Experiment 1 (the bank dataset). It should be noted that in the bank dataset experiment, the P-Net model has more information at its disposal (i.e., the duration model). We explain this improvement by noting that the PFPN framework allows construction of a holistic activity model that models events in both the activity and nonactivity context. In contrast, the P-Net only models the activity, not taking into account events that might occur outside the activity context. Recall that many activities in the surveillance domain have similar event composition, and sometimes differ only in the temporal configuration of the component events. In other words, events often occur outside the context of the activity being modeled (as part of a competing activity or as a result of image processing errors). If this type of occurrence is not appropriately modeled, a stochastic activity recognition approach may inappropriately estimate the activity state. Often these types of event observations result in false positive activity recognitions.

Like other stochastic approaches, PFPN models the activity recognition problem as a joint probability estimation over many random variables. The size of this state space is directly related to the time complexity of these algorithms. In particular, particle filter algorithms, whose complexity depends on the number of particles, require a number of particles that increase with the size of the state space. Thus, in order to make stochastic activity recognition tractable, assumptions are made to reduce the size of this space. In the P-Net model, each event participating in the activity is represented as a variable which contains a start time and a duration. During recognition, the PFPN approach applies an assumption to allow simplification of the joint distribution over the activity state: once an event node becomes active (the corresponding event is in progress), its continued activation depends only on the duration of the activation. Thus, it is assumed that the event duration (or its distribution) is known at the time of recognition. While this assumption works well to reduce the space of feasible solutions for activity recognition, it is only valid in cases where event durations are known to be distributed according to a parametric distribution and a satisfactory amount of labeled training data are available to allow tuning of the duration model parameters. However, in many surveillance applications, these conditions do not hold and this assumption reduces the robustness of this approach to the recognition of activities whose composing event durations have large variance and whose distribution cannot be described by common parametric distributions.

The PFPN approach uses the information encoded in the PN definition of the activity to reduce the state space of the problem. That is, rather than evaluating every possible marking and event pairing at each frame, the PFPN evaluates only those markings that are reachable from the current hypothesis of the state. The possible event occurrences are also limited to only those events that may occur in a given marking. Note that no assumption of a known duration model of the events’ time interval, or the interevent time interval is made. Indeed such a model is not available in any real dataset, rather only in synthetic data such as our bank dataset. However, as can be seen in Fig. 8, this advantage did not enable the P-Net approach to greatly outperform the PFPN approach.

![ROC curve for the Technion Parking Lot dataset](image)

**Fig. 11.** Experiment 3b: ROC curve for the Technion Parking Lot dataset ("ideal" tracking).
VII. CONCLUSION

In this paper, we have described PFPN, an approach to modeling and recognition of activities in video. Activities are defined as a partially ordered set of events. Events are defined as atomic occurrences restricted to a temporal interval. Activities in PFPN are modeled using the PN formalism [3], [11], [23], [29], which enables the modeling of complex temporal relations. These activity specifications are translated into the Bayesian recursive framework (BRF) which allows a probabilistic state estimation using the particle filter algorithm. This estimation enables reasoning on event observations, which is improved up to a certainty value. Furthermore, the state estimation can be updated dynamically with each new observation, enabling recognition of activities as they occur.

Our experiments evaluate the PFPN approach across three datasets: the publicly available ETISEO Building Entrance dataset, the Technion Parking Lot dataset, and a synthetic dataset, which serves to increase the amount of examples our approach is evaluated on. We compare the performance of our approach to two [33], [35] leading approaches, representing the two main types of works in the activity recognition literature (deterministic and stochastic). In contrast with recent activity recognition works, our evaluations isolate the activity recognition component of the system by keeping the low-level video processing and event recognition components constant across all approaches.

Our results reveal the general tendency of deterministic approaches to produce fewer false recognitions, but also fewer true recognitions. When the level of uncertainty present in the data is high, a stochastic approach such as PFPN is shown to improve this tradeoff. Furthermore, we conclude that having a formalism that allows holistic modeling of the application domain (such as PFPN), improves the activity recognition performance without relying on an available duration model for each of the events composing the activities.

In the future, we plan to study the effects of merging several activity models into a single global model of the scene. As many activity models have overlapping information, we hope this kind of approach can both increase recognition performance (by reducing confusion between similar activities) and reduce computation time (by evaluating the possibility of all activities simultaneously).

REFERENCES


Gal Lavee received the B.Sc. and M.Sc. degrees in computer science from the University of North Texas, Denton, and the University of Texas at Dallas, Dallas, respectively. He is currently working toward the Ph.D. degree at the Technion, Israel Institute of Technology, Haifa, Israel. His research interests include video event understanding, knowledge representation, and learning of semantic knowledge.

Michael Rudzsky received the Ph.D. degree in physics and mathematics from the Institute of Space Research, Moscow, Russia, in 1980. He was with the Scientific and Industrial Association for Space Research, Baku, Azerbaijan, until 1990. In 1991, he became a Research Fellow with the Department of Physics, Technion, Haifa, Israel, where he has been with the Department of Computer Science since 1995. His current research interests include computer vision, pattern recognition, and compression of images.

Ehud Rivlin (M’93) received the B.Sc. and M.Sc. degrees in computer science and the M.B.A. degree from the Hebrew University in Jerusalem, Jerusalem, Israel, and the Ph.D. degree from the University of Maryland, College Park.

He is currently a Professor with the Department of Computer Science, Technion, Israel Institute of Technology, Haifa, Israel. His current research interests include machine vision and robot navigation.