IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 25, 2019, accepted August 12, 2019, date of publication August 20, 2019, date of current version September 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2936493

High-Level Video Event Modeling, Recognition, and Reasoning via Petri Net

ZHIJIAO XIAO, JIANMIN JIANG, AND ZHONG MING
College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

Corresponding author: Zhijiao Xiao (cindyxzj@szu.edu.cn)

This work was supported in part by the National Engineering Laboratory of China for Big Data System Computing Technology, in part by the National Natural Science Foundation of China under Grant 61836005 and Grant 61672358, in part by the Natural Science Foundation of Guangdong Province, China, under Grant 2017A030310521, and in part by the Science and Technology Innovation Commission of Shenzhen, China, under Grant JCYJ20160422151736824.

ABSTRACT A Petri net based framework is proposed for automatic high level video event description, recognition and reasoning purposes. In comparison with the existing approaches reported in the literature, our work is characterized with a number of novel features: (i) the high level video event modeling and recognition based on Petri net are fully automatic, which are not only capable of covering single video events but also multiple ones without limit; (ii) more variations of event paths can be found and modeled using the proposed algorithms; (iii) the recognition results are more accurate based on automatic built high level event models. Experimental results show that the proposed method outperforms the existing benchmark in terms of recognition precision and recall. Additional advantages can be achieved such that hidden variations of events hardly identified by humans can also be recognized.

INDEX TERMS Automated video event modeling, video event recognition, video event reasoning, Petri net.

I. INTRODUCTION

With the rapid development of artificial intelligence [1], [2], computerized video content analysis is moving towards high level semantics based approaches, where video event recognition and reasoning remain to be one of the actively researched topics over the past decades. To narrow the gap between low level visual features and high level semantics, existed methods focus on two levels of video event analysis. The low level is to recognize atomic actions. Researches on this area are often key-frame based. Global and local features are extracted from those key frames and semantic concept classifiers are applied to capture crucial patterns for event recognition. There are many ways to extract low level features which have been successfully applied in many areas [3]–[5]. Hasan and Roy-Chowdhury [6] propose a framework for continuous activity learning using deep hybrid feature models and active learning. Samanta and Chanda [7] use three-dimensional facet model to detect and describe space time interest points in videos. Those methods can extract low level semantics for action recognition and event analysis based on key actions or scenes. They give no or less consideration about temporal and logical relations among actions when they are used to classify events. Some improvement researches are done. Wang *et al.* [8] propose a new motion feature to compute the relative motion between visual words and present approaches to select informative features. Cui *et al.* [9] propose a novel unsupervised approach for mining categories from action video sequences. They use pixel prototypes quantized by spatially distributed dynamic pixels to represent video data structuration. Abbasnejad *et al.* [10] present a model based on the combination of semantic and temporal features extracted from video frames. The model is able to detect the events with unknown starting and ending locations.

The work in this paper focuses on the high level of video event recognition. The high level is conducted on the results of the low level to recognize events with complex action sequences. Veeraraghavan and Papanikolopoulos [11] present semi supervised event learning algorithms. The models of events are represented as stochastic context-free grammars. Kitani *et al.* [12] create a hierarchical Bayesian

The associate editor coordinating the review of this manuscript and approving it for publication was Huimin Lu.

network by combining stochastic context-free grammar and Bayesian network. They apply the network on action sequences via deleted interpolation to recognize events. Shet *et al.* [13] use Prolog based reasoning engine to recognize events from log of primitive actions and predefined rules. Song *et al.* [14] present a multi-modal Markov Logic framework for recognizing complex events. Liu *et al.* [15] present an interval-based Bayesian generative network approach to model complex activities. The approach constructs probabilistic interval-based networks with temporal dependencies in complex activity recognition. Song *et al.* [16] present a framework for high-level activity analysis. It consists of multi-temporal analysis, multi-temporal perception layers, and late fusion. The method can handle temporal diversity of high-level activities. Nawaz *et al.* [17] propose a framework for predictive and proactive complex event reasoning. It processes, integrates, and provides reasoning over complex events using the logical and probabilistic reasoning approaches. Skarlatidis *et al.* [18] present a system for recognizing human activity given a symbolic representation of video content. They use a dialect of the Event Calculus for probabilistic reasoning. Cavaliere *et al.* [19] employ semantic web technologies to encode video tracking and classification data into ontological statements. It allows the generation of a high-level description of the scenario through activity detection. By semantic reasoning, the system is able to connect the simple activities into more complex activities. Azorin-Lopez *et al.* [20] propose a predictive method based on a simple representation of trajectories of a person in the scene. It allows a high level understanding of the global human behavior. Their method does not need predefined models and rules to evaluate behaviors.

Since Castel *et al.* [21] introduce Petri nets for high level representation of image sequences, Petri nets and its variations are widely used in modeling video events for their detection and recognition. Petri net is a powerful event model tool that supports the representation of high level events. While places denote different states of objects inside videos, transitions represent switches of states that are usually caused by primitive actions performed by the tracked objects. When such a model is used to recognize an event, each tracked object will be modeled as a token moving in the Petri net model according to its action sequence. If any token reaches the end place of the event model, the event is claimed to have happened. During the tracking process, event reasoning can be done to predict which event has the biggest possibility to happen.

Albanese *et al.* [22] propose an extended Probabilistic Petri Nets. They present the PPN-MPS algorithm to find the minimal sub-videos that contain a given activity with a probability above a certain threshold. Ghanem *et al.* [23] and Ghanem [24] address the advantages of using Petri nets for event recognition. They propose a framework which provides a graphical user interface for user to define objects and primitive events. Then it expresses composite events using logical, temporal and spatial relations. Lavee *et al.* [25] propose the Particle Filter Petri Net to model and recognize activities in videos. They also propose a method to transform semantic descriptions of events in formal ontology languages to Petri net event models [26]. The surveillance event recognition framework they proposed uses a single Petri net for recognition of event occurrences in video. It allows modeling of events having variances in duration and predicting future events probabilistically [27]. Borzin *et al.* [28] present video event interpretation approach using GSPN. Through adding marking analysis into a GSPN model, their methods provide better scene understanding and next marking state prediction using historic data. Ghrab *et al.* [29] present an approach to automatically detect abnormal high-level events in a parking lot. A Petri net model is used to describe and recognize high-level events or scenarios that incorporate simple events with temporal and spatial relations. Hamidun *et al.* [30] translate the event sequence in the crossing scenario to the PN model. The combined effects of spatial and temporal information are analyzed using the steady state analysis built in the model. They point out that modeling with Petri Nets also allows the development of model in hierarchical structure. Szwed [31], [32] proposes Fuzzy Semantic Petri Nets (FSPN) as a tool aimed at solving video event modeling and recognition problems. Linear Temporal Logic is used as a language for events specification and FSPN is used as a tool for recognition. SanMiguel and Martínez [33] use Petri nets in the long-term layer, which is in charge of detecting events with a temporal relation among their counterparts. They extend the basic PN structure to manage uncertainty obtained by the sub-events.

Existing researches focus on event recognitions, video event modeling is often ignored and remains as one of the unsolved research problems. Existing efforts are primarily limited to manual modeling approaches, including knowledge-based or rule-based schemes through semantics extractions. Although significant progress has been achieved, it is stated by many researchers [34]–[37] that automatic event modeling is still a challenge. In this paper, we introduce a high level video event modeling, recognition and reasoning approach based on Petri net to forward the existing state of the arts on Petri net based video event recognition, providing a pioneering framework for computerized high level video content interpretation, analysis and understanding. To this end, our main contribution can be highlighted as:

(i) We systematically propose a Petri net based high level video event description model, which can be expanded for describing any high level video event for video content analysis and semantics organization;

(ii) We present algorithms which can directly build up a video event model from the labeled video training dataset automatically without any intermediate entities such as ontology, etc.

(iii) More variations of event paths can be captured and the recognition results can be more accurate based on automatic built high level event models.

The rest of the paper is organized as follows. Section 2 presents some concepts of the Petri net based video event modeling, in order to pave the way for our proposed work. Section 3 describes our proposed algorithms for high level video event modeling and recognition based on Petri net, and Section 4 reports the experimental results. Comparative analysis of the results is also included in this section to evaluate the performance of the proposed methods, and finally, a conclusion and proposals for future research are addressed in Section 5.

II. PETRI NET BASED VIDEO EVENT MODELING

Petri net is a directed graph constructed with four essential elements: place, transition, arc and token. While the first three elements are used to model the static structures, token is designed to reflect the dynamic states of a Petri net.

Definition 1(PN): A Petri Net is a 5-tuple, PN $= (P, T, F, W, M_0)$ where:

1) $P = \{p_0, p_2, \ldots, p_{n-1}\}$ is a finite set of places;

2) $T = \{t_1, t_2, \ldots, t_{mt}\}$ is a finite set of transitions;

3) $F \sqsubseteq (P \times T) \cup (T \times P)$ is a set of arcs;

4) W: F-> $(1, 2, 3, \ldots)$ is a weight function;

5) M_0: P-> $\{0, 1, 2, 3, \ldots\}$ is the initial marking;

6) $P \cap T = \Phi$ and $P \cup T \neq \Phi$

For a detailed Petri net introduction, we refer to [38].

Let the places representing possible states of tracked objects, the transitions representing possible primitive actions of tracked objects, and tokens standing for tracked objects, we can define a single event model(SE-Tree) as follows based on the concept of a classical Petri net.

Definition 2 (SE-Tree): A *PN* is a SE-Tree if and only if:

1) There exists one and only one source place $p_0 \in P$, and $\cdot p_0 = \Phi$, and for $\forall p \in P$-$\{p_0\}$, $|\cdot p| = 1$;

2) $P_\mathrm{e} \subset P$ is a finite set of end places. An end place denotes a final state of the object that conducted an event.

3) If p is an end place, $|p \cdot| \geq 0$; otherwise, $|p \cdot| \geq 1$;

4) For $\forall t \in T$, $|t \cdot| = 1$ and $|\cdot t| = 1$;

Since there are many uncertainties inside an event, a SE-Tree often needs to model all its possible variations, and each of such variations is referred as an instance. To provide efficient and effective coverage of all the possible uncertainties, we introduce the concept of a path to describe the route of an event instance.

Definition 3 (Path): A Path, $path =< p_0, t_0, p_1, \ldots, t_i, p_{i+1} \ldots, t_{n1-1}, p_{n1} >$, is a sequence of nodes, which connects the source place p_0 to one of the end place $p_{n1} (p_{n1} \in P_\mathrm{e})$.

A SE-Tree modeling all paths of an event has a tree structure. We choose the tree structure other than net because there is less ambiguity and inaccuracy. For example, as shown in Fig. 1(a), there are two paths to accomplish a certain event, i.e. $path_1 =< p_0, t_{01}, p_1, t_{12}, p_2, t_{23}, p_3 >$, and $path_2 =< p_0, t_{04}, p_4, t_{42}, p_2, t_{25}, p_5 >$. Suppose there is an object which goes through a path such as: $path_3 =< p_0, t_{01}, p_1, t_{12}, p_2, t_{25}, p_5 >$, it will be misjudged that the specific event has happened in terms of net representation. But if the tree structure is used as shown in Fig. 1(b), however, no path like $path_3$ could be included in the model. For the same reason, the loop structures are converted to sequence structures.

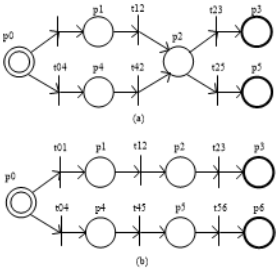


FIGURE 1. Examples of event model based on different structures.

Based on the concept of SE-Tree as described above, a Petri net based multi event model is defined as follows.

Definition 4(ME-Tree): A *PN* is a ME-Tree if and only if:

1) Each $p \in P$ has an attribute called *par_event*. The value of this attribute is the ids of all possible events that the place participated;

2) Each $p \in P_\mathrm{e}$ has an attribute called *end_event*. The value of this attribute is the id of the most possible event that the place is the end place.

III. HIGH LEVEL VIDEO EVENT MODEL BUILDING

The process of high level video event model building is depicted in Fig. 2.

To build a high level event model, a certain amount of video segments containing specific high level events should be prepared as the training dataset. Here we suppose that all objects and their primitive actions and states are recognized and target events are labeled. Our work focuses on the last two steps. The symbols used are explained in Table 1.

A. SINGLE EVENT MODEL BUILDING

The places and transitions of SET_k are created based on following rules.

Rule 1: A process of creation will be fired for f_u in a video segment if and only if:

1) $\exists v\, (FGF\,(f_u, o_v, event) = e_k)$, and

2) $FGO\,(o_v, state) \neq FGF\,(f_u, o_v, state)$, and

3) $\neg\exists j \left(p_i = FGO(o_v, place) \wedge t_{ij} \in p_i \bullet \wedge p_j \in t_{ij} \bullet \wedge FGP\left(p_j, label\right) = FGF\,(f_u, o_v, state)\right)$

Condition (1) requires a tracked object o_v conducting e_k; (2) means that there is a change of o_v's state in f_u; (3) denotes that there is no output transitions t_{ij} of p_i (the place that o_v stay currently) whose output place p_j has the same label as the new state of o_v.

If a tracked object o_v is conducting event e_k, all its states and switches of states will be modeled. The creation will first trigger the creation of a new place p_j which describes the new state of the object inside f_u. After the new place is created, a new transition t_{ij} will be created to connect the two places p_i and p_j, while p_i denotes the old state and p_j denotes the new state of the object o_v. After the creation, the token

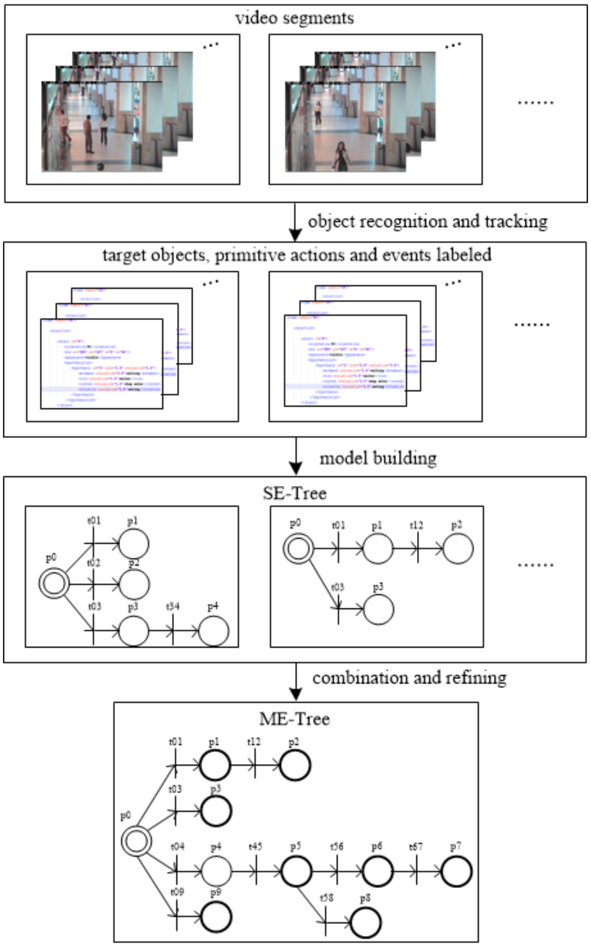


FIGURE 2. Process of video event model building.

```
Algorithm 1 SE-Tree Building (Single Even Modeling)
Input: $e_k$: the $k$th event needed to be modeled
FQ: A queue of frames of a video segment containing the event $e_k$
Output: $SET_k$: A SE-Tree of Event $e_k$
Initialize empty $P, T, P \times T$ and $T \times P$ of se
Create a place $p_0$ as the source place and add it to $P$
$FSP(p_0, label,$ "unknown")      //initialize the label of $p_0$
while $FQ$ not empty do
    $f_u = FQ.pop()$
    for each object $o_v$ conducting event $e_k$ in $f$ do
        if $o_v$ just appears
            create a new token $o_v$
            add $o_v$ to $p_0$
            $FSO\ (o_v, place, p_0)$    //initialize the stay place of $o_v$
            $FSO\ (o_v, state,$ "unknown")    //initialize the state of $o_v$
            $TP_{0k} + +$
        end
        $p_i = FGO(o_v,\ place)$
        if rule 1 is satisfied
            create a new place $p_j$ and $FSP(p_j, label, FG_f(f_u, o_v, state))$      //set the label of $p_j$
            add $p_j$ to $P$
            create a new transition $t_{ij}$, add $t$ to $T$
            create a new arc $pt$ from $p_i$ to $t_{ij}$, add $pt$ to $PT$
            create a new arc $tp$ from $t_{ij}$ to $p_j$, add $tp$ to $TP$
        else if only the third condition in rule 1 is broken
            find $p_j$ that breaks the third condition in rule1
        end
        $FSO(o_v, place, p_j)$      //update the stay place of $o_v$
        $FSO(o_v, state, FG_p(p_j, label))$    //update the state of $o_v$
        $TP_{jk} + +$
        $TT_{ijk} + +$
        if $o_v$ ends in $p_j$
            $TEP_j + +$
            $TEP_{jk} + +$
        end
        if rule 2 is satisfied
            add $p_j$ to $P_{\mathrm{e}}$
        end
    end
end
```

representing o_v will be moved from p_i to p_j, and the values of o_v's attributes of o_v will be updated.

Rule 2: A place is marked as an end place in SET_k if and only if the place is the final state of an object conducted e_k.

In order to support the certainty reasoning of video event, the probability distribution over the places in an event model needs to be learned to estimate the likelihood of an event's occurrence. The number of tokens that once stay in each place and the number of tokens that fire each transition will be counted. If a place is marked as an end place, it will count the number of tokens who end in this place and which event each token ends. Specific values of those numbers are learned for generating the probability of the event's occurrence.

Based on the above rules, the algorithm for SE-Tree building can be described as follows.

The computational complexity of the algorithm for SE-Tree building depends on the number of frames in a video segment and the number of tracked objects conducting event e_k in each frame. Given on_i represents the number of objects conducting event e_k in the ith frame of a video ($i = 1, 2, \ldots fm, fm$ is the number of frames in a video segment), the computational complexity of Algorithm 1 is $O(\sum_{i=1}^{fm} on_i)$. Give N clips of videos, the computational complexity of SE-Tree building would be $O(\sum_{i=1}^{N} \left(\sum_{i=1}^{fm_j} on_i\right))$, where fm_j is the number of frames in the jth$(j = 1, 2, \ldots N)$ video segment.

B. MULTI EVENT MODEL BUILDING

A ME-Tree can be built up by combining and refining several given single event models. Not all places in all SE-Trees are added to ME-Tree. Those duplicated places will be composed

TABLE 1. List of symbols and their descriptions.

Symbols	Descriptions
n	The number of places
P	The set of places which includes all objects' available states
p_i	The ith place, i=0, ..., n-1
P_e	$P_\mathrm{e} \subset P$ is a finite set of end places
T	The set of transitions which includes primitive actions or changes of objects' states
t_{ij}	The transition connecting p_i to p_j
$\bullet p_i$	$\{t_{ji} \vert <t_{ji}, p_i> \in T\times P\}$ is the set of input transitions of p_i
$p_i\bullet$	$\{t_{ij} \vert < p_i, t_{ij}> \in P\times T\}$ is the set of output transitions of p_i
$\bullet t_{ij}$	$\{p_i \vert <p_i, t_{ij}> \in P\times T\}$ is the input place of t_{ij}
$t_{ij}\bullet$	$\{p_j \vert <t_{ij}, p_j> \in T\times P\ \}$ is the output place of t_{ij}
o_v	The vth token, which is the vth tracked object
m	The number of events
e_k	The kth event, k=0, ..., m-1
f_u	The uth frame of a video segment
SET_k	The SE-Tree for e_k
TP_i	Number of tokens that pass by p_i
TP_{ik}	Number of tokens that pass by p_i and conduct event e_k
TEP_i	Number of tokens that end in p_i
TEP_{ik}	Number of tokens that end in p_i and conduct event e_k
TT_{ij}	Number of tokens that pass by t_{ij}
TT_{ijk}	Number of tokens that pass by t_{ij} and conduct event e_k
PE_i	Probability of an token ended in p_i
PP_{ik}	Probability of a token reached in p_i who will conduct event e_k
PEE_{ik}	Probability of a token ended in p_i who may conduct event e_k
PT_{ij}	Fire probability of t_{ij} who connects p_i to p_j for a token that passes by p_i and not ends in p_i
S_{vu}	State of o_v in f_u
SC_v	Current state of o_v
$FGF\ (f_u, o_v, w)$	the function which can get the value of the attribute w of o_v in f_u
$FGP(p_i, w)$	the function which can get the value of the attribute w of p_i
$FGO(o_v, w)$	the function which can get the latest value of the attribute w of o_v
$FSO(o_v, w, a)$	the function which can set the value of the attribute w of o_v to a
$FSP(p_i, w, a)$	the function which can set the value of the attribute w of p_i to a

into one place to simplify the combined model. Let p' be the current place considered in SET_k, p_i is the corresponding place of p' in MET, a new place will only be created in MET according to the following rule.

Rule 3: A process of creation will be fired for MET if and only if:

(1) $\exists t' \left(t' \in T_k \wedge t' \in p'_\blacksquare\right)$, and

(2) $\neg\exists j \left(t_{ij} \in T \wedge p_j \in P \wedge (t_{ij} \in p_i\, _\blacksquare \wedge t_{ij} \in {}_\blacksquare p_j) \wedge \left(FGP\left(p_j, label\right) = FGP\left(t'_\blacksquare, label\right)\right)\right)$

Condition (1) considers if there is a transition which is an output transition of current place p'. If condition (1) is satisfied, for an output place of t', condition (2) checks if the corresponding place has already existed in MET that is derived from the same source place p_i (p' in SET_k is modeled as p_i in MET). According to rule 3, those transitions connecting the same source and destination places will only be added to the multi event model once.

The algorithm for ME-Tree building is described as follows.

According to Algorithm 2, the ME-Tree will be simplified by eliminating those unnecessary nodes as soon as the number of events in the checking list of *end_event* is narrowed to a unique one.

The computational complexity of the algorithm for ME-Tree building depends on the number of SE-Trees and the number of places and transitions in each SE-Tree. Given tn_i represents the number of transitions in the ith SE-Tree($i = 1, 2, \ldots m, m$ is the number of SE-Trees, i.e. The number of events), the computational complexity of Algorithm 2 is O($\sum_{i=1}^{m} tn_i$).

Based on the numbers learned during single event modeling, the probabilities for event reasoning can be calculated as follows.

$$PP_{ik} = \begin{cases} PEE_{ik}, & p_{i\blacksquare} = \emptyset \\ PEE_{ik} \times PE_i + (1 - PE_i) & \\ \times \sum_{t_{ij} \in p_{i\blacksquare}} (PT_{ij} \times PP_{jk}), & others \end{cases} \quad (1)$$

Algorithm 2 ME-Tree Building (Multi Events Modeling)

```
Input: $SET$: A set of single event models, each of which models a single event.
Output: $MET$: A ME-Tree
Initialize empty $P$, $T$, $P \times T$ and $T \times P$ of $MET$
Create a place $p_0$ as the source place and add it to $P$
$p = p_0$
for each $SET_k \in SET$ do
   $p' = p'_0$        // $p'_0 \in SET_k$
   if $p' \bullet \neq \Phi$
      creation($p_0$, $p'$)
   end
end
creation($p_i$, $p'$)
for each $t' \in p' \bullet$
   if rule 3 is satisfied,
      create a new place $p_j$
      add $p_j$ to $P$
      add the information of $se$ to update the $par\_event$ of $p_j$
      create a new transition $t_{ij}$, add $t_{ij}$ to $T$
      create a new arc $pt$ from $p_i$ to $t_{ij}$, and add $pt$ to $P \times T$
      create a new arc $tp$ from $t_{ij}$ to $p_j$, and add $tp$ to $T \times P$
   else
      find $p_j$ that breaks the second condition in rule 3
   end
   $p' = t' \bullet$
   if $p' \in P_e'$
      add $p_j$ to $P_e$
      add the information of $se$ to update the $end\_event$ of $p$
      search other models in $SE$
      if there is no path through which a token can reach $p'$
         break;
      end
   end
   if $p' \bullet \neq \Phi$
      creation($p_j$, $p'$)
   end
end
```

where

$$PT_{ij} = \frac{TT_{ij}}{TP_i - TEP_i} \tag{2}$$

$$PE_i = \frac{TEP_i}{TP_i} \tag{3}$$

$$PEE_{ik} = \frac{TEP_{ik}}{TEP_i} \tag{4}$$

$$TT_{ij} = \sum\nolimits_{k=0}^{m-1} TT_{ijk} \tag{5}$$

$$TP_i = \sum\nolimits_{k=0}^{m-1} TP_{ik} \tag{6}$$

$$TEP_i = \sum\nolimits_{k=0}^{m-1} TEP_{ik} \tag{7}$$

$$\sum\nolimits_{j=0}^{n-1} PT_{ij} = 1 \tag{8}$$

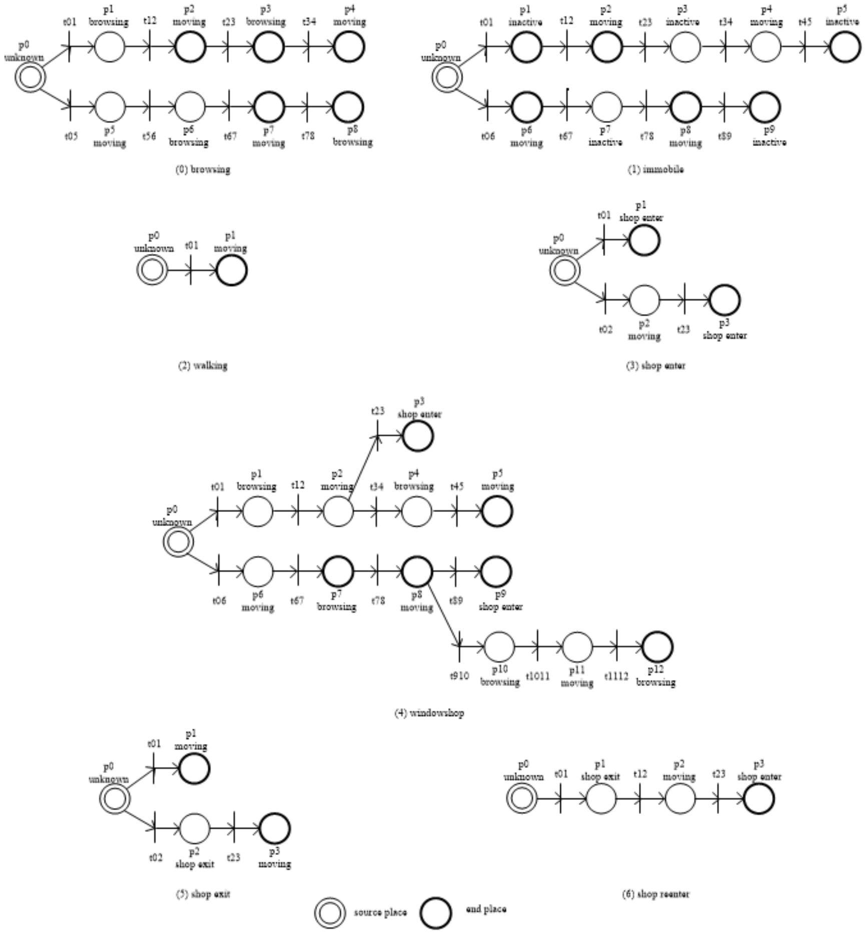
Algorithm 3 MER(Multi Events Recognition)

```
Input: $ES$: A set of events needed to be detected
$MET$: A ME-Tree
$FQ$: A queue of frames of a video segment with recognized objects and recognized states of each object in each frame
Output: $EN$: An array containing numbers of happened events
for each event $e_k \in ES$ do
   $EN_k = 0$
end
while $FQ$ not empty do
   $f_u = FQ.pop()$
   for each object $o_v$ in $f_u$ do
      if $o_v$ just appear
         create a new token $T(o_v)$
         add $T(o_v)$ to $p_0$ and initialize the attribute values of $o_v$
      end
   end
   for each $t_{ij} \in T$ do
      if $t_{ij}$ is enabled according to rule 4
         fire $t_{ij}$ and change marking according to rule 5
         update attribute values of objects enabled the firing
         for each object $o_v$ enabled the firing in $f_u$ do
         $FSO(o_v, pos\_event, FGP(FGO(o_v, place), par\_event))$
               //set the possible event of $o_v$
         if $FGO(o_v, place) \in P_e$     //$o_v$ reached an end place
            $FSO(o_v, event, FGP(FGO(o_v, place), end\_event))$
                  //set the happened event of $o_v$
         end
         end
      end
   end
end
for each object $o_v$ do
   if $FGO(o_v, event)$ is empty
      $FSO(o_v, event, FGO(o_v, \mathrm{MOST}(pos\_event)))$
      // set the happened event of $o_v$ as the most possible event
   end
   for each event $e_k == FGO(o_v, event)$ do
      $EN_k{+}{+}$
   end
end
return $EN$
```

$$\sum\nolimits_{k=0}^{m-1} PEE_{ik} = 1 \tag{9}$$

$$\sum\nolimits_{k=0}^{m-1} PP_{ik} = 1 \tag{10}$$

This information about event reasoning is added as attributes' values of places and transitions of a ME-Tree, which can be used to make predictions.



(0) browsing

(1) immobile

(2) walking

(3) shop enter

(4) windowshop

(5) shop exit

(6) shop reenter

FIGURE 3. SETs for seven events.

C. VIDEO EVENT RECOGNITION AND REASONING

A marking is a distribution of tokens over places in a ME-Tree. A token in a ME-Tree denotes a tracked object. The events that tracked objects in a video segment have conducted or will conduct can be recognized by observing the distribution of tokens in the ME-Tree. The initial marking has no token. As soon as a tracked object appears in a video segment, a token will be added to the source place. This will change the marking. The marking will also be changed along with the moving of tokens. The firing of transitions will cause the moving of tokens. A transition is enabled if and only if there are tokens in the input place of the transition as defined in rule 4.

Rule 4: t_{ij} is enabled if and only if $M\left(p_i\right) \geq 0 \wedge G\left(t_{ij}\right) = true$.

$M(p_i)$ is the number of tokens stay in p_i under marking M. $G(t_{ij})$ denotes the set of the guard functions on transition t_{ij} for firing. Here, the main guard function for firing a transition is a state change caused by a primitive action of an object or a group of objects.

If a change of the vth object's state is detected in current frame, the transition will be fired. The token representing the vth object will be moved from p_i to p_j with the firing of t_{ij}.

TABLE 2. Seven contexts.

id	event	id	event
0	browsing	4	windowshop
1	immobile	5	shop exit
2	walking	6	shop reenter
3	shop enter		

After the firing of transition t_{ij}, a new marking is generated according to rule 5.

Rule 5: t_{ij} is fired for the vth object in the uh frame and the marking M is replaced by a new marking M' produced according to Eq.(11) and (12).

$$M'\left(p_i\right) = M\left(p_i\right) - 1 \tag{11}$$

$$M'\left(p_j\right) = M\left(p_j\right) + 1 \tag{12}$$

An instance of an event is happened if there is a token(an object) reached one of the end places.

The probabilities for event reasoning can be updated and deduced for each object in each frame. The algorithm for event recognition based on a ME-Tree can be described as follows.

TABLE 3. Attributes of places.

Id	*label*	*end place*	*end_event*	*PE_i*	*PEE_{ik}*							*PP_{ik}*						
					0	1	2	3	4	5	6	0	1	2	3	4	5	6
0	unknown	/	/	0	0	0	0	0	0	0	0	0.0323	0.0484	0.2339	0.3065	0.0968	0.2742	0.0081
1	browsing	/	/	0	0	0	0	0	0	0	0	0.5000	0	0	0	0.5000	0	0
2	moving	√	0	0.1667	1.0000	0	0	0	0	0	0	0.5000	0	0	0	0.5000	0	0
3	shop enter	√	4	1.0000	0	0	0	0	1.0000	0	0	0	0	0	0	1.0000	0	0
4	browsing	√	0	0.2500	1.0000	0	0	0	0	0	0	0.5000	0	0	0	0.5000	0	0
5	moving	√	4	1.0000	0.3333	0	0	0	0.6667	0	0	0.3333	0	0	0	0.6667	0	0
6	inactive	√	1	1.0000	0	0	0	0	1.0000	0	0	0	0	0	0	1.0000	0	0
7	shop exit	/	/	0	0	0	0	0	0	0	0	0	0	0	0	0	0.9697	0.0303
8	moving	√	5	0.9697	0	0	0	0	0	1.0000	0	0	0	0	0	0	0.9697	0.0303
9	shop enter	√	6	1.0000	0	0	0	0	0	0	1.0000	0	0	0	0	0	0	1.0000
10	shop enter	√	3	1.0000	0	0	0	1.0000	0	0	0	0	0	0	1.0000	0	0	0
11	moving	√	2	0.4286	0	0.0606	0.8788	0	0	0.0606	0	0.0130	0.0779	0.3766	0.4416	0.0649	0.0260	0
12	shop enter	√	3	1.0000	0	0	0	1.0000	0	0	0	0	0	0	1.0000	0	0	0
13	browsing	√	4	0.1667	0	0	0	0	1.0000	0	0	0.1667	0	0	0	0.8333	0	0
14	moving	√	0	0.4000	0.5000	0	0	0	0.5000	0	0	0.2000	0	0	0	0.8000	0	0
15	browsing	√	0	0	0	0	0	0	0	0	0	0	0	0	0	1.0000	0	0
16	moving	/	/	0	0	0	0	0	0	0	0	0	0	0	0	1.0000	0	0
17	browsing	√	4	1.0000	0	0	0	0	1.0000	0	0	0	0	0	0	1.0000	0	0
18	shop enter	√	4	1.0000	0	0	0	0	1.0000	0	0	0	0	0	0	1.0000	0	0
19	inactive	/	/	0	0	0	0	0	0	0	0	0	1.0000	0	0	0	0	0
20	moving	√	1	1.0000	0	1.0000	0	0	0	0	0	0	1.0000	0	0	0	0	0

The computational complexity of the algorithm mainly depends on the number of frames in a video segment, the number of objects in each frame. Given on_i represents the number of objects in the ith frame of a video($i = 1,2,\ldots fm$, fm is the number of frames in a video segment), the computational complexity of Algorithm 3 is O($\sum_{i=1}^{fm} on_i$).

IV. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we report experimental results to demonstrate the performance of the proposed method. In order to compare our method with the existing research, we select the published work [27] using manual event model as the benchmark and the CAVIAR [39] dataset as one of our experimental dataset. The CAVIAR dataset contains 52 clips of videos of a shopping center in Lisbon. This set of sequences contains 1500 frames on average. The ground truth and labels are provided. Half of videos in a dataset are randomly chosen as the training dataset, and the other half are used as the test dataset.

A. RESULTS OF AUTOMATIC SINGLE EVENT MODEL BUILDING

Seven different contexts are considered here which are numbered and referred as given in Table 2. The ground truth information, including context and situation information provided by the CAVIAR dataset, is used to build SE-Tree for each high level event, from which the context of each object is treated as a video event, and the situation information is used to define the contexts. Seven SE-Trees built for those events are shown in Fig. 3.

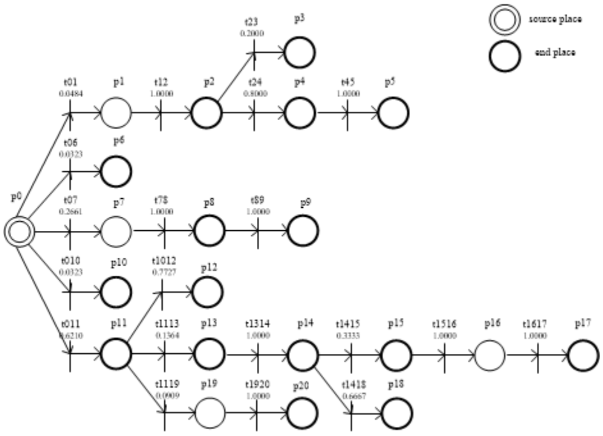


FIGURE 4. ME-Tree for CAVIAR contexts.

B. RESULTS OF AUTOMATIC MULTI EVENTS MODEL BUILDING

After all the SE-Trees are built from the training dataset, algorithm for building ME-Tree will be called to create a ME-Tree for all events involved. The ME-Tree built up in

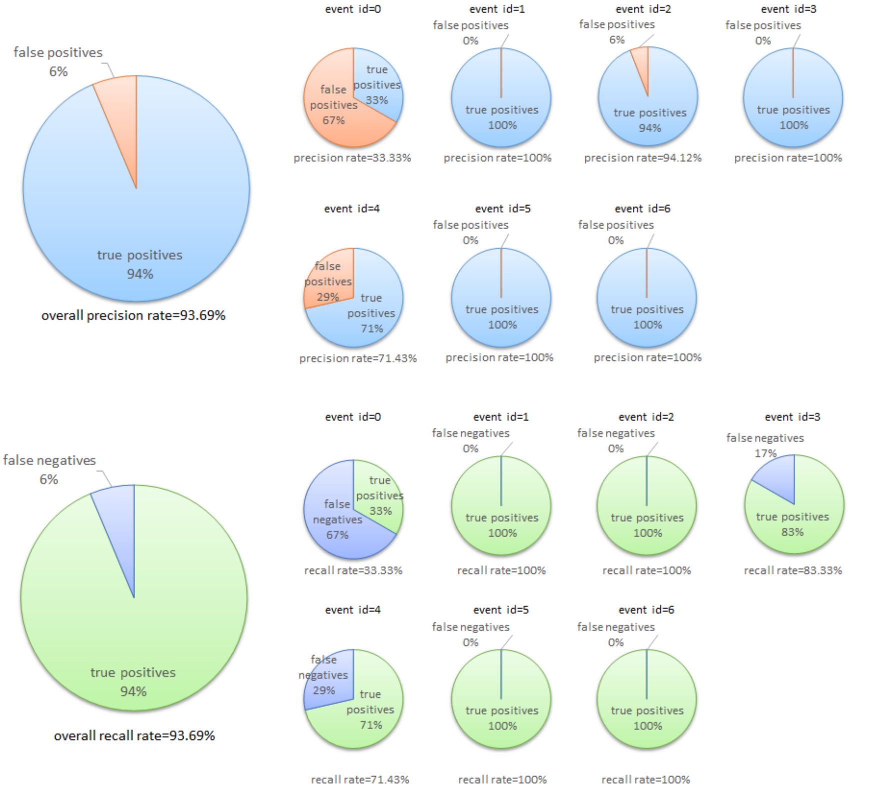


FIGURE 5. Event classification results based on proposed model.

this way is depicted in Fig. 4. The source place and the end places are marked in Fig. 4. The values of attribute PT_{ij} of transitions are also labeled in Fig. 4. Other details are described in Table 3, which lists all the attributes of places. As mentioned in Definition 4, the value of the *end_event* attribute is the id of the most possible event that ends in the place.

Compared to the model depicted in [27], our model captures more hidden variations of events. For example, there is no direct "unknown" to "shop enter" sequence for "shop enter" in the model presented in [27], which is captured by our method. As a result, precision rates and recall rates for event recognition are both improved. The reason is that there is less manual intervention during the building process of event models using the proposed method, which can find some hidden variations of events that are often ignored by human.

As stated in Section 3.2, the complexity of ME-tree depends on two factors: the number of SE-Trees and the number of places and transitions in each SE-tree. A ME-Tree will combine and refine SE-Trees. If SE-Trees have similar or same paths or sub-paths, the ME-Tree will combine them. It will decrease the scale of the model. But under this circumstance, event recognition will be more difficult due to the often conflicts.

If SE-Trees contain many different paths from each other, the complexity of ME-tree will increase since there are few nodes can be combined. But event recognition will be much easier. If those paths have unique actions to distinguish themselves from others, then the rest nodes of those paths after the unique actions can be eliminated according to Algorithm 2, which can bring down the complexity of the model. However, if the unique actions appear at the end or near the end of the path, then there are not too many nodes left to be eliminated.

Another limitation of the proposed framework is that it only considers events that involve one object. But there are many events in reality that involve more than one object. To model this kind of event, extensions should be done on the proposed framework.

C. RESULTS OF VIDEO EVENT RECOGNITION BASED ON DIFFERENT MODELS

A token will be created and added to the ME-Tree for each tracked object. According to the firing rules described in section 3.3, tokens will be moved from places to places. An instance of an event is happened if there is a token reaches one of the end places. Fig. 5 shows the recognition results of our proposed methods applied to the test dataset.

As seen in Fig. 5, there exist some false negatives and false positives, especially for "browsing" and "windowshop". The main reason is that they are similar events, which lead to similar sub-paths of the two events in the model. Actually, it is also very hard for human to distinguish the two events.

TABLE 4. Comparison of recognition results.

event id	precision rate		recall rate	
	result reported in [27]	result of our method	result reported in [27]	result of our method
0	0.0000	0.5714	0.0000	0.5714
1	1.0000	1.0000	0.1250	0.8750
2	0.8333	0.9167	0.9740	1.0000
3	0.9630	1.0000	0.9286	0.9464
4	0.5455	0.8000	0.8000	0.8000
5	0.9500	1.0000	0.9661	0.9661
6	0.8333	1.0000	1.0000	1.0000
overall accuracy		result reported in [27]		0.8638
		result of our method		0.9447

TABLE 5. Comparison of recognition performance.

Performance	non-Petri net based method	Petri net based methods	
	result reported in [20]	result reported in [27]	result of our method
Sensitivity	0.772	0.864	0.945
Specificity	0.962	0.978	0.991
Accuracy	0.935	0.962	0.984

When a conflict occurs, we choose the event with the largest probability according to our model building algorithm.

In order to compare to the results reported in [27] on the same test dataset, we conduct event recognition on the whole dataset(including training dataset and test dataset). Table 4 compares the recognition results of our proposed method to the results listed in [27].

Since our model captures more event paths, there are improvements in terms of precision rates, recall rates, and the overall accuracy. The results of "browsing" and "windowshop" are also the worst due to the similar paths in the model. According to the experimental results, we can draw a conclusion that the more unique actions the events have, the more accuracy the results are. The reason is less similar paths will exist among those events in the model if each event has its own unique actions.

We also conduct a series of experiments to compare the sensitivity, specificity, and accuracy of Petri net based methods and non-Petri net based method [20]. The definitions of the three performance indicator can be found in [20].

As shown in Table 5, our method outperforms the other two methods, while both Petri net based methods outperform the non-Petri net based one. The main reason is the method in [20] uses a pattern recognition approach assigning an event for just a trajectory instead of a predefined, semantic model of event. Hence, the semantic gap between the input and the output could be very large.

V. CONCLUSION

A framework for high level video event modeling, recognition and reasoning is proposed in this paper to facilitate the video content analysis. Experimental results show that Petri net based model is a good choice for high level event modeling. The proposed method can capture variants of paths inside a high level event and achieve improved performances in comparison with the benchmark. The accuracy of event recognition is improved based on the proposed automatic model in comparison with that based on manual models.

With a simple extension of our method, it can be applied to build models of multiple high level events that involve more than one object. That is one of the future research directions. In addition, the proposed method provides an efficient tool for managing video content analysis and interpretation in terms of high level events, leading to potential applications for high level understanding of the video content by computers. How to apply the proposed method into those areas deserves further efforts.

REFERENCES

[1] H. Lu, Y. Li, M. Chen, H. Kim, and S. Serikawa, "Brain intelligence: Go beyond artificial intelligence," *Mobile Netw. Appl.*, vol. 23, no. 2, pp. 368–375, Apr. 2018.

[2] H. Lu, Y. Li, S. Mu, D. Wang, H. Kim, and S. Serikawa, "Motor anomaly detection for unmanned aerial vehicles using reinforcement learning," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2315–2322, Aug. 2018.

[3] H. Lu, D. Wang, Y. Li, J. Li, X. Li, H. Kim, S. Serikawa, and I. Humar, "CONet: A cognitive ocean network," *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 90–96, Jun. 2019.

[4] S. Serikawa and H. Lu, "Underwater image dehazing using joint trilateral filter," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 41–50, 2014.

[5] H. Lu, Y. Li, T. Uemura, H. Kim, and S. Serikawa, "Low illumination underwater light field images reconstruction using deep convolutional neural networks," *Future Gener. Comput. Syst.*, vol. 82, pp. 142–148, May 2018.

[6] M. Hasan and K. A. Roy-Chowdhury, "A continuous learning framework for activity recognition using deep hybrid feature models," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 1909–1922, Nov. 2015.

[7] S. Samanta and B. Chanda, "Space-time facet model for human activity classification," *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1525–1535, Oct. 2014.

[8] F. Wang, Z. Sun, Y.-G. Jiang, and C.-W. Ngo, "Video event detection using motion relativity and feature selection," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1303–1315, Aug. 2014.

[9] P. Cui, F. Wang, L.-F. Sun, J.-W. Zhang, and S.-Q. Yang, "A matrix-based approach to unsupervised human action categorization," *IEEE Trans. Multimedia*, vol. 14, no. 1, pp. 102–110, Feb. 2012.

[10] I. Abbasnejad, S. Sridharan, S. Denman, C. Fookes, and S. Lucey, "Complex event detection using joint max margin and semantic features," in *Proc. Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA)*, Gold Coast, QLD, Australia, Nov./Dec. 2016, pp. 1–8.

[11] H. Veeraraghavan and P. N. Papanikolopoulos, "Learning to recognize video-based spatiotemporal events," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 628–638, Dec. 2009.

[12] K. M. Kitani, Y. Sato, and A. Sugimoto, "Deleted interpolation using a hierarchical Bayesian grammar network for recognizing human activity," in *Proc. IEEE Int. Workshop Vis. Surveill. Perform. Eval. Tracking Surveill.*, Beijing, China, Oct. 2005, pp. 239–246.

[13] V. D. Shet, D. Harwood, and L. S. Davis, "VidMAP: Video monitoring of activity with prolog," in *Proc. IEEE Conf. Adv. Video Signal Based Surveill. (AVSS)*, Sep. 2005, pp. 224–229.

[14] Y. C. Song, H. Kautz, J. Allen, M. Swift, Y. Li, J. Luo, and C. Zhang, "A Markov logic framework for recognizing complex events from multimodal data," in *Proc. 15th ACM Int. Conf. Multimodal Interact. (ICMI)*, Dec. 2013, pp. 141–148.

[15] L. Liu, S. Wang, B. Hu, Q. Qiong, J. Wen, and D. S. Rosenblum, "Learning structures of interval-based Bayesian networks in probabilistic generative model for human complex activity recognition," *Pattern Recognit.*, vol. 81, pp. 545–561, Sep. 2018.

[16] D. Song, C. Kim, and S.-K. Park, "A multi-temporal framework for high-level activity analysis: Violent event detection in visual surveillance," *Inf. Sci.*, vol. 447, pp. 83–103, Jun. 2018.

[17] F. Nawaz, N. K. Janjua, and O. K. Hussain, "PERCEPTUS: Predictive complex event processing and reasoning for IoT-enabled supply chain," *Knowl.-Based Syst.*, vol. 180, pp. 133–146, Sep. 2019.

[18] A. Skarlatidis, A. Artikis, J. Filippou, and G. Paliouras, "A probabilistic logic programming event calculus," *Theory Pract. Logic Program.*, vol. 15, no. 2, pp. 213–245, Mar. 2015.

[19] D. Cavaliere, V. Loia, A. Saggese, S. Senatore, and M. Vento, "A human-like description of scene events for a proper UAV-based video content analysis," *Knowl.-Based Syst.*, vol. 178, pp. 163–175, Aug. 2019.

[20] J. Azorin-Lopez, M. Saval-Calvo, A. Fuster-Guillo, and J. Garcia-Rodriguez, "A novel prediction method for early recognition of global human behaviour in image sequences," *Neural Process. Lett.*, vol. 43, no. 2, pp. 363–387, Apr. 2016.

[21] C. Castel, L. Chaudron, and C. Tessier, "What is going on? A high-level interpretation of a sequence of images," in *Proc. ECCV Workshop Conceptual Descriptions Images*, Cambridge, U.K., 1996, pp. 13–27.

[22] M. Albanese, R. Chellappa, V. Moscato, A. Picariello, V. S. Subrahmanian, P. Turaga, and O. Udrea, "A constrained probabilistic Petri net framework for human activity detection in video," *IEEE Trans. Multimedia*, vol. 10, no. 6, pp. 982–996, Oct. 2008.

[23] N. Ghanem, D. DeMenthon, D. Doermann, and L. Davis, "Representation and recognition of events in surveillance video using Petri nets," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop (CVPRW)*, Jun./Jul. 2004, p. 112.

[24] N. Ghanem, "Petri net models for event recognition in surveillance videos," Ph.D. dissertation, Dept. Comput. Sci., Univ. Maryland, College Park, MD, USA, 2007.

[25] G. Lavee, M. Rudzsky, and E. Rivlin, "Propagating certainty in Petri nets for activity recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 2, pp. 337–348, Feb. 2013.

[26] G. Lavee, A. Borzin, E. Rivlin, and M. Rudzsky, "Building Petri nets from video event ontologies," in *Proc. Int. Symp Vis. Comput. (ISVC)*, vol. 4841, 2007, pp. 442–451.

[27] G. Lavee, M. Rudzsky, E. Rivlin, and A. Borzin, "Video event modeling and recognition in generalized stochastic Petri nets," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 1, pp. 102–118, Jan. 2010.

[28] A. Borzin, E. Rivlin, and M. Rudzsky, "Surveillance event interpretation using generalized stochastic Petri nets," in *Proc. 8th Int. Workshop Image Anal. Multimedia Interact. Services (WIAMIS)*, Jun. 2007, p. 4.

[29] N. B. Ghrab, R. R. Boukhriss, E. Fendri, and M. Hammami, "Abnormal high-level event recognition in parking lot," in *Intelligent Systems Design and Applications. ISDA* (Advances in Intelligent Systems and Computing), vol. 736, A. Abraham, P. Muhuri, A. Muda, and N. Gandhi, Eds. Cham, Switzerland: Springer, 2018, pp. 389–398.

[30] R. Hamidun, N. E. Kordi, I. R. Endut, S. Z. Ishak, and M. F. M. Yusoff, "Estimation of illegal crossing accident risk using stochastic Petri nets," *J. Eng. Sci. Technol.*, vol. 10, pp. 81–93, Aug. 2015.

[31] P. Szwed, "Modeling and recognition of video events with fuzzy semantic Petri nets," in *Knowledge, Information and Creativity Support Systems: Recent Trends, Advances and Solutions* (Advances in Intelligent Systems and Computing), vol. 364, A. M. J. Skulimowski and J. Kacprzyk, Eds. Cham, Switzerland: Springer, 2016, pp. 507–518.

[32] P. Szwed, "Video event recognition with fuzzy semantic Petri nets," in *Man-Machine Interactions 3* (Advances in Intelligent Systems and Computing), vol. 242, D. Gruca, T. Czachórski, and S. Kozielski, Eds. Cham, Switzerland: Springer, 2014, pp. 431–439.

[33] J. C. SanMiguel and J. M. Martínez, "A semantic-based probabilistic approach for real-time video event recognition," *Comput. Vis. Image Understand.*, vol. 116, no. 9, pp. 937–952, 2012.

[34] L. Liu, S. Wang, G. Su, B. Hu, Y. Peng, Q. Xiong, and J. Wen, "A framework of mining semantic-based probabilistic event relations for complex activity recognition," *Inf. Sci.*, vols. 418–419, pp. 13–33, Dec. 2017.

[35] K. Kardas and N. K. Cicekli, "SVAS: Surveillance video analysis system," *Expert Syst. Appl.*, vol. 89, pp. 343–361, Dec. 2017.

[36] G. Acampora, P. Foggia, A. Saggese, and M. Vento, "A hierarchical neuro-fuzzy architecture for human behavior analysis," *Inf. Sci.*, vol. 310, pp. 130–148, Jul. 2015.

[37] L. Caruccio, G. Polese, G. Tortora, and D. Iannone, "EDCAR: A knowledge representation framework to enhance automatic video surveillance," *Expert Syst. Appl.*, vol. 131, pp. 190–207, Oct. 2019.

[38] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.

[39] EC Funded CAVIAR Project/IST 2001 37540. *CAVIAR*. Accessed: May 12, 2011. [Online]. Available: http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/

ZHIJIAO XIAO was born in Hunan, China, in 1980. She received the Ph.D. degree in computer software and theory from Sun Yat-sen University, in 2007. Since 2007, she has been a Lecturer with the College of Computer Science and Software Engineering, Shenzhen University. Her main research interests include event detection, process mining, cloud computing, and intelligent optimization.

JIANMIN JIANG received the Ph.D. degree from the University of Nottingham, U.K., in 1994. From 1997 to 2001, he was a Full Professor of computing with the University of Glamorgan, U.K. In 2002, he joined the University of Bradford, U.K., as a Chair Professor of digital media and the Director of the Digital Media and Systems Research Institute. He was a Full Professor with the University of Surrey, U.K., from 2010 to 2015, and a Distinguished Chair Professor (1000-plan) with Tianjin University, China, from 2010 to 2013. He is currently a Distinguished Chair Professor and the Director of the Research Institute for Future Media Computing, School of Computer Science and Software Engineering, Shenzhen University, China. He has published around 400 refereed research articles in international leading journals and conferences. His research interests include image/video processing in compressed domain, digital video coding, stereo image coding, medical imaging, computer graphics, machine learning, and AI applications in digital media processing, retrieval, and analysis. He was a Chartered Engineer, a Fellow of IET and RSA, a member of EPSRC College, U.K., and an EU FP-6/7 Evaluator.

ZHONG MING is currently a Professor with the College of Computer Science and Software Engineering, Shenzhen University. He led two projects of the National Natural Science Foundation, including one key project and one normal project. His major research interests include AI and cloud computing. He is a member of a council and a Senior Member of the China Computer Federation.