IJCV manuscript No.
(will be inserted by the editor)

Stochastic Representation and Recognition of High-level Group Activities

M. S. Ryoo[1,2], **J. K. Aggarwal**[2]

[1] Robot Research Department, Electronics and Telecommunications Research Institute, Korea, e-mail: `mryoo@etri.re.kr`
[2] Computer and Vision Research Center, The University of Texas at Austin, U.S.A., e-mail: `aggarwaljk@mail.utexas.edu`

Received: date / Revised version: date

Abstract This paper describes a stochastic methodology for the recognition of various types of high-level group activities. Our system maintains a probabilistic representation of a group activity, describing how individual activities of its group members must be organized temporally, spatially, and logically. In order to recognize each of the represented group activities, our system searches for a set of group members that has the maximum posterior probability of satisfying its representation. A hierarchical recognition algorithm utilizing a Markov chain Monte Carlo (MCMC)-based probability distribution sampling has been designed, detecting group activities and finding the acting groups simultaneously. The system has been tested to recognize complex activities such as 'a group of thieves stealing an object from another group' and 'a group assaulting a person'. Videos downloaded from *YouTube* as well as videos that we have taken are tested. Experimental results show that our system recognizes a wide range of group activities more reliably and accurately, as compared to previous approaches.

1 Introduction

A significant amount of research has addressed the recognition of human activities recently. Researchers have been particularly successful in recognizing the activities of one individual or between two individuals, such as punching and hand-shaking. Notably, we, in our previous work [21], have presented a representation syntax to describe high-level human-human interactions based on their sub-events, and proposed a hierarchical algorithm to recognize represented interactions probabilistically. Not only simple interactions such as pushing, kicking, and hugging are recognized, but also recursive interactions like 'fighting' between two persons are recognized with our previous framework. In this paper, we take our next evolutionary step in human activity recognition: recognition of group activities.

Group activities are the activities that can be characterized by movements of members who belong to one or more conceptual groups. Recognition of groups and their activities makes the analysis of high-level events possible, which are semantically meaningful when overall actions of multiple persons are considered jointly but not when they are considered individually. Automated recognition of suspicious groups and their activities such as 'a group of thieves robbing a bank' is essential for the construction of high-level surveillance systems. The analysis of movements and plays in team sports also becomes possible with the group activity recognition system. The semantic understanding of military operations and joint works is another application of it. Figure 1 shows example group activities.

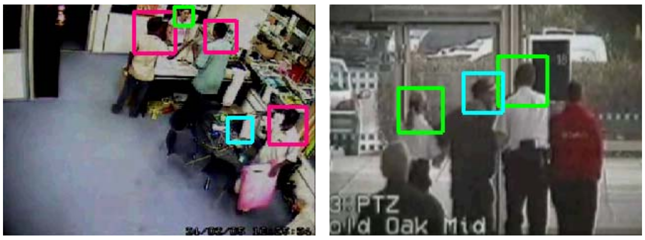

Fig. 1 Snapshots of group activities. The left figure shows a group-group interaction, 'group stealing'. The right figure shows a group-group interaction, 'group arresting'.

The recognition of complex group activities is a challenging task, particularly due to noisy observations and structural uncertainties of group activities. A group activity is performed by a varying number of participants, and the sub-events composing it (i.e. actions and interactions) are dependent on the situation. A sub-event of a group activity may occur for certain executions and may not for others, suggesting its stochastic nature. Further, the relationship between two sub-events may have multiple possibilities, implying that they need to be represented and recognized probabilistically.

In this paper, we present a novel methodology for the probabilistic recognition of high-level group activities. Our approach is to encode human knowledge on the structure of group activities while considering their stochastic nature, and to make the system recognize group activities based on their representation hierarchically. That is, we are crossing the horizon of previous description-based human activity recognition approaches [29,8] toward the recognition of group activities. We believe that ours is the first paper presenting stochastic recognition methodology for group activities with complex temporal, spatial, and logical structures. We focus on both a new format for the group activity representation and a new recognition algorithm.

Our system describes group activities in terms of a formal representation using a context-free grammar (CFG) as its syntax. A group activity is decomposed into several single person actions and person-person interactions between members of groups (i.e. sub-events), and our programming language-like representation describes the group activity by attaching universal quantifiers ($\forall$) and/or existential quantifiers ($\exists$) to those sub-events. For example, the group activity 'all members are carrying their baggage' must be represented by applying the universal quantifier to the individual activity 'a person carries baggage', while 'one member of the group raises his/her hand' must be represented by applying the existential quantifier to 'a person raises his/her hand'. Spatial constraints such as 'all members of one group should be spatially close' must also be listed as well. Furthermore, we have extended our representation to include the concept of subgroups.

Importantly, our representation is designed to describe stochastic group activities. As presented above, our representation describes the structure of each group activity in terms of sub-events which can either be simpler group activities or activities of individual members. In order to capture the probabilistic characteristics of such structures, our representation specifies the probability of each sub-event appearing given the group activity as well as the probability of each relationship between time intervals of the sub-events being satisfied.

Our system recognizes group activities by stochastically searching for individuals whose activities satisfy the representation of the group activity with the highest probability. That is, our system does not rely on the spatial correct segmentation of groups like most previous systems. In our approach, individual activities of persons in the scene are first recognized, and then used for the group activity recognition by comparing them with the representation. A hierarchical algorithm is designed to prune group member candidates that violate temporal constraints of the group activity. We model the probability distribution of the group activity using the Markov chain Monte Carlo (MCMC), and search for group members with the maximum posterior probability. For example, recognition of the group activity 'all members are carrying their baggage' is done by detecting individuals who performed the activity 'a person carries baggage' concurrently with a high probability. As a result of the algorithm, group activities and groups performing the group activities are recognized simultaneously.

We review previous works on group activity recognition while comparing them with our work in Section 2. In Section 3, our group activity representation syntax is provided. Section 4 presents our stochastic recognition methodology utilizing MCMC. The recognition problem is defined as a hierarchical Bayesian inference in Subsection 4.2, and the methodology to solve it is presented in Subsections 4.3 and 4.4. We discuss the experimental results in Section 5, and Section 6 concludes the paper.

2 Related works

The methodology we introduce throughout the paper is designed to recognize various types of group activities including group actions, group-persons interactions, group-group interactions, and intra-group interactions. The motivation is to construct a universal framework for the representation and recognition of complex group activities. Even though recognition of group activities has been paid less amount of attention, there has been a large amount of previous works on human activity recognition since early 90s [1,26]. In this section, we review various previous works on human activity recognition, while comparing their abilities (e.g. whether they are able to recognize group activities, whether they are able to make probabilistic decisions, ...) with our proposed system.

2.1 Approaches with sequential models

Activity recognition approaches using sequential models have been widely studied by many researchers. These sequential models represent an activity as a particular sequence of observations (i.e. features or sub-events), and recognize it from videos by probabilistically matching them with the model. Various methods including hidden Markov models [15,30], dynamic Bayesian networks [16,6], and stochastic context-free grammars [10] have been developed for the sequential recognition. Because of their characteristics, they were limited on handling human activities with complex temporal structures, and have focused on recognition of relatively simple and sequentially organized activities.

Oliver *et al.* [15] have adopted coupled HMMs to recognize interactions between two persons. Their HMMs model an interaction as a sequence of hidden states, each describing status of actors in the scene. Similarly, Park and Aggarwal [16] recognized interactions between two persons using DBNs. Ivanov and Bobick [10] designed a methodology to recognize multi-agent activities using SCFGs. They have represented human activities in a parking lot in terms of production rules of a CFG. These production rules are designed/learned to generate a sequence of terminals probabilistically, where each terminal correspond to an atomic-level action. The system was able to recognize hierarchical activities using stochastic parsing techniques, but only sequential relations (i.e. A occurred $before$ B) were allowed in the representation.

While most of the approaches with sequential models have focused on individual-level human activities, there have been attempts to recognize group activities using these models. Cupillard *et al.* [4] recognized intra-group interactions with a fixed number of participants using state models. Their focus was on the recognition of one specific group activity, 'a group is fighting'. Gong and Xiang [6] successfully recognized interactions between multiple objects using dynamic probabilistic networks, similar to DBNs. Their system also focused on only one class of group activity, intra-group interactions, and the number and types of participants were also fixed, as in [4]. Vaswani *et al.* [27] have analyzed the sequential changes in polygonal shapes formed by locations of multiple objects. They were able to measure overall abnormality in the movements of multiple objects. Similarly, Khan and Shah [11] recognized group activities by analyzing group members' rigidity formation sequentially. Their system recognized spatially structured group activities, a group parading for example, which is a type of 'all members of a group showing an identical action'. Zhang *et al.* [30] recognized intra-group interactions among limited number of participants in a meeting room using multi-layered HMMs.

However, each of these systems was designed to focus on the recognition of a single type of group activities. They either focused on group activities where each group member has its own role different from others [4,6,30], or those represented by an overall spatial formulation of group members [27,11]. The group activities of the first type are very similar to multi-agent activities recognized by [15,10,16], except for the fact that they are conceptually grouped by thev system. Further, most previous works assume that group members are spatially separable from non-members, to recognize group activities.

2.2 Description-based approaches

Description-based approaches are the approaches recognizing human activities by maintaining their knowledge on temporal, spatial, and logical structure of the activities. These approaches are particularly suitable for representing and recognizing high-level human activities having complex organizations and structures (e.g. A occurred $during$ B), such as 'stealing' and 'fighting' required for surveillance applications. Various models have been proposed to describe human activities, including representation languages [5,21], logical forms [22], and network forms [17,25]. Overall idea of a description-based approach is illustrated in Figure 2. Notice that many description-based approaches have used CFGs as a syntax to represent activities formally [5,21], but their usage is very different from the sequential approaches using SCFGs mentioned in the previous subsection. Our methodology presented in this paper falls into this category, and we discuss previous description-based approaches while comparing them with ours in this subsection.

Previous description-based approach focused on the recognition of actions performed by a single person, or interactions by a limited number of persons. Allen [2] introduced his temporal predicates, enabling one to describe temporal organization of events and activities in terms of first order logic. Allen's temporal predicates have been adopted by many researchers for the representation of activities' temporal structures. Pinhanez and Bobick [17] converted temporal networks of Allen [2] into a past-now-future (PNF) network, recognizing human actions in a kitchen environment. The system was able to compensate for a single failure. Intille and Bobick [9] recognized activities of multiple agents by constructing a representation similar to a programming language. Even though temporal structures of activities have been described only using two types of predicates ($before$ and $around$), they have shown successful results for American football play analysis. Siskind [22] represented human actions in a form similar to a first order logic with Allen's temporal predicates. Their event logic focused on a particular class of activities called 'liquid' events, and it was able to represent activities having multiple levels of hierarchy by limiting a sub-event to be used only once.

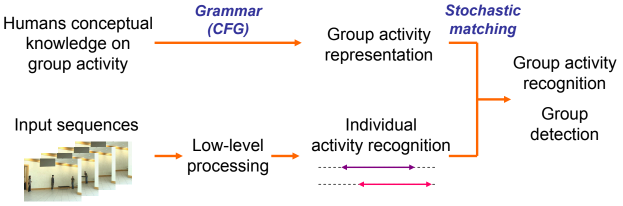
Humans conceptual knowledge on group activity
Grammar (CFG)
Group activity representation
Stochastic matching
Group activity recognition
Group detection
Input sequences
Low-level processing
Individual activity recognition

Fig. 2 Overall process of our description-based group activity recognition. Recognition is performed by semantically matching activities' representations with given observations. For example, the system maintains the representation of 'group stealing', specifying that 'a thief must take object while the other group members are distracting its owners'. Our system stochastically searches for observations satisfying such representation. Notice that CFGs are used as a 'representation syntax' to formally encode human knowledge of activities, in contrast to sequential approaches using SCFGs to directly 'recognize' activities by parsing.

Francois *et al.* [5] have developed their representation language called 'VERL' to describe human activities. Their language categorizes activities into primitive events, single thread events, and multi-thread events, enabling the representation of human activities having three levels of hierarchy. Allen's temporal predicates, spatial predicates, and logical predicates have been used to represent the conditions necessary for the activities. Hakeem *et al.* [7] also have introduced an activity representation language, 'CASEE', which is similar to VERL. They have represented an activity as a conjunction of necessary temporal and causal relations, and have recognized various activities involving persons and vehicles. Vu *et al.*'s [29] hierarchical approach also recognized activities represented as conjunctions of sub-events. These representation languages can be viewed as constrained versions of full first order logic particularly tuned toward computer vision based recognitions.

Furthermore, there have been attempts to incorporate probabilistic uncertainties into description-based logical models. Even though the above mentioned description-based approaches are able to recognize activities with complex structures, they have difficulties when their observations are noisy and/or subevents have a stochastic (i.e. uncertain) nature. The following methods have been designed to overcome the limitations of deterministic description-based approaches on handling noisy inputs and failures of low-level components probabilistically. Ryoo and Aggarwal [21] presented a description-based recognition approach that probabilistically compensates for the noisy observations and low-level components. They have designed their representation language using a CFG as its syntax, which enables the explicit description of the time interval of the activity being represented in contrast to the above mentioned works. Although their recognition system was limited to process interactions between two persons only, it was able to represent and recognize activities having complex structures with any levels of hierarchy (even recursive activities).

In the similar context, there have been attempts to adopt general logical models for probabilistic inferences to recognize human activities. Artificial intelligence researchers have developed a probabilistic inference framework for logically represented (i.e. description-based) activity models, including Bayesian Logic (BLOG) [14], Relational Markov Networks (RMNs) [24], and Markov Logic Networks (MLNs) [18]. Tran and Davis [25] have successfully applied MLNs for the computer vision-based activity recognition, probabilistically inferring events in a parking lot. In addition, RMNs have been successfully used for modeling temporal patterns for location sensor-based (e.g. GPS) activity recognition [13].

However, even though the above-mentioned approaches have attempted to integrate logical inference-based methodologies into a probabilistic framework, they were limited on recognizing groups and their activities. The number of participants involved in a group activity is unknown and their relationships change dynamically, preventing the previous approaches from directly being applied. The concept of groups must be introduced and represented to recognize complex group activities. Further, previous inference engines have difficulty recognizing highly hierarchical activities with multiple actors, since most of them (e.g. MLNs) make an inference using binary predicates without inferring the occurring time of the activity being recognized.

The contribution of our paper is on the stochastic representation and recognition methodology for group activities, which is designed to represent and recognize as broad range of high-level group activities as possible. Even though the recognition of complex group activities is important for many applications including surveillance, sports play analysis, and military systems, it has been largely unexplored by previous researchers. Our methodology captures uncertainties and variations in the structure of complex group activities, reliably recognizing them. We in the previous version of our paper [20] proposed a deterministic recognition system for group activities, but it was not able to compensate for the failures of low-level detections. In this paper, we overcome the limitations of our previous approach by proposing a methodology to represent and recognize structurally stochastic group activities. The performance of our new system is compared with the previous one in the experimental section. Table 1 compares the abilities of our approach with those of other activity recognition approaches, illustrating the advantages of our approach.

Table 1 Comparisons among abilities of the recognition systems for multi-agent activities and/or group activities. The column 'complex temporal relations' specifies whether the system is able to represent activities having complex temporal structures (e.g. Allen's temporal predicates [2]). 'Stochastic' indicates whether the activity recognition is performed stochastically or not. [9,5] are designed probabilistically, but is limited on compensation for the failures of their low-level components because of conditional independence assumption. 'Varying number of group members' describe the systems' ability to handle groups with various sizes. [27,20] are able to handle a certain portion of size changes in groups, but have limitations when the group size is large.

System\Ability	Types of activities recognized			Levels of Hierarchy	Complex temporal relations	Stochastic	Varying # of group members
	multi-agent activities	group actions	group interactions				
Intille *et al.* '99	√			unlimited		Δ	
Ivanov *et al.* '00	√			unlimited		√	
Cupilllard *et al.* '02	√			4-levels			
Vaswani *et al.* '03		√		1-level		√	Δ
Gong *et al.* '03	√			1-level		√	
Hakeem *et al.* '04	√			unlimited	√		
Khan *et al.* '05		√		1-level			√
Francois *et al.* '05	√			3-levels	√	Δ	
Zhang *et al.* '06	√			2-levels		√	
Ryoo *et al.* '08	√	√	√	Unlimited	√		Δ
Tran *et al.* '08	√			2-levels	√	√	
Ryoo *et al.* '09	√			Unlimited	√	√	
Our method	√	√	√	**unlimited**	√	√	√

3 Representation

The approach we take to represent a high-level group activity is to decompose it into several simpler activities, which we call *sub-events* of the activity. Sub-events of a group activity can be actions of a group member, interactions between the members, and/or other group activities of the same group. We first discuss different types of group activities that our system represents, and then present our formal representation syntax.

3.1 Types of group activities

We categorize group activities by considering the number of participating groups, the number of participants not in any group, and types of the activities' sub-events.

Group actions. If a group activity can be specified only using actions of its group members, we call it a group action. 'Marching' is a typical example of group actions: the activity can be characterized as all group members showing one type of individual action, 'moving'. The 'marching' can be denoted as *March(Group G1)*.

Group-persons interactions. If a group as well as persons outside of the group participates in the activity, we denote it as a group-persons interaction. The activity 'march by signal', which indicates an activity where a group starts marching after getting an order from a commander outside the group, is an example. 'March by signal' is denoted in the form of *MarchBySignal(Group G1, Person p1)*.

Group-group interactions. Two groups fighting and two groups having a conversation are good examples of group-group interactions. A group-group interaction can be composed of the actions of a group member of any group and/or interactions between two members from each group. A group-group fighting can be notated as *GroupGroupFighting(Group G1, Group G2)*.

Intra-group interactions. Intra-group interactions are group activities that involve interactions between members of the same group as sub-events. A group activity indicating that two members of a group are fighting is an example of intra-group interactions: *InterGroupFighting(Group G1)*.

Combinations. Our system is designed to represent group activities of the above-mentioned four elementary types as well as more complicated activities that can be decomposed into the elementary types (i.e. interactions between multiple groups and persons).

3.2 Group activity representation

We present a general representation syntax that is able to describe group activities of any of above-mentioned categories hierarchically. The concept of the *member variables* and the *group spatial predicates*, which have not been covered by any of previous activity representation methodologies, are newly introduced to denote participating group members and to describe spatial constraints needed among the group members. Based on new concepts and predicates, we represent a group activity in terms of three components: time intervals of activities of individual members (or other simpler group activities) composing it, the relationship specifying the temporal structure among sub-events, and necessary spatial conditions among group members. A detailed context-free grammar (CFG) syntax of our representation is presented in this subsection.

Member variables. A member variable is a variable used to denote one arbitrary member or all members of a group. We attach an existential quantifier ($\exists$) or a universal quantifier ($\forall$) to a member variable, in order to describe conditions that have to be satisfied by one member or all members of a group. If an existential quantifier is attached to a member variable, there has to be at least one individual member of the group who can be associated with the member variable to make related conditions true. If a universal quantifier is attached, all members of the group must be able to be associated with the member variable. That is, by using member variables as participants of sub-events, we are able to describe sub-events needed to be performed by all group members or by any one member. In addition, sub-events needed to be performed by the same individual may also be specified by using the same member variable as their participant. Our syntax to represent a list of member variables and its example are presented below.

$$
\begin{array}{l}
MemberVariableDefs\\
\quad \rightarrow \quad MemberVariableDef\text{“,”}MemberVariablsDefs\\
\quad | \quad\; MemberVariableDef\\
MemberVariableDef\\
\quad \rightarrow \quad Quantifier\, person\_var \text{ “}in\text{” } group\_var\\
Quantifier \qquad\quad \rightarrow \text{“}\forall\text{”}\,|\,\text{“}\exists\text{”}\\
Ex > \forall\; a\; in\; G1,\; \exists\; b\; in\; G2,\; \exists\; c\; in\; G3,\; ...
\end{array}
$$

Time intervals. A time interval specifies a starting time and an ending time of an occurring sub-event. A group activity is composed of multiple sub-events whose participants are specified using member variables and/or other non-member participants. In order to describe temporal structure of a group activity, both the sub-events composing the group activity and their time intervals must be listed. The formal syntax is as follows:

$$
\begin{array}{l}
TimeIntervalDefs\\
\quad \rightarrow \quad \text{“}def\text{” “(”}time\_var\text{“,”}ActivityName\text{“)”}\\
\quad | \quad\;\; \text{“}list\text{” “(” “}def\text{” “(”}time\_var\text{“,”}ActivityName\text{“)” “,”}\\
\qquad\qquad\qquad\qquad TimeIntervalDefs\text{“)”}\\
Ex > list(\; def(t1,\; Carrying(a)),\; def(t2,\; ...)\;)
\end{array}
$$

Predicates. Predicates are binary functions that are used to describe temporal, spatial, and logical relationships needed for the activity. Our system adopts Allen's temporal predicates (*before, meets, overlaps, during, starts, finishes,* and *equals*) [3], which have been widely used to specify temporal structures. Spatial predicates between individual persons, *near* and *touch*, are also used. Spatial predicates for describing a spatial status of a group are newly designed and added for the representation, whose definition is listed below. The predicate *dense* and *sparse* describe whether all group members are close to each other or not. Logical predicates (*and, or,* and *not*) are defined in a conventional manner to concatenate multiple predicates.

$$
\begin{array}{l}
dense(Group\; G,\; threshold) \iff\\
\quad Relative\; distance\; between\; any\; (g1,\; g2) \in G < threshold\\
sparse(Group\; G,\; threshold) \iff\\
\quad Relative\; distance\; between\; any\; (g1,\; g2) \in G > threshold
\end{array}
$$

Therefore, CFG syntax to represent necessary relationships of a group activity is defined using predicates. Note that the special time interval 'this' is used to specify the temporal relationship between the defining group activity itself and its other sub-events.

$$
\begin{array}{l}
Relationship\\
\quad \rightarrow LogicalPredicate\text{“(”}Relationship\text{“,”}Relationship\text{“)”}\\
\quad |\;\; TemporalRelationship\\
\quad |\;\; SpatialRelationship\\
\quad |\;\; \text{“}null\text{”}\\
TemporalRelationship\\
\quad \rightarrow TemporalPredicate\text{“(” “}this\text{” “,”}time\_var\text{“)”}\\
\quad |\;\; TemporalPredicate\text{“(”}time\_var\text{“,” “}this\text{” “)”}\\
\quad |\;\; TemporalPredicate\text{“(”}time\_var\text{“,”}time\_var\text{“)”}\\
SpatialRelationship\\
\quad \rightarrow IndividualSpatialPredicate\\
\qquad\qquad \text{“(”}person\_var\text{“,”}person\_var\text{“,”}threshold\text{“)”}\\
\quad \rightarrow GroupSpatialPredicate\text{“(”}group\_var\text{“,”}threshold\text{“)”}
\end{array}
$$

$IndividualSpatialPredicate \quad \rightarrow \text{“}near\text{”} \mid \text{“}touch\text{”}$
$GroupSpatialPredicate \quad \rightarrow \text{“}dense\text{”} \mid \text{“}sparse\text{”}$

As a result, the full representation is composed of three main parts: a list of member variables *MemberVariableDefs*, a list of time intervals of sub-events *TimeIntervalDefs*, and a list of relationships *Relationship*. Participants, member variables, and time intervals defined through participants, *MemberVariableDefs*, and *TimeIntervalDefs* are used in the term *Relationship* to describe necessary relationships. Three terms are integrated in our final CFG syntax where *GroupActivityDefine* is the starting variable. Example representations of the group activity ‘a group of people are carrying a large object by command of another person’ and ‘group fighting’ are presented as well.

$GroupActivityDefine$
$\quad \rightarrow name\text{“}(\text{”}participants\text{“})\text{”}\ \text{“} = \text{”}$
$\quad\quad \text{“}\{\text{”}MemberVariableDefs\text{“},\text{”}TimeIntervalDefs\text{“},\text{”}$
$\quad\quad Relationship\text{“}\};\text{”}$
$Ex > \textbf{CarryByCommand}(Group\,G1,\ Person\,p1)\ =\ \{$
$\quad\quad \forall a\ in\ G1,$
$\quad\quad list(\ def(t1,\ \textbf{Carry}(a)),\ def(t2,\ \textbf{Command}(p1))),$
$\quad\quad and(\ equals(t1,\ this),\ meets(t2,\ t1))$
$\quad \};$
$\quad \textbf{GroupGroupFighting}(Group\,G1,\ Group\,G2)\ =\ \{$
$\quad\quad \forall a\ in\ G1,\ \exists b\ in\ G2,$
$\quad\quad list(\ def(t1, \textbf{Approach}(G1,\ G2)),$
$\quad\quad\quad def(t2, \textbf{Fight}(a,\ b))),$
$\quad\quad and(\ and(dense(G1),\ dense(G2)),$
$\quad\quad\quad and(equals(t1,\ this),\ meets(t1,\ t2)))$
$\quad \};$

A group activity can always be decomposed into four elementary types if and only if member variables can be divided into independent pairs and/or singles. That is, we limit a member variable to have interaction with only one other variable to make the recognition process tractable.

Subgroups. We further extend our representation to include the concept of subgroups. We say that the group B is a subgroup of group A, if and only if all group members of B are members of A as well. The subgroups are particularly useful when describing a group activity of a portion of group members showing a specific type of interaction. Our representation syntax allows the description of subgroups. When defining a subgroup, we make its group variable $group\_var$ to have the form of the defining subgroup name $subgroup\_name$, attached at the end of the existing group name $group\_name$ followed by a dot (i.e. “.”).

$group\_var \rightarrow group\_name | group\_name\text{“}.\text{”}\,subgroup\_name$
$Ex > \forall a \in G1,\ \exists b \in G1.G2$

3.3 Stochastic representation

Human activities, especially group activities, are often composed of sub-events having a stochastic nature. Certain sub-events may occur during one execution of the group activity, while not in another. In addition, the relationship between time intervals of these sub-events may change depending on the environment or participants. For example, in the case of ‘group assault’ where a group of persons are attacking a particular target person, there may (or may not) exist some group members who are just watching or guarding the area. These sub-events of ‘watching’ and ‘guarding’ are stochastic sub-events, where each of them has a certain probability of occurring.

Our group activity representation is designed to capture such structural variations caused by stochastic sub-events. In the case of a stochastic sub-event, the time interval associated with it is defined together with its occurrence probability, describing how likely the sub-event appears when the group activity containing it occurs. As a consequence, the following production rule has been added to our CFG syntax:

$TimeIntervalDefs$
$\quad \rightarrow \quad \text{“}def\text{”}\ \text{“}(\text{”}time\_var\text{“},\text{”}ActivityName\text{“},\text{”}probability\text{“})\text{”}$
$Ex > def(t1,\ Guard(a),\ 0.6)$

Furthermore, our representation allows the stochastic description of a temporal relationship between two intervals. If the relationship between two time intervals is flexible and has multiple possibilities, we represent it as a list of all predicates with non-zero probabilities. The probability associated with each temporal relationship must be specified. The constraint is that the sum of the probabilities of relationships between any two time intervals must be 1. The following production rule has been added to represent stochastic relationships.

$TemporalRelationship$
$\quad \rightarrow \quad \text{“}stochastic(\text{”}\ TemporalRelationship\ \text{“},\text{”}\ prob\text{“})\text{”}$

By following the production rules of our CFG syntax, we are able to represent group activities with a stochastic nature. Both the uncertainties in sub-events and their relations are described with our representation. The stochastic representation of the example of ‘group assault’ is provided below. The representation is composed of the sub-event ‘attack’, which is a ‘fight’ interaction followed by an ‘approach’, as well as stochastic sub-events of ‘watch’ and ‘guard’. The actors of these sub-events are described using member variables, where G indicates the group performing the activity and S indicate a sub-group of G. Stochastic temporal relationships are described using time intervals as well.

$Ex > \textbf{GroupAssault}(Group\,G,\ Person\,p1)\ =\ \{$
$\quad\quad \forall a\ in\ G.S,\ \exists b\ in\ G,\ \exists c\ in\ G$
$\quad\quad list(\ def(t1, \textbf{Attack}(a,\ p1)),$
$\quad\quad\quad list(\ def(t2, \textbf{Watch}(b,\ p1),\ 0.5),$
$\quad\quad\quad\quad def(t3, \textbf{Guard}(c, door),\ 0.6))),$
$\quad\quad and(\ and(\ stochastic(during(t2,\ t1),\ 0.9),$
$\quad\quad\quad\quad stochastic(overlaps(t1,\ t2),\ 0.1)),$
$\quad\quad\quad and(\ during(t3,\ t1),\ equals(t1,\ this)))$
$\quad \};$

4 Recognition

This section discusses an algorithm to recognize high-level group activities that have been represented stochastically us-

```
GROUP_ACTIVITY_RECOGNIZE(Activity $G$) {
    Detect a set of time intervals $W_i$ of individual
      activities per sub-event $S_i$; // Subsection 4.1.

    List $LC$ = CANDIDATE_DETECT($G$);

    for $j$ = 1 to sizeof($LC$) {
      Actors $C_{1,...,n}$ = $LC(j)$;
      Interval $t$ = time interval of special variable 'this';
      Group $M^*$ = GROUP_ESTIMATE($G$, $t$, $C_{1,...,n}$);

      if ($P(G^t|O)$ is high) return $M^*$; // Equation (6).
    }
}
```

Fig. 3 Pseudo codes describing our overall recognition algorithm. It takes advantage of functions presented in Figures 6 and 9, detecting groups with high probability of executing the activity.

ing our CFG syntax. Our recognition process conveys the hierarchical structure of our group activity representation, recognizing group activities based on the recognition results of their sub-events. We have defined the problem as a hierarchical Bayesian inference: we calculate the posterior probability of the group activity given video observations. The goal is to find an occurring time interval of the activity, which has a high enough probability of group members satisfying the activity structure (i.e. the representation).

We first discuss the base case of the recognition, activities of individuals, in Subsection 4.1. In Subsection 4.2, we present the Bayesian formulation of our group activity recognition problem. We describe how our system computes the posterior probability of a group activity, given its starting time and ending time (i.e. a time interval). Next, in Subsection 4.3, we present an algorithm to calculate candidate time intervals which are guaranteed to have non-zero posterior probabilities. A pool of candidate group members performing the group activity is computed together with their corresponding candidate interval. In Subsection 4.4, we evaluate each candidate time interval by searching for the set of group members providing the highest probability within the interval. A MCMC-based sampling methodology has been designed to search for the approximate optimum solution. Overall recognition algorithm is presented in Figure 3.

4.1 Activities of individuals

The base case of our hierarchical group activity recognition is the recognition of individual activities. High-level group activities are represented in terms of activities of individual persons and other simpler group activities (which themselves can be decomposed as well), suggesting that the recognition of human actions and human-human interactions must be performed first. We in our previous work have presented an activity recognition methodology which is able to probabilistically recognize human-human interactions such as a 'fighting' [21], and we take advantage of it in this paper.

In addition, we have adopted the tracking algorithm developed by Ryoo and Aggarwal [19]. The tracking algorithm is especially designed to handle several types of occlusions among persons and other objects (e.g. pillars). It utilizes the background subtraction as well as the head detection, as illustrated in Figure 4. Tracked person blobs serve as low-level features for human-human interaction recognition system. Once a person is correctly segmented, color histograms are used to classify the type of the person (e.g. policeman vs. pedestrians), if needed. Viola and Jones's detector [28] is used for objects (e.g. laptop computer). Dynamic time warping (DTW) models are constructed to estimate motion of each individual, where width/height ratio and the center position of a bounding box are used as features for the DTW. These results are passed to the human action and interaction recognition system.

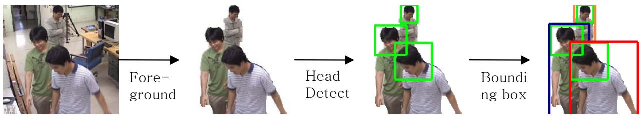

Fig. 4 Low-level processing of the system.

4.2 Hierarchical problem formulation

Here, we define the group activity recognition problem as a hierarchical Bayesian inference. The objective of the group activity recognition is to find the time interval t of the group activity G, given a video observation O of groups performing the activity. That is, we must calculate the posterior probability $P(G^t|O)$, and deduce that the group activity occurred only when it has a high value for the time interval t. What we present in this subsection is a methodology to compute such probability, assuming that t is provided (how to search for such t candidates will be discussed in the next subsection).

Even with a fixed occurring time t, there are multiple possible groupings of persons in the scene. Therefore, the actor group of the activity (i.e. the group members who performed the activity) providing the highest probability must be identified to recognize the group activity.

$$\begin{aligned} P(G^t|O) &= max_M P(G^t(M)|O) \\ &= max_M \frac{\pi_G^t(M)}{\pi_G^t(M) + \pi_{\neg G}^t(M)} \end{aligned} \tag{1}$$

where

$$\pi_G^t(M) = P(O|G^t(M))P(G^t(M)). \tag{2}$$

M is a set of group members $\{m_1, m_2, ..., m_{|M|}\}$. $G^t(M)$ indicates that group members M are performing the activity G at the time interval t. $P(O|G^t(M))$ describes the probability that the video observations are generated by the group activity G performed by members M at time interval t. $P(G^t(M))$ is the prior probability.

The recognition process must convey the hierarchical nature of the representation. Group activities are composed of several sub-events, as we have represented in Section 3. That is, the system must posses an ability to make a hierarchical inference based on the detection of its sub-events, $S_1, ..., S_n$. The system must be able to detect the sub-events first, and take advantage of the detection results for the calculation of the probability as follows:

$$\begin{aligned} P(O|G^t(M)) = \sum_{S_1^{t_1},...,S_n^{t_n}} & [P(O|M,Q,S_1^{t_1},...,S_n^{t_n}) \\ & P(S_1^{t_1},...,S_n^{t_n}|G^t(M))] \end{aligned} \tag{3}$$

where $S_1, ..., S_n$ are the sub-events of the G performed by the corresponding actors, and $S_i^{t_i}$ indicates whether the sub-event S_i occurred at the time interval t_i or not. We assumed that sub-event detection results $S_1^{t_1}, ..., S_n^{t_n}$ are strictly dependent on the group activity $G^t(M)$, and video observation O are dependent on the sub-event detection results. Since a certain sub-event may have a stochastic property (i.e. it may occur and may not), we need to consider both cases of having the sub-event and not having it, integrating their probabilities. Q is the set of quantifiers associated with the member variables of the group activity (i.e. $P(Q|G^t(M)) = 1$).

The similarity between the time intervals of the detected sub-events and the stochastic representation of the group activity are measured with $P(S_1^{t_1}, ..., S_n^{t_n}|G^t(M))$. The confidence of the detected sub-events corresponds to $P(O|M, Q, S_1^{t_1}, ..., S_n^{t_n})$.

The structural similarity, $P(S_1^{t_1}, ..., S_n^{t_n}|G^t(M))$, can further be enumerated in terms of the relationship predicates described in 3.2. If the detected sub-events of the group members have a similar structure to the representation probabilistically, the probability of them satisfying the specified relations must be high.

$$\begin{aligned} P(S_1^{t_1},...,S_n^{t_n}|G^t(M)) = & \prod_{rel} P(rel|S_a^{t_a}, S_b^{t_b}) \\ & \prod_i P(S_i^{t_i}|G^t(M)) \end{aligned} \tag{4}$$

where each $rel(S_a, S_b)$ is the relationship between ath sub-event and bth sub-event stated in our representation. These terms are calculated per each combination $(S_1, S_2, ..., S_n)$ based on the grouping M, the sub-events S_i, and their relations rel stated in the activity representation.

The key of the hierarchical recognition is the computation of the probability of the observations given sub-events: $P(O|M, Q, S_1^{t_1}, ..., S_n^{t_n})$. Particularly, a sub-event whose actor is described using a group member variable must be performed by multiple members of the group M. This type of sub-events needs to be performed by all members of the group or any one member of the group depending on the quantifiers associated with the member variables. Detection results of the sub-event performed by all possible actors must be evaluated, and the overall probability must be computed while considering the fact that the group members should execute the sub-events and non-group members should not.

Assuming the conditional independence among sub-events, we evaluate the similarity of the observation as follows:

$$\begin{aligned} P(O|M,Q,S_1^{t_1},...,S_n^{t_n}) &= \prod_i P(O_i|S_i^{t_i}(M)) \\ &= \prod_i d \cdot e^{-(|K_i - C_i|/|K_i| + |L_i \cap C_i|/|K_i|)} \end{aligned} \tag{5}$$

where O_i is the video regions related with the sub-event S_i. C_i is the set of all persons performing the sub-event while satisfying the activity structure specified in the representation, K_i is the set of essential group members who must perform the sub-event, and L_i is the set of non-members who should not perform the sub-event. Thus, $|K_i - C_i|$ indicates the number of essential members who are not performing the sub-event, and $|L_i \cap C_i|$ specifies the number of anti-essential individuals performing the sub-event. Similar to [23], we are calculating the error ratio, which can be viewed as the distance between the optimal structure and the structure formed by M. The optimal case is that $K_i - C_i$ and $L_i \cap C_i$ are empty sets. C_i is dependent on the sub-event recognition results. On the other hands, the K_i and L_i are dependent on the grouping M, and the system must choose M so that the overall probability of the group activity is the maximum. We discuss the process of deciding the sets K_i and L_i further in Subsections 4.3 and 4.4. In the case of a sub-event whose actor is not a group member, K_i always is a set with a single element (i.e. an acting person). d is a constant which can be ignored when computing $P(G^t|O)$.

In summary, the posterior probability of a group activity G occurring at the time interval t given video O is enumerated as follows.

$$P(G^t|O) = \frac{\pi_G^t(M^*)}{\pi_G^t(M^*) + \pi_{\neg G}^t(M^*)} \tag{6}$$

where M^* is the optimum group maximizing the term $\pi_G^t(M)$:

$$\begin{aligned} \pi_G^t(M) &= P(O|G^t(M))P(G^t(M)) \\ &= c \cdot \sum_{S_1^{t_1},...,S_n^{t_n}} [\prod_{rel} P(rel|S_a^{t_a}, S_b^{t_b}) \prod_i P(S_i^{t_i}|G^t(M)) \\ &\qquad \prod_i d \cdot e^{-(|K_i - C_i|/|K_i| + |L_i \cap C_i|/|K_i|)}] \end{aligned} \tag{7}$$

where c is a constant indicating the prior probability, $P(G^t(M))$. We assume a uniform prior probability.

This implies that the computation of the posterior probability associated with each t involves searching of the optimum group M^*. Given a pool of sub-event detection results from multiple actors, the system must search for group members $M = \{m_1, ..., m_{|M|}\}$ maximizing the above probability (7). However, searching for such a set of group members is a traditional constraint satisfaction problem, which is known to be NP-hard. A brute force searching will take an exponential amount of time to find the optimum solution.

In order to find the solution (i.e. M^*) while avoiding the exponential amount of computations, we in this paper are making the following approximations: Instead of fully enumerating the entire search space for M, we use a MCMC-based sampling for the modeling of the probability $\pi_G^t(M)$. The sample M with the maximum probability will be selected for each t, which we will discuss further in Subsection 4.4.

Multi-group activities. In the above-mentioned Bayesian equations, we have assumed that the acting group of the activity is only one. In the case of group-group interactions, there are two or more groups involved in the activity. In order to recognize a group-group interaction, we need to search for two groups $M1$ and $M2$ jointly, which provide the maximum posterior probability $P(G^t(M1, M2)|O)$. Sub-events of group-group interactions include interactions between members of two groups described using two member variables. In this case, the C_i, K_i, and L_i, which are needed to calculate $P(O_i| S_i^{t_i}(M1, M2))$, become sets of pairs of group members from $M1$ and $M2$. C_i indicates pairs of actors who are performing the sub-event S_i, K_i is a set of essential actor pairs, and L_i is a set of anti-essential actor pairs. Similar to the case of a single-group activity, the criteria of the K_i and L_i selection given M will be discussed in Subsection 4.4.

4.3 Candidate time interval detection

The goal of the group activity recognition is to find the interval t giving a high probability. A brute force method to find such an interval is to evaluate all possible T^2 intervals where T is the number of frames observed, which is computationally inefficient. Our approach is to propose a number of promising candidate intervals without spending too much computational time, and evaluate only those candidates in detail for the final recognition. In this subsection, we present a methodology to compute candidate time intervals for a group activity.

We focus on the fact that the sub-events' temporal constraints must be satisfied in order for the group activity to occur. That is, a time interval with a grouping that makes its members to violate the constraints of the activity has a zero posterior probability, and must be discarded. The algorithm presented throughout this subsection hierarchically finds the group activity's valid time interval candidates based on the temporal structure matching. In addition, individuals who performed the sub-events are computed for each valid t. These individuals form a pool of group member candidates, which will be passed to the algorithm in Subsection 4.4 to search for the approximate optimal grouping.

Focusing on the fact that not many persons in the scene satisfy the temporal constraints of the activity, we are pruning temporally inconsistent time intervals of the group activity. A pool of member candidates computed for each t is guaranteed to be a superset of the group with a non-zero probability of performing the activity. That is, the algorithm provides an upper bound per group that satisfies the temporal relationships of the activity representation at least with a small probability.

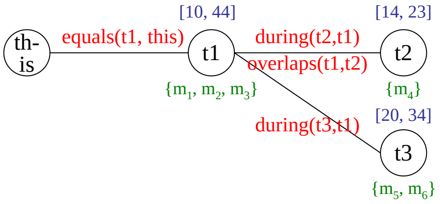

Fig. 5 An example relationship tree of the stochastic representation of the 'group assault'. Example time intervals assigned are specified on top on the nodes, and the pool of actors who performed the sub-event are listed below the nodes.

An important fact is that our representation of a group activity may contain stochastic sub-events, indicating only a potion of them will occur during the activity. This implies that each activity has multiple possible structures, having a different number of sub-events and different relations. The system needs to consider all of the possible structures: we apply our algorithm to obtain valid time intervals and their group candidate supersets to multiple possible structures of the activity. The algorithm is repeatedly applied to each subset of the stochastic sub-events together with the set of all non-stochastic sub-events. Relationships of sub-events not selected are ignored. Since there are only a limited number of stochastic sub-events, the number of possible structures is tractable: 2^z where z is the number of stochastic sub-events which is a small constant in general.

Clustered time intervals. Here, we define the concept of the *clustered time interval*, which describes an occurring time of a sub-event whose actors are represented in terms of group member variables. In the case of a sub-event performed by a group member, multiple time intervals are generated by various group members. In order to construct one representative time interval describing all of them, we cluster time intervals of the same actions and interactions ignoring the actors. The clustering is done based on the time intervals' starting and ending times. A clustered time interval is treated as a single time interval, so that the sub-event's relationship with others can be analyzed.

Clustered time intervals provide an efficient approximation for the calculation of the group members satisfying the temporal relationship of the activity. When calculating the structural similarity $P(S_1^{t_1}, ..., S_n^{t_n}|G^t(M))$, the ratio of each time interval in the clustered interval satisfying the proposed relationship by it is considered to measure the overall similarity.

Temporal constraint matching. The goal of the temporal constraint matching is to assign a time interval (which maybe a clustered time interval) to each sub-event, so that the temporal constraints described using the predicates are satisfied. A time interval per sub-event must be chosen, while satisfying the representation of the group activity. The result is a time interval candidate of the group activity (i.e. t), which is computed for each combination of the sub-event assignments.

The problem of assigning detected time intervals to satisfy temporal relationships is a traditional constraint satisfaction problem itself. An activity can occur multiple times, suggesting that each sub-event has multiple possible time interval assignments. Therefore, if r is the average number of time intervals of one sub-event and n is the number of sub-events, there are r^n possible combinations of time interval associations on average. The algorithm must search for combinations that satisfy the temporal relationship of the represented group activity, so that the time interval candidates for the group activity are obtained.

In order to detect such combinations efficiently without spending an exponential amount of computations, we model the relationship of a group activity as a set of trees: We first enumerate relationships to make them in DNF (disjunctive normal form). Each clause of DNF is a conjunction of temporal relations, and we construct an undirected graph for each clause where variables indicating time intervals are nodes and predicates between them are edges. Figure 5 shows an example relationship tree of the stochastic group activity 'group assault'. In the case when temporal relationships for a group activity contain a cycle, our system breaks the cycle (i.e. converts to a tree) to perform the recognition process, which is an approximation of the actual temporal constraints.

With a tree structure assumption of temporal relationships, searching for a valid combination can be done in polynomial time. Figure 6 shows a detailed pseudo code of our time interval detection algorithm. For each tree, our algorithm searches for valid combinations by assigning time intervals to nodes (i.e. time variables) from the root to the leaves. Only when a time interval of a child node satisfies the relationship with any one of the time intervals assigned to the parent node, the interval can be assigned to the child node. In the case of a stochastic relation with multiple possibilities, a time interval is assigned to the child node if any relationship among the possible relations is satisfied with a time interval of the parent node. The probability associated with the stochastic relationship is ignored at this point, and it will be evaluated in a later stage when we search for the group M^* with the maximum posterior probability.

The algorithm treats sub-events done by any persons as valid candidate time intervals as long as they satisfy the temporal constraints. However, in order for a group activity to occur, the sub-events associated with the same member variable must be done by the same person. Therefore, our system discards time interval combinations which violate the constraint that 'sub-events associated with the same member variable must be done by the same person'.

Once valid time intervals of sub-events are assigned, time intervals of a group activity itself, t, can be computed by calculating the range of the special time interval 'this'. This will become the result time interval of the group activity, if the system later decides that it has a high posterior probability. The computation of 'this' also suggests a hierarchical recognition. Figure 7 shows an example hierarchical recognition process enabled with our temporal constraint matching algorithm. The time complexity of the overall algorithm is $O(r^2 + b)$, where b is the total number of combinations satisfying temporal constraints.

```
CANDIDATE_DETECT(Activity $G$) {
    Tree $D$ = Temporal structure tree of $G$, composed of
        {$i$ = 1, ..., $n$} nodes;
    Node $r$ = root node of $D$, always corresponding to the
        special time interval `this';

    ASSIGN(r);
    List $LC$ = CANDIDATE_MEMBERS($V_{1,\ldots,n}$, $A_{1,\ldots,n}$);

    return $LC$;
}

ASSIGN(Node i) {
    Node $p$ = $i$'s parent;
    Intervals $V_i$ = {}; Actors $A_i$ = {};

    Intervals $W_i$ = List of all possible time intervals that
        can be assigned to node $i$;

    Actors $B_i$ = {};
    for (each $w_i$ in $W_i$) {
        $b_i$ = an actor of $w_i$; // $b_i$ can be a set of actors, if
                                      $w_i$ is a clustered interval.
        add $b_i$ to $B_i$;
    }

    if ($p$==null) {$V_i$ = $W_i$; $A_i$ = $B_i$;}
    else for $j$ = 1 to sizeof($V_i$)
            for $k$ = 1 to sizeof($V_p$)
              if($W_i(j)$ and $A_p(k)$ satisfies temporal relation)
                {add $W_i(j)$ to $V_i$; add $B_i(j)$ to $A_i$;}

    for (each node $c$ who is a child of $i$)
      ($V_c$, $A_c$) = ASSIGN($c$);
      for $j$ = 1 to sizeof($V_i$)
        if (no $V_c(k)$ satisfies temporal relation with $V_i(j)$)
          {remove $V_i(j)$ from $V_i$; remove $A_i(j)$ from $A_i$;}
    return ($V_i$, $A_i$);
}

CANDIDATE_MEMBERS($V_{1,\ldots,n}$, $A_{1,\ldots,n}$)
{
    List $LC$ = {};
    for (each combination ($a_1$, $a_2$, ..., $a_n$) $\in$ ($A_1$, $A_2$, ...,$A_n$))
      if (($a_1$, $a_2$, ..., $a_n$) satisfies temporal relations)
        add ($a_1$, $a_2$, ..., $a_n$) to $LC$;
    return $LC$;
}
```

Fig. 6 Pseudo code of the detecting group member candidates that satisfy temporal constraints.

Group member candidates. As a result of the algorithm, a set of valid time interval candidates is obtained. For each candidate t, we also compute clustered time intervals assigned to group-related sub-event. The actors of a sub-event's clustered time interval form a pool of candidate group members. The actors of the clustered time intervals, which have been verified to satisfy the temporal constraints using the algorithm proposed above, suggest the upper bound of the grouping. In the case when a sub-event is an interaction of members of two different groups, the pairs of actors of the clustered interval form a pool of candidate pairs to be assigned jointly to the two groups. These actors form a superset of the optimal

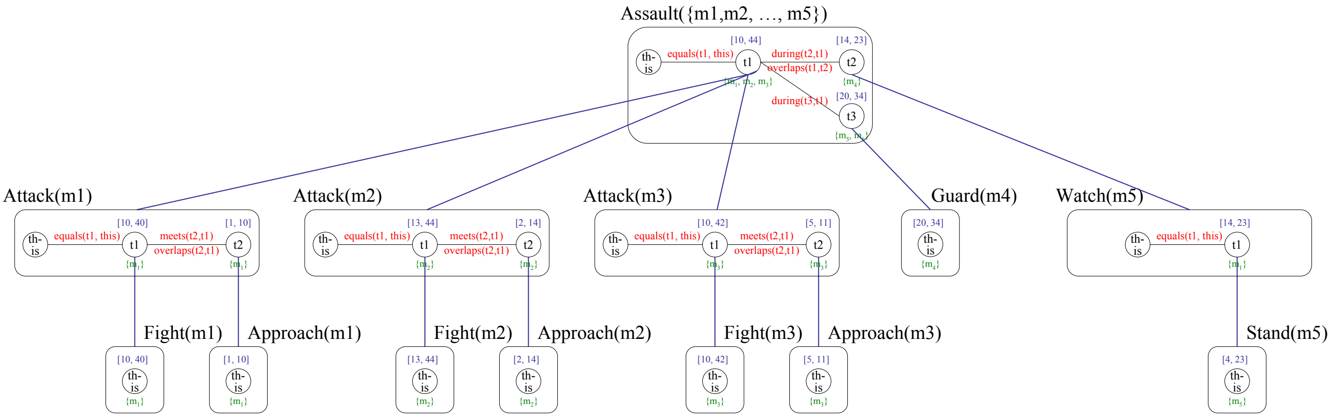

Fig. 7 An example hierarchical process tree of the activity 'group assault'. Each node with a box shape corresponds to the activity, and it contains its relationship tree. We are able to observe that the root node contains the tree described in Figure 5. Edges connecting box nodes specify the hierarchy.

grouping given the current time interval combination. Each clustered time interval generates one superset, and we denote the candidate superset of the sub-event S_i as C_i.

For example, in the case of the 'group assault', there are three clustered time intervals generated by sub-events 'attack', 'watch', and 'guard': C_{attack}, C_{watch}, and C_{guard}. Once the combination of the time intervals satisfying the temporal relationship of the 'group assault' is found (e.g. Figure 5), the pool of candidate actors are decided for each sub-event. In Figure 5, the persons $\{m_1, m_2, m_3\}$ performed attacking in the time interval $[10, 44]$, the person m_4 was guarding in $[14, 23]$, and the persons $\{m_5, m_6\}$ was watching in $[20, 34]$. The result pools are: $C_{attack} = \{m_1, m_2, m_3\}$, $C_{watch} = \{m_4\}$, and $C_{guard} = \{m_5, m_6\}$. Any subset of these sets satisfies the temporal relationships.

4.4 MCMC-based group estimation

In this subsection, we present a methodology to find the optimal group M^* which provides us the maximum posterior probability for each time interval candidate t. We have designed an algorithm using a Markov chain Monte Carlo (MCMC) technique to search for M^*. The Markov chain Monte Carlo (MCMC) is a methodology to obtain samples following a particular probability distribution. The MCMC sampling is particularly useful when the domain space is large, since the enumeration of an entire distribution is intractable in such cases. For example, in our case, there are 2^s possible assignments for each group M where s is the number of persons in the scene. Maintaining samples obtained from the distribution using MCMC provides a good approximation of the distribution model, and it has been applied to several computer vision problems including object detection and tracking [12, 23].

More specifically, we use the MCMC to solve the M^*:

$$M^* = argmax_M P(G^t(M)|O) = argmax_M \ \pi^t_G(M). \tag{8}$$

Our search space is a multi-dimensional space, having s dimensions. Each dimension has a discrete value, either 0 or 1, where 0 indicates that the actor corresponding to the dimension did not participate in the group activity and 1 indicates that the actor participated. In order to find the M^*, we sample the distribution $\pi^t_G(M)$ of Equation (7) assuming the uniform prior probability.

The sets K_i and L_i in Equation (7) are dependent on the grouping M, while C_is are provided from the previous subsection. Since each sub-event is executed stochastically, $|K_i - C_i|$ and $|L_i \cap C_i|$ need to be computed considering the probability associated with each member (or member-pair) performing the sub-event. We calculate the expectation of the number of set members, based on the sub-event detection results $P(S^{t_i}_i(k)|O_i)$.

$$\begin{aligned} |K_i - C_i| &= \sum_{k \in (K_i - C_i)} E[S^{t_i}_i(k)|O_i] \\ |L_i \cap C_i| &= \sum_{l \in (L_i \cap C_i)} E[S^{t_i}_i(l)|O_i] \end{aligned} \tag{9}$$

where k and l are the individuals performing the sub-event.

The terms $\prod_{rel} P(rel|S^{t_a}_a, S^{t_b}_b)$ and $\prod_i P(S^{t_i}_i|G^t(M))$ in Equation (7) are also calculated depending on M. The clustered time intervals of the sub-events are dependent on the members of M. The spatial relationships need to be measured based on the spatial distances between each members of M. Thus, the overall probability of $\pi^t_G(M)$ is dependent on the grouping M, and we must sample them to find the optimum solution.

A Metropolis-Hastings algorithm with reversible jumps is applied to obtain samples following $\pi^t_G(M)$. The probability of the samples will be compared later to find the sample with the maximum value. The following shows the dynamics of the sampling using the algorithm.

$$a = \frac{\pi^t_G(M') \cdot q(M', M)}{\pi^t_G(M) \cdot q(M, M')} \tag{10}$$

The transition probability is as follows: $P(M, M') = min(1, a)$. After each transition, the new M is selected as a sample of the $\pi^t_G(M)$. The $q(M, M')$ is the proposal probability, which we model to have a uniform probability of selecting one of the discrete moves described below. Broadly, there are two types of moves: the move of adding a person to the group M, and the move of removing a group member from the group.

- Add a group member m from a pool of sub-event actors C_i calculated in Subsection 4.3, to M. There are multiple pools, if there exist more than one clustered time intervals. Select $m \in M^c$ randomly among all of those pools of C_i. In the case when the sub-event is an interaction between two group members, select a pair $(m1, m2)$ from C_is and add it to the groups $M1$ and $M2$ respectively.
- Remove a group member m from the current M. In the case of an interaction, select a pair $(m1, m2)$ and remove them from the groups $M1$ and $M2$ respectively.
- If a group member is added into a group which is a subgroup of another group, then the member is added to this group as well. Similarly, if a member is being removed from a super-group, it must be removed from all its subgroups as well.

In order to compute $\pi^t_G(M)$, the sets K_i and L_i must be decided. The set K_i is chosen for each sub-event S_i, and it describes the necessary actors of the sub-event. That is, K_i describes a set of group members who are required to perform the sub-event, in order to satisfy the structure of the group activity. Similarly, the set L_i contains the persons who should not perform the sub-event. All elements of the set L_i are nonmembers of the group. The intuition behind the consideration of L_i is that if a person is not in the group, he/she is not likely to perform the sub-event satisfying the spatio-temporal structure of the group activity.

For example, in the case of 'group marching', there is only one member variable, and the sub-event of 'move' needs to be performed by all members corresponding to the member variable. If the current M is $\{m_1, m_2, m_3\}$ and C_{move} is $\{m_1, m_2, m_4\}$ for 'group marching', the K_{move} and L_{move} is as follows: $K_{move} = \{m_1, m_2, m_3\}$, and $L_{move} = \{m_4\}$. This indicates that all $\{m_1, m_2, m_3\}$ must be 'moving' while m4 should not, if M is the optimal group (which is not the case).

The selection of K_i and L_i depends on the number of member variables of the sub-event (i.e. action vs. interaction) and the types of quantifiers (i.e. $\exists$ and $\forall$) associated with them. In the case when a sub-event needs a pair of actors (i.e. an interaction), the K_i and L_i are composed of the essential and anti-essential actor pairs. Figure 8 shows each case of interaction sub-events. Edges between members shows an example of necessary pairs (i.e. K_i) for the group activity structure to be satisfied. We categorize the cases into 6 types (2 for actions and 4 for interactions), and discuss methodologies to compute K_is and L_is given M.

Group action case 1: $\exists$**.** This is the case where only one member of a group needs to perform the sub-event. Any one member with the maximum probability of executing the sub-event is chosen.

$$K_i = \{k \mid k = argmax_{k \in M} P(S_i^{t_i}(k)|O_i)\}$$
$$L_i = \{\}$$

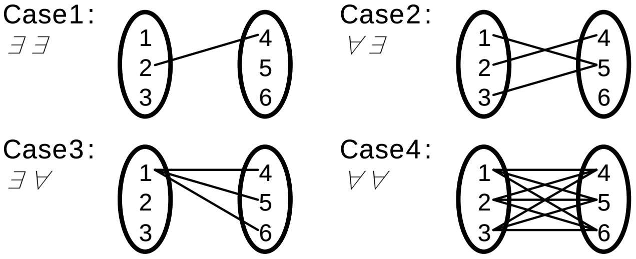

Fig. 8 Example K_is in 4 cases of group-group interactions.

Group action case 2: $\forall$**.** All members of the group must perform the sub-event. Therefore,

$$K_i = \{k \mid k \in M\}$$
$$L_i = \{l \mid l \in M^c\}$$

Multi-group activity case 1: $\exists$ $\exists$**.** In multi-group activity cases, the goal is to search the optimal group pair $(M1^*, M2^*)$ jointly. Therefore, K_i and L_i contain pairs of actors computed based on $M1$ and $M2$.

$$\begin{aligned} K_i = \{&(k1, k2) \mid k1 \in M1 \wedge k2 \in M2 \wedge \\ &argmaxP(S_i^{t_i}(k1, k2)|O_i)\} \\ L_i = \{&\} \end{aligned}$$

Multi-group activity case 2: $\forall$ $\exists$**.**

$$\begin{aligned} K_i &= \{(k1, k2) \mid k1 \in M1 \wedge \\ &\quad k2 = argmax_{k2 \in M2} P(S_i^{t_i}(k1, k2)|O_i)\} \\ L_i &= \{(l1, l2) \mid l1 \in M1^c\} \end{aligned}$$

Multi-group activity case 3: $\exists$ $\forall$**.**

$$\begin{aligned} K_i &= \{(k1, k2) \mid k2 \in M2 \wedge \\ &\quad k1 = argmax_{k1 \in M1} \prod_{k \in M2} P(S_i^{t_i}(k1, k)|O_i)\} \\ L_i &= \{(l1, l2) \mid l2 \in M2^c \wedge \\ &\quad l1 = argmax_{l1 \in M1} \prod_{k \in M2} P(S_i^{t_i}(l1, k)|O_i)\} \end{aligned}$$

Multi-group activity case 4: $\forall$ $\forall$**.**

$$K_i = \{(k1, k2) \mid k1 \in M1 \wedge k2 \in M2\}$$
$$L_i = \{(l1, l2) \mid l1 \in M1^c \vee l2 \in M2^c\}$$

Furthermore, we define K_i' and L_i' used for the calculation of $P(O_i|\neg S_i^{t_i}(M))$:

$$K_i' = \{\},\ L_i' = \{l \mid l \in M^c\}$$

Once M^* is computed based on K_is and L_is, it is applied to $P(G^t(M^*)|O)$ to calculate the posterior probability of the group activity occurred given the observation. Only when the probability $P(G^t(M^*)|O)$ is high, our system decides that the activity G occurred at the time interval t with its group members M^*. Figure 9 shows a pseudo of our MCMC algorithm.

```
GROUP_ESTIMATE(Activity $G$, Interval $t$, Actors $C_{1,\dots,n}$) {
    Group $M$ = {};
    Group $M^*$ = {};

    do {
      Randomly select transition move $tm$;
      Group $M'$ = apply $tm$ to $M$;

      for $i$ = 1 to $n$ (i.e. # of sub-events)
        ($K_i'$, $L_i'$)  = compute for $M'$, based on group
                           activity cases of Subsection 4.4

      Compute $\pi^t_G(M)$ with all $K_i$, $L_i$, $C_i$;   // Equation (7).
      Compute $\pi^t_G(M')$ with all $K_i'$, $L_i'$, $C_i$;

      $a = \pi^t_G(M') / \pi^t_G(M) * c1$;        // $c1$ is a constant.
      $r$ = random number between 0 and 1;
      if ($r$ < min(1, $a$)) {
        $M = M'$;
        if ($\pi^t_G(M) > \pi^t_G(M^*)$) $M^* = M$;
      }
    } while stabilized;

    return $M^*$;
}
```

Fig. 9 A pseudo code of our MCMC algorithm for finding groups with a maximum posterior probability.

5 Experiments

We implement the system presented in this paper, and test it to recognize high-level group activities such as 'group stealing' and 'group assault'. Notably, we are using CCTV videos that have been downloaded from *YouTube* as well as videos that we have taken in various environments. We implement and test our group activity recognition system for various types of group activities, while measuring the performance of our stochastic system compared to our previous deterministic recognition approach.

We have represented and recognized eight different types of group activities. 'Group move', 'group carry', 'group carry by signal', 'group fighting', 'intta-group fighting', 'group stealing', 'group arresting', and 'group assault' are the activities tested. 'Group move' indicates a group of people moving in the same direction and 'group carry' describes a group of people carrying a table or other large objects. We already have defined and represented 'group carry by signal' and 'group fighting' in Section 3.2. 'Intra-group fighting' is an intra-group version of group fighting. 'Group stealing' is a complex group-group interaction where one of thieves is stealing an object (e.g. laptop) while other thieves are distracting a group of owners of the object. 'Group arresting' indicates the situation where policemen are arresting a group of criminals. 'Group assault', which we discussed its representation in 3.3, is a highly stochastic group activity of people attacking a person with lots of variations.

A total of 45 sequences, ten videos for the 'group assault' and five videos for each of the other group activities, are tested to measure the performance. Videos downloaded from *YouTube* as well as videos taken with total of six participants in various environments have been collected. The videos were taken in 15 frames per second in the resolution of 320 * 240. The duration of a video varies depending on the type of the activity, ranging from 3 seconds to more than 30 seconds. As a result, approximately 12000 frames were obtained from 45 sequences for the testing.

We have used 5 separate videos sequences taken in a similar environment for training atomic actions of the human-human interaction 'fighting'. Note that the other individual-level interactions (e.g. taking an object, approaching, ...) are represented solely in terms of spatial relations among persons and objects, and thus do not require training. The object detector (e.g. head detector) also has been trained with separate training images from similar environments. The representation of group activities is encoded by a human expert, following our representation syntax. For example, the representation of 'group stealing', a group activity with 3 quantifiers, is as follows:

$$
\begin{array}{l}
\textbf{GroupStealing}(Group\,Thieves,\ Group\,Owners)\ =\ \{\\
\quad \exists a\ in\ Thieves,\ \forall b\ in\ Owners,\ \exists c\ in\ Thieves,\\
\quad list(\ def(t1, \textbf{Approach}(Thieves,\ Owners)),\\
\qquad\quad list(\ def(t2, \textbf{TakeObject}(a)),\\
\qquad\qquad\quad def(t3, \textbf{Distract}(c, b)))),\\
\quad and(\ equals(t2,\ this),\\
\qquad\quad and(before(t1,\ t2),\ during(t2,\ t3)))\\
\};
\end{array}
$$

Once the low-level part of our system is trained and the representation is encoded by the human expert, our group activity recognition system behaves fully automatically. The system was tested on all 45 sequences.

In order to make the recognition process more reliable, we made all representations (including the above 'group stealing' representation and other representations presented in 3.2) stochastic. Because of noisy observations and erroneous low-level components, some sub-events may not be detected correctly. We gave a small probability to the case where a sub-event is not occurring $P(\neg S_i^{t_i} | G^t(M))$, so that the overall probability of the group activity is high even when a portion of sub-events is not detected. The stochastic relationship between sub-events where described in the interaction 'group assault' only. For the probabilistic inference, the probability of each sub-event occurring when no group activity is present must be given as well: $P(S_i^{t_i} | \neg G^t(M))$. Our current system assumes that this has been correctly estimated and provided by a human expert or an automated learning system.

Figures 11 and 12 show the example sequences of group activities which our system successfully recognized. Bounding boxes have been drawn for each person's head or entire body, depending on the features used by the recognition system. Groups detected as a result of our algorithm are indicated using the color of bounding boxes. Figure 10 shows example time interval recognition results of the topmost sequence of Figure 11, the *YouTube* downloaded video of 'group stealing'.

Table 2 illustrates the final recognition accuracy of our algorithm. The type of each group activity is specified: GA stands for 'group action', GP for 'group-persons interaction',

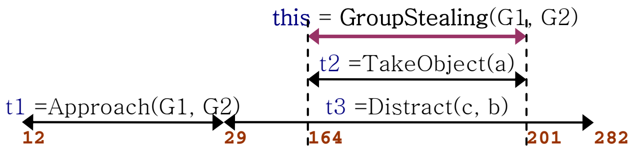

Fig. 10 Example time interval detection results of 'stealing'.

Table 2 Recognition accuracy of the system

Activity\System	Type	Qntfs.	Prev. system	Our system
Move	GA	∀	5/5	5/5
Carry	GA	∀	5/5	5/5
Carry by signal	GP	∀	4/5	4/5
Fight	GG	∀∃	3/5	5/5
Fight	IG	∃∃	3/5	3/5
Steal	GG	∃∀∃	4/5	5/5
Arrest	GG	∀∃	4/5	4/5
Assault	GG	∀∃∃	5/10	8/10
total			**33/45**	**39/45**

GG for 'group-group interaction', and IG for 'intra-group interaction'. Types of quantifiers attached to member variables of each activity are also listed. The performance is compared with the deterministic version of our system, presented in our previous paper [20]. The previous system maintains deterministic representations of group activities, and applies a heuristic algorithm to recognize them.

Only true positive rates are shown in Table 2. False positive rates were almost 0 in all cases with both systems, since recognizing multiple sub-events satisfying the specific relationship by 'mistake' is extremely unlikely. The recognition accuracy depends on inherent uncertainties and difficulties of the structures of the activities. Our stochastic system overcomes the limitations of the previous system, by making a hierarchical Bayesian inference. We are able to observe that our system performs superior to the previous method. The previous method did not perform well especially when recognizing 'group assault', since they are not able to handle variations in the activity structure.

6 Conclusions

We have presented a novel representation and recognition algorithm for complex high-level group activities. The technical contributions of this paper are the stochastic representation scheme to represent various types of group activities, and the new hierarchical algorithm for the probabilistic recognition. We presented recognition methodology for group activities with complex temporal, spatial, and logical structures, which has not been studied in depth previously.

In the future, we plan to study an automated learning of group activities. Currently, the representation of activities, including the probability associated with the occurrences of sub-events, are encoded manually by human experts. The stochastic representation of human activities suggests that automated learning is able to benefit the system with more accurate modeling of the activities.

Acknowledgement

This work was supported partly by the R&D program of the Korea Ministry of Knowledge and Economy (MKE) and the Korea Evaluation Institute of Industrial Technology (KEIT) [2008-S-031-01, Hybrid u-Robot Service System Technology Development for Ubiquitous City], and partly by Texas Higher Education Coordinating Board under award no. 003658-0140-2007.

References

1. J. K. Aggarwal and Q. Cai. Human motion analysis: A review. *Computer Vision and Image Understanding: CVIU*, 73(3):428–440, 1999.
2. J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.
3. J. F. Allen and G. Ferguson. Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4(5):531–579, 1994.
4. F. Cupillard, F. Bremond, and M. Thonnat. Group behavior recognition with multiple cameras. In *Proceedings of Sixth IEEE Workshop on Applications of Computer Vision (WACV)*, pages 177–183, 2002.
5. A. R. J. Francois, R. Nevatia, J. Hobbs, and R. C. Bolles. Verl: An ontology framework for representing and annotating video events. *IEEE MultiMedia*, 12(4):76–86, 2005.
6. S. Gong and T. Xiang. Recognition of group activities using dynamic probabilistic networks. In *IEEE International Conference on Computer Vision (ICCV)*, page 742, 2003.
7. A. Hakeem, Y. Sheikh, and M. Shah. CASEE: A hierarchical event representation for the analysis of videos. In *Proceedings of the 20th national conference on Artificial intelligence (AAAI)*, pages 263–268, 2004.
8. S. Hongeng, R. Nevatia, and F. Bremond. Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding: CVIU*, 96(2):129–162, 2004.
9. S. S. Intille and A. F. Bobick. A framework for recognizing multi-agent action from visual evidence. In *AAAI/IAAI*, pages 518–525, 1999.
10. Y. A. Ivanov and A. F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):852–872, 2000.
11. S. M. Khan and M. Shah. Detecting group activities using rigidity of formation. In *ACM Multimedia*, 2005.
12. Z. Khan, T. Balch, , and F. Dellaert. Mcmc-based particle filtering for tracking a variable number of interacting targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1805–1819, 2005.
13. L. Liao, D. Fox, and H. Kautz. Location-based activity recognition using relational markov networks. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence (IJCAI)*, 2005.

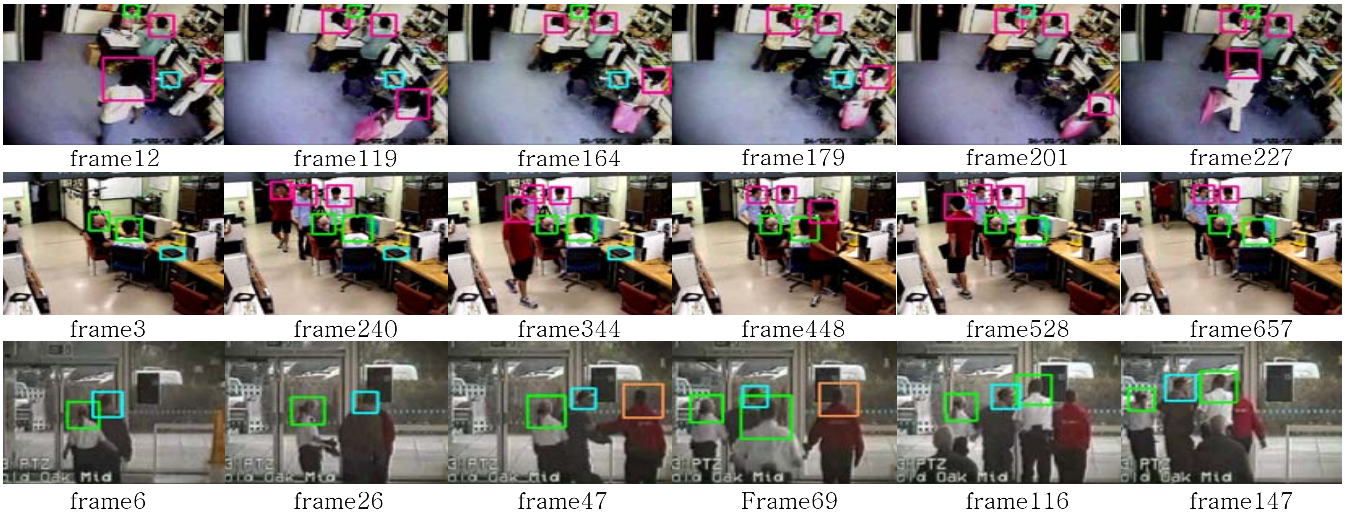

Fig. 11 Processed video sequences of group activities. The top-most sequence and the middle sequence are example videos of ’group stealing’. The bottom-most sequence is an example video of ‘group arresting’. The top sequence and the bottom sequence are from real CCTV videos that have been downloaded from *YouTube*, and the middle sequence is from the videos that we have taken in an office environment. In the case of ‘group stealing’, red bounding boxes are used to denote thieves, while green bounding box are used to denote owners. A cyan bounding box is used to label the object, a laptop. We are able to observe that the system recognizes group stealing from both sequences correctly, even though their video appearance information differs greatly. In the case of ‘arresting’, green boxes indicate policemen, and cyan boxes indicate persons being arrested. (This figure is best viewed in color.)

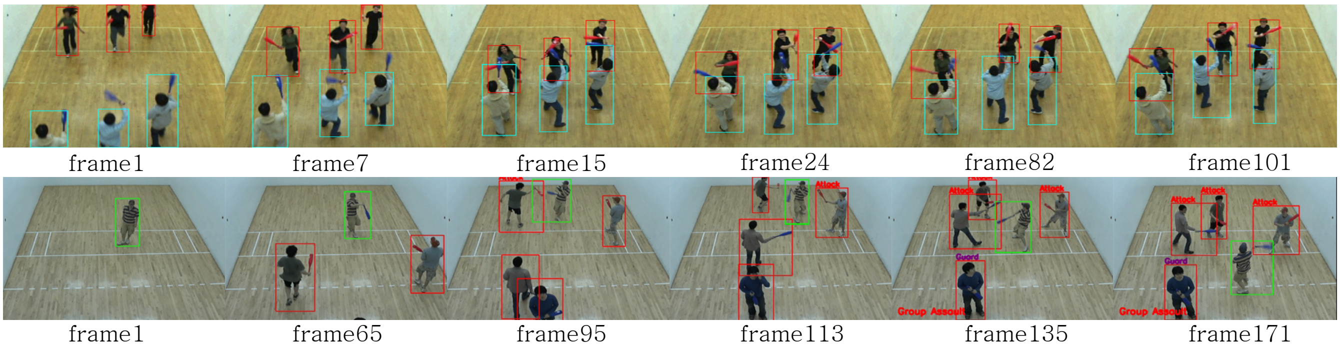

Fig. 12 Processed video sequences of group activities taken in a similar environment. The top sequence shows the group-group interaction ‘group fighting’, and the bottom sequence shows the ‘group assault’. Colors of the bounding boxes are used to distinguish the groups involved in the activities. Our system is able to recognize different types of activities in a similar environment. (This figure is best viewed in color.)

14. B. Milch, B. Marthi, S. Russell, D. Sontag, D. L. Ong, and A. Kolobov. Blog: Probabilistic models with unknown objects. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1352–1359, 2005.
15. N. M. Oliver, B. Rosario, and A. P. Pentland. A Bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.
16. S. Park and J. K. Aggarwal. A hierarchical Bayesian network for event recognition of human actions and interactions. *Multimedia Systems*, 10(2):164–179, 2004.
17. C. S. Pinhanez and A. F. Bobick. Human action detection using pnf propagation of temporal constraints. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, page 898, 1998.
18. M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.
19. M. S. Ryoo and J. K. Aggarwal. Observe-and-explain: A new approach for multiple hypotheses tracking of humans and objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.
20. M. S. Ryoo and J. K. Aggarwal. Recognition of high-level group activities based on activities of individual members. In *Proceedings of IEEE Workshop on Motion and Video Computing (WMVC)*, 2008.
21. M. S. Ryoo and J. K. Aggarwal. Semantic representation and recognition of continued and recursive human activities. *International Journal of Computer Vision (IJCV)*, 32(1):1–24, 2009.
22. J. M. Siskind. Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal of Artificial Intelligence Research (JAIR)*, 15:31–90, 2001.
23. X. Song and R. Nevatia. Detection and tracking of moving vehicles in crowded scenes. In *Proceedings of IEEE Workshop on Motion and Video Computing (WMVC)*, 2004.
24. B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2002.

25. S. D. Tran and L. S. Davis. Event modeling and recognition using markov logic networks. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 610–623, 2008.
26. P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, Nov 2008.
27. N. Vaswani, A. Roy Chowdhury, and R. Chellappa. Activity recognition using the dynamics of the configuration of interacting objects. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.
28. P. Viola and M. J. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.
29. V.-T. Vu, F. Brémond, and M. Thonnat. Automatic video interpretation: A novel algorithm for temporal scenario recognition. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1295–1302, 2003.
30. D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Modeling individual and group actions in meetings with layered hmms. *IEEE Transactions on Multimedia*, 8(3):509–520, 2006.