Discrete Relative States to Learn and Recognize Goals-based Behaviors of Groups

Jérémy Patrix[*]
GREYC (UMPR 6072)
University of Caen, France
jeremy.patrix@unicaen.fr
&
IPCC department,
CASSIDIAN, France
jeremy.patrix@cassidian.com

Abdel-Illah Mouaddib[†]
GREYC (UMPR 6072)
University of Caen
Basse-Normandie, France
abdel-illah.mouaddib
@unicaen.fr

Simon Le Gloannec[‡]
& Dafni Stampouli
& Marc Contat
IPCC department,
CASSIDIAN
Val-de-Reuil, France
{simon.legloannec,marc.contat
dafni.stampouli}@cassidian.com

ABSTRACT

In a crisis management context, situation awareness is challenging due to the complexity of the environment and the limited resources available to the security forces. The different emerging threats are difficult to identify and the behavior of the crowd (separated in groups) is difficult to interpret and manage. In order to solve this problem, the authors propose a method to detect *threat* and understand the situation by analyzing the *collective behavior of groups* inside the crowd and detecting their *goals*. This is done according to a set of learned, *goal-based, group behavior* models and observation sequences of the group. The proposed method computes the group estimated state before using *Hidden Markov Model* to recognize *the goal by the group behavior*. A realistic emergency scenario is simulated to demonstrate the performance of the algorithms, where a suicide-bomber wearing a concealed bomb enters a busy urban street. The proposed algorithms achieve the detection of the dangerous person in the crowd, in order to raise an alert and also predict casualties by identifying which groups did not notice the threat. *Complex Event Processing* is used to compare and evaluate the results. The algorithms were found more precise and more tolerant to noisy observations.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Distributed Artificial Intelligence;
I.5 [**Pattern Recognition**]: Models, Design Methodology;
I.6 [**Simulation and Modeling**]

General Terms

Algorithms, Measurement, Security, Human Factors

Keywords

Agent-based simulation:Emergent behavior; Systems and Organisation: Complex systems, Self-organisation;

1. INTRODUCTION

In this paper, we present a part of a system which has been developed within the *EUSAS* (*European Urban Simulation for Asymmetric Scenarios*) project. The project is financed by twenty nations under the *Joint Investment Program Force Protection* of the *European Defense Agency* (*EDA*). Within this project, we aim to obtain improved situation awareness by using new methods for detecting high-level information from observed variables of the monitored crowd. Previous work [21] developed methods to assess risks during social gatherings in the case of emergency and panicked crowd evacuation, by identifying problematic escape paths and possible stampedes and points of crowd crushing. The first methods allow training, recognition and anticipation, in real-time, of the recurrent emerging collective movements of panicked agents (each agent corresponds to a person in the crowd). Then, in order to detect the reasons of this type of behaviors, we developed methods for group detection. This paper presents consequent work on **the detection of these groups' behavior, their goals and intentions**, and the detection of the reasons that caused these behaviors. Group behavior modeling methods were used, such as *Hidden Markov Model* (*HMM*) by learning from observations of successive states of members and which extract high-level facts of current behaviors as *Complex Event Processing* (*CEP*).

Real-time detection of a multi-agent behavior is a highly complex problem. Indeed, if the group state is defined based on the states of its members, a model of group behavior will have a number of the variables $|A|.|S|.t$ where A is the set of agents and S is the number of variables for each agent state from the start of the observation till the current time step t. Therefore, the size of the behavior model becomes intractable, over time.

Our approach consists in defining a set of ***discrete relative* (*DR*) state** sequences that **model *goal-based group behavior***. The set of *DR* states represents a discretization of the evolution between two successive times concerning the difference between a goal state and a centroid state of a group. This centroid of the group is computed from local states of people of the group. By this way, the number of agents/people in the group has no impact on the complexity of the recognition problem of a collective behavior.

An important fact about the ***goals*** of a group is that they are all entities of the environment (excluding the members of the current

[*] Author of this article, is both in the IPCC team (*Information Processing Control and Cognition*), a part of the *System Design Center* of CASSIDIAN France & MAD team of the GREYC laboratory.

[†] Professor, head of MAD team (*Model, Agent and Decision*) .

[‡] R&D engineers in the IPCC team working on decision support systems for civilian authorities using field of artificial intelligence (decision and robotic), information fusion and semantic reasoning.

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, Ito, Jonker, Gini, and Shehory (eds.), May 6–10, 2013, Saint Paul, Minnesota, USA.

group, such as other people, other groups, locations, critical infrastructures). The action sequence of a group has been decided according to these external *goals*/entities, for example: a group is following this target or a group is fleeing from an identified threat. The proposed method provides a probability value denoting the likelihood that this group is executing this behavior according to the specific goal, thus they are called *goal-based group behaviors*.

The algorithm is used to obtain the most likely groups and goals among observed agents. The probability that such group executes such collective behavior to reach a specific goal is established according to an observation sequence and a *goal-based group behavior* model. During the simulation, we used a highly realistic virtual environment, as presented in the last section.

Section 2 and 3 present the background and previous work relevant to our framework, followed by our contribution in section 4. Finally, section 5 presents the results from the experiments in the virtual environment.

2. BACKGROUND

In order to facilitate the comprehension of our model and the suitability of our experiments, this section presents relevant background knowledge in the areas of teamwork analysis.

The objective of the teamwork analysis [26] is to identify the team's goal, the executed actions of members to achieve the goal and the coordination patterns between its members. Three general classes of characteristics are then potentially valuable for the team's activity/plan recognition [26]: (1) spatial relationships between team members and/or physical landmarks that remain fixed over a period of time; (2) temporal dependencies between behaviors in a plan or between actions in a team behavior; and (3) the coordination constraints between agents and the actions that are performed. Some properties of the observed interactions are invariant over all possible permutations of behavior, but the multi-agent activity/plan recognition includes two difficulties [26] : (1) the multi-agent behaviors are more variable in the observed spatio-temporal traces than simple behavior; (2) the team composition can change over time.

The observation sequence of agent behavior can be segmented to obtain its state sequence. In real-time, this sequence is continuous and thus must be tested at regular intervals. This requires additional memory for the state space of observed data for the learning and additional execution time for the methods that learn and recognize the behavior models.

In our model, we use *Hidden Markov Models* (*HMM*) that allows representing a behavior on a set of *hidden* states with a transition probability distribution between these states. All *observable state sequences* of this behavior allows us to obtain the transition probabilities.

We adapt **the *HMM* definition** [23] by this 5-tuple $\langle Q, A, O, B, \Pi \rangle$:

Q is a finite set of hidden states, where q_i is the i^{th} hidden state;

A is a finite set of transition probabilities, where $a_{ij} = Pr(s_{t+1} = q_j | s_t = q_i)$ is the transition probability that the hidden state will be q_j after q_i ;

O is a finite set of observable states, where o_i is the partially observable state of q_i (as obtained from a monitored system after using data fusion methods to reduce the noise);

B is a finite set of observation probabilities, where $b_{ik} = Pr(o_k|q_i)$ is the probability to observe o_k in the state q_i;

Π is a finite set of initial state probabilities, where $\pi_i = Pr(s_0 = q_i)$ is the probability to be initially in the state q_i.

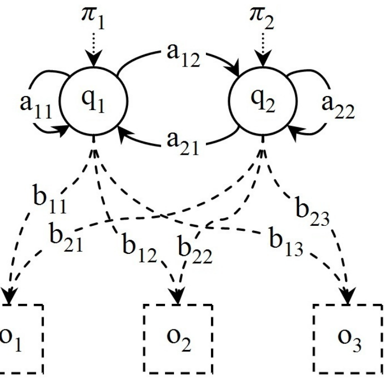


Figure 1: An example of HMM with two hidden states and three observable states

This system is kept under following constraints

$$\forall i \in Q, \quad \textstyle\sum_{j=0}^{|Q|} a_{ij} = 1; \quad \sum_{k=0}^{|Q|} b_{ik} = 1; \quad \sum_{m=0}^{|Q|} \pi_m = 1;$$

and allows three main computations [23]:

1. The probability of the current state by the equation $Pr(s_t = q_t|O_t)$ that gives the most likely hidden state q_t at the time t knowing an observation sequence $O_t = \{o_1, ..., o_t\}$. It is solved by a dynamic programming algorithm such as *Forward-Backward* [22];

2. The most likely hidden state sequence knowing an observation sequence. It is solved by the *Viterbi* algorithm [22];

3. The *HMM* learning is able to give the maximum probability when it estimates a sequence. It is solved by the *Baum-Welch* algorithm [22] that uses the *Expectation-Maximization* method.

In our work, different levels of observations and modeling produce a stochastic representation of a group behavior: (1) the *micro*-level is about the roles of the agents impacting on their possible individual activities; (2) the *meso*-level is about the activities executed in teamwork; (3) the *macro*-level is about the goals that explain the interactions between each group and the environment.

To model a behavior of agents, the model must be canonical [27] whatever the position, the speed, the orientation, the observation scale, the number of agents and goals. For a canonical representation of the team state, the concept of *centroid* is used [27]. This corresponds to a vector where each variable is the average value of states variables of the team members. Thus, our team's behaviors are defined on their relative movement and not on the absolute variables of agents to be recognized in any monitored situations. The extracted characteristics of the behavior models could be classified by minimal information and discrete according to their type of agent, environment or team. Such examples are the *Centroid-Relative Position Vector* [16] or the *Role-Relative Position Vector* that change according to members' roles during team behavior.

$$p^c_{x,y,t} = \frac{\sum_{i=1}^{|c|} p^{a^i}_{x,y,t}}{|c|} \qquad (1)$$

At time t, the group position $p^c_{x,y,t}$ of the centroid state s^c_t is computed by the average of positions $p^{a^i}_{x,y,t}$ of each agent a^i of the group c. The *centroid* state s^c_t includes all required information of the agents (such as position and speed).

In conclusion, a *centroid* state formalizes a group state with an unspecified number of members. The sequence of these states characterizes team behavior, and the behavior can be learned and recognized using *HMM* methods, thus achieving detection of group behavior. Our model presents the innovative points that consist of a *discrete relative* state sequences which are used to formalize and detect *goal-based group behaviors*. Before the proposed method is explained in greater detail, previous related work is presented in the following section.

3. RELATED WORK

There are several studies that investigated the requirements, advantages and limitations of methods on group behavior and goal detection. This section describes the evolution of relevant technical solutions to demonstrate the advantages achieved by the proposed model.

The behavior planning and detection have a difference in the direction of information propagation like in the *Dynamic Bayesian Networks* [11] that propagates it in the two directions and model time influence. The probabilistic plan/activity recognition [7, 25] of *Bayesian Network* (representing links between events/actions with a partial order constraint) is applied by an inference algorithm exactly as the *Viterbi* algorithm on *HMM* (computing the posterior probabilities of distribution on possible interpretations).

In order to model *complex behaviors*, the main idea has been the decomposition of the parameter space to decrease their complexity: in initial, accept, intermediate and reject states (*Behavior HMM* [12]); or until each observed agent characteristic (*Observation Decomposed HMM* [14]); or for each distinct temporal granularity level and each kind of information (*Layered HMM* [20]); or (*Switching Hidden Semi-Markov Model* [9]) using a duration distribution for each state to consider that a state remains unchanged for a certain duration before its state transition. Group behavior can be executed by a variable number of agents, thus the decomposition solution increases the complexity of the problem (because each group behavior should be decomposed and learned for each possible team size and roles members).

In order to detect *team behaviors*, the probabilistic recognition of agent's role can be executed before the team activity recognition to improve the detection quality. The *Unique or Multiple Role Assignment* algorithm [16] selects agents for each identified valid roles having the highest probabilities (travelling from the root to the leafs of roles in a probabilistic decision tree). For the team activity recognition, they represent an *Idealized Team Action* [16] after the normalization of observations to deal with the noise. This method's limitation is that it must produce a library of the same team activity for each member condition, as in the *Multi-Agent Plan Recognition* [1]. In order to detect a team's behavior, this method identifies the structures of agent behaviors in the team by a matrix, where p_{ij} is the expected action of the member of the line j at the step of the column i and must be compare in each team plan P.

Recent works have been focused on *Abstract HMM* [5] models which are used to recognize an agent policy on long term. Each upper level abstract behavior includes lower level simple action sequences. Therefore to manage its explosion in the state space, a hybrid inference method of *Factored State-AHMM* [29] is used to remove the redundant probabilistic links of state transitions.

Each state of each level of a *Hierachical HMM* [4] can be itself a *HHMM*. This *hierarchical policy* recognition of a team is based on a multi-agent decision method called *Hierarchical Multiagent Markov Process* [24] that coordinates actions of agents on a upper level by a controller and without coordination on a lower level. An end state for each level indicates when the activity is finished in order to go to the upper level state.

Using the *Joint Probabilistic Data Association Filters* method, *HHMM-JPDAF* [18] recognizes the *behaviors of multiple persons* on its set of *HHMM* and the noisy observations. Its learning algorithm discovers [17] the *HHMM* structure and estimates the link probabilities by the *Expectation-Maximization* method.

For recognition on long term, a memory node of each level in the *Abstract Hidden Markov mEmory Model* (*AHMEM*) [3] memorizes the actions sequence that is executed according to the chosen activity by the current policy of the upper level. Knowing a noisy observation sequence, an approximate probabilistic inference computes, in real-time, the probability of various possible behaviors of *AHMEM* [19].

In addition to a real-time probabilistic behavior recognition of observed multi-agent system, *M(ulti-agent)-AHMEM* [10] allows a contextual, non-deterministic approach and a high level multi-modal fusion of various sources using an ontology as library of behaviors describing the possible relationships between entities and the environment.

The learning of these last multi-agent behavior models is more and more complex according to the multiple kinds of states. Furthermore, the use of approximate inference methods for a real-time detection decreases the quality of detection. This raises the question whether there are alternative ways to *HMM*.

Techniques based on learning could be an alternative to HMM. *Hierarchical Conditional Random Field* [13, 30] encodes relations between observations including directly the conditional distribution, or *Inverse Reinforcement Learning* (*IRL*) [8] describes the intention/goal/behavior of each agent by computing a reward function to reproduce the observed policy of the agent. However, these techniques suffer from the need of significant data/time for learning and are not suitable to detect goal-based group behaviors.

Like an alternative to the *Bayesian inference*, the *Dempster-Shafer theory* is useful to assign any agent to a team and behavior for each time step. The *STABR (Simultaneous Team Assignment and Behavior Recognition)* algorithm [28] creates a set of potential teams. The formation recognition by the *RANSAC (Random Sampling and Consensus)* algorithm generates assumptions of teams that explain all the observed space-time traces of members. The advantages are the quick behavior recognition in comparison with *HMM*s and the team member reassignment. But it cannot guaranty obtaining the best match between the observed data of the team behavior and the selected behavior models among all assumptions.

Another possible solution is the *Complex Event Processing* (*CEP*) [15] that includes an inference engine that uses the *forward-* and/or *backward-chaining* on its rules to detect the behaviors and situations requiring a reaction. *CEP* can be represented using levels of the *JDL Data Fusion model* [2]: (1) the standardization rules on various sources, (2) the filter rules and the simple rules execution on *Complex Events* (*CE*s), (3) the aggregation rules for new *CE*s and the detection rules for significant items in an events cloud, (4) a feedback loop. But the inference engine has difficulties to deal with a great volume of facts with regular updates by many *CE*s and production rules. We have already implemented and often used this technique and it will be used as the comparison method with our model during experimentation.

In conclusion, three main approaches are presented in this state of the art section: probabilistic, grammatical and statistical / fuzzy methods. **All these approaches are unable to detect in the same way the collective behavior and the target goal**. The following section presents the proposed approach based on an aggregated state (centroid) of the group allowing the recognition of the behavior and the goal of the group, in real-time.

4. THE MODEL OF GOAL-BASED GROUP BEHAVIOR

In the framework of the *EUSAS* project, an algorithm, similar to the one presented in [6], detects groups of agents. This study advances the previous work in order to achieve *goal-based group behavior* detection. In this section, we present how to compute a *discrete relative* (*DR*) state, how to model a collective behavior according to a goal and how to detect *goal-based group behavior*. In the study, the state of an agent contains only its position p and its speed v.

4.1 Discrete Relative states

To compute a *DR* state as in Figure 2, we get observations $\{o_t^{a^i}, o_t^{g^j}\}$ on the hidden states $\{s_t^{a^i}, s_t^{g^j}\}$ for each agent a^i of the group c and each goal $g^j \notin c$ (which could be mobile) from the time 0 to t. Each o_t and s_t contains the position and the speed $\langle p_t, v_t \rangle$.

1. We compute $\{s_{t-1}^c, s_t^c\}$ the centroid states at time $t-1$ and t of a group c using equation 1.

2. We compute relative states $\{\Delta s_{t-1}^{c,g}, \Delta s_t^{c,g}\}$ on the difference between centroid states $\{s_{t-1}^c, s_t^c\}$ and goal states $\{s_{t-1}^g, s_t^g\}$. The relative variable $\Delta r_t^{c,g} \in \Delta s_t^{c,g}$ is given by:

$$\Delta r_t^{c,g} = \sqrt{|r_{x,t}^c - r_{x,t}^g|^2 + |r_{y,t}^c - r_{y,t}^g|^2} \qquad (2)$$

   where $r_{x,y,t}$ is a spatial variable (as $p_{x,y,t}$ of the equation 1).

3. We discretize the relative states $\{\Delta s_{t-1}^{c,g}, \Delta s_t^{c,g}\}$ on their difference from $t-1$ and t for each variable $\Delta r^{c,g} \in \Delta s^{c,g}$. The *discrete relative variables* $\{dr_n, dr_d, dr_c, dr_i\}$ are introduced to represent any *DR* state which are, in our case, the *DR* positions $\{p_n, p_d, p_c, p_i\}$ and the *DR* speeds $\{v_n, v_d, v_c, v_i\}$. These *DR* variables are given by:

$$\begin{array}{lll} dr_n \text{ (is null)} & & \Delta r_t^{c,g} < \delta r^{min} \\ dr_d \text{ (decrease)} & \text{if} & \Delta r_t^{c,g} < \Delta r_{t-1}^{c,g} - \delta r_{min} \\ dr_i \text{ (increase)} & & \Delta r_t^{c,g} > \Delta r_{t-1}^{c,g} + \delta r_{min} \\ dr_c \text{ (is constant)} & & |\Delta r_t^{c,g} - \Delta r_{t-1}^{c,g}| < \delta r_{min} \end{array}$$

The value δr_{min} indicates a minimal state transition between two successive times according to the variable $r \in s$, thus it is directly proportional with the number of obtained observations per seconds. If we take the relative position p and speed v between a group and a goal from $t-1$ to t, the computed set of *DR* states represents:

$$Q^{DR} = \begin{pmatrix} \langle p_n, v_n \rangle & \langle p_n, v_d \rangle & \langle p_n, v_i \rangle & \langle p_n, v_c \rangle \\ \langle p_d, v_n \rangle & \langle p_d, v_d \rangle & \langle p_d, v_i \rangle & \langle p_d, v_c \rangle \\ \langle p_i, v_n \rangle & \langle p_i, v_d \rangle & \langle p_i, v_i \rangle & \langle p_i, v_c \rangle \\ \langle p_c, v_n \rangle & \langle p_c, v_d \rangle & \langle p_c, v_i \rangle & \langle p_c, v_c \rangle \end{pmatrix}$$

The advantage of this discretized state space is to facilitate the *HMM* learning and this kind of discretized state is human-readable. The *DR* state $\langle p_n, v_n \rangle$ represents a null discrete relative position p_n and a null discrete relative speed v_n (the group position and speed is equivalent to the goal position and speed) from $t-1$ to t.

In a sequence of *DR* states, we can show the relative movement over time between a group and a goal. For example, the three following *DR* state sequences show three different *goal-based group behaviors*:

$\Delta S_{0,4}^{c,g} = \{\langle p_n, v_n \rangle, \langle p_n, v_i \rangle, \langle p_i, v_i \rangle, \langle p_i, v_c \rangle, \langle p_i, v_c \rangle\}$
$\Delta S_{5,9}^{c,g} = \{\langle p_c, v_d \rangle, \langle p_c, v_i \rangle, \langle p_c, v_c \rangle, \langle p_c, v_c \rangle, \langle p_c, v_d \rangle\}$
$\Delta S_{10,14}^{c,g} = \{\langle p_d, v_c \rangle, \langle p_d, v_c \rangle, \langle p_d, v_d \rangle, \langle p_n, v_d \rangle, \langle p_n, v_n \rangle\}$

First during $\Delta S_{0,4}^{c,g}$, the group c quitted the goal g and has moved away from it, second during $\Delta S_{5,9}^{c,g}$, c has moved around g at the same distance, and last during $\Delta S_{8,12}^{c,g}$, c has moved towards g and is now close to it.

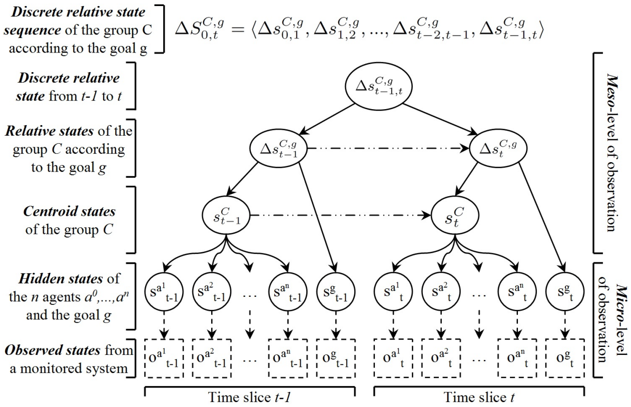

Discrete relative state sequence of the group C according to the goal g
$\Delta S_{0,t}^{C,g} = \langle \Delta s_{0,1}^{C,g}, \Delta s_{1,2}^{C,g}, ..., \Delta s_{t-2,t-1}^{C,g}, \Delta s_{t-1,t}^{C,g} \rangle$
Discrete relative state from t-1 to t
$\Delta s_{t-1,t}^{C,g}$
Relative states of the group C according to the goal g
$\Delta s_{t-1}^{C,g}$
$\Delta s_t^{C,g}$
Centroid states of the group C
s_{t-1}^C
s_t^C
Hidden states of the n agents $a^0,...,a^n$ and the goal g
$s^{a^1}_{t-1}$
$s^{a^2}_{t-1}$
...
$s^{a^n}_{t-1}$
s^g_{t-1}
$s^{a^1}_t$
$s^{a^2}_t$
...
$s^{a^n}_t$
s^g_t
Observed states from a monitored system
$o^{a^1}_{t-1}$
$o^{a^2}_{t-1}$
...
$o^{a^n}_{t-1}$
o^g_{t-1}
$o^{a^1}_t$
$o^{a^2}_t$
...
$o^{a^n}_t$
o^g_t
Time slice t-1
Time slice t
Meso-level of observation
Micro-level of observation

Figure 2: A discrete relative state computed from a set of observable states of n agents of a group c and a goal g

An observed sequence of this *DR* states can be seen as a low level group behavior sequence of a recurrent high level group behavior. Each group's behavior has its own set of recurrent state sequences and we use *HMM* to discover it.

4.2 Goal-based Group Behavior Assignment

To produce *goal-based group behaviors* detection, we consider the tuple $\langle O_t^{c,G}, \Lambda, F \rangle$ where:

$O_t^{c,G}$ $= \{\{o_0^{a^1}, o_0^{a^2}, ..., o_0^{a^{|c|}}, o_0^{g^1}, ..., o_0^{g^{|G|}}\}, ..., \{o_t^{a^1}, ..., o_t^{g^{|G|}}\}\}$ is an observation sequence of each agent of the group c and each goal of G from 0 to t, where $\{o_t^{a^i}, o_t^{g^j}\}$ is an observation of the i^{th} agent and the j^{th} possible goal that an observed group c can seek to reach;

Λ is a set of *HMM*s, where $\lambda^h \in \Lambda$ is the *HMM* modeling the h^{th} defined group behavior;

F is a set of functions, where f^h is a function (i.e. section 4.1) that computes the *DR* state sequence ΔS_t^h from an observation sequence according to *HMM* λ^h.

$Pr(\lambda|O_t)$ is the probability that the behavior λ is executed during the observation sequence O_t (computed by *Forward-Backward* algorithm [22]). During the T last observations (in order to capture the recent behavior), the computation of observations sequence $O_{t-T,t}^{c,g^j}$ in *DR* states sequence $\Delta S_{t-T,t}^{c,g^j}$ is given by:

$$M_t^c(\lambda^h, g^j) = Pr(\lambda^h | \Delta S_{t-T,t}^{c,g^j} = f^h(O_{t-T,t}^{c,g^j})) \qquad (3)$$

For each group behavior λ^h and each possible goal g^j, we compute the probability (3) that a group c is executing λ^h to reach g^j. The result gives us thus the following matrix:

$$M_t^c = \begin{pmatrix} M(\lambda^1, g^1) & \cdots & M(\lambda^1, g^j) & \cdots & M(\lambda^1, g^{|G|}) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ M(\lambda^h, g^1) & \cdots & M(\lambda^h, g^j) & \cdots & M(\lambda^h, g^{|G|}) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ M(\lambda^{|\Lambda|}, g^1) & \cdots & M(\lambda^{|\Lambda|}, g^j) & \cdots & M(\lambda^{|\Lambda|}, g^{|G|}) \end{pmatrix}$$

We obtain the most likely tuple $\langle \lambda^H, g^L \rangle$ by:

$$M(c,t) = argmax_{\forall \lambda^h \in \Lambda}^{\forall g^j \in G} \{M_t^c(\lambda^h, g^j)\} \qquad (4)$$

Here, only one *goal-based group behavior* is selected for the group by the best probability of M_t^c. But a group can execute different group behaviors for different goals at the same time. We obtain the most likely group behavior λ^H for each goal $\forall g^j \in G$ by the result of:

$$M(c, g^j, t) = argmax_{\forall \lambda^h \in \Lambda}\{M_t^c(\lambda^h, g^j)\} \quad (5)$$

In the same way, for another use case, for a group, we could obtain $M(c, \lambda^i, t)$ the most likely goal g^h for each group behavior λ^i on M_t^c.

To obtain a probability on the recent behavior of agents, we keep only the T last observations. We will show in experiments the impact on the detection quality by changing the observation sequence length T for the learning and the recognition.

In order to understand, at this point, following the instructions of this section, we model a *goal-based group behavior* using *discrete relative* state sequences computed from observation sequence of monitored groups of agents and their possible goals. The following section presents our method to extract the specific goals that are responsible of observed behavior sequences of groups.

4.3 Valued Goal-based Group Behavior

In a monitored situation with lots of possible goals for a group, we detect its current *goal-based group (GbG)* behaviors. In order to detect if the group is executing a sequence of *GbG* behaviors according to a particular goal (its intention), we measure its value using the following process that we call *Valued Goal-based Group Behavior Detection*.

This value can be measured in different ways according to the kind of *intention* that we want to detect. From the time $t - T$ to t, we use the following *goal values* in order to represent:

- $V_{t-T,t}^{c^i,g^j}$ the *intention* of the group c^i according to the goal g^j;
- $V_{t-T,t}^{c^i,G^j}$ the *intention* of the group c^i according to a set of goals G^j ($\in G$, for example: all the security forces, the critical infrastructures, the safe areas or the dangerous areas...);
- $V_{t-T,t}^{C^i,g^j}$ the *intention* of a set of groups C^i according to the goal g^j.

In order to measure this *goal value*, we use a *distribution* where a goal g and a *GbG* behavior λ^h of a group c give the value $V(c, g, \lambda^h)$. A distribution is created according to the intention ϕ^p ($\in \Phi$ the set of defined intentions) that we want to detect by the observed *GbG* behavior sequence. The advantage of a distribution is that it can be created or learned to detect each type of intentions that is needed. So, the equation of the *goal values* are:

$$V_{t-T,t}^{\phi^p,c^i,g^j} = \sum_{t_c=t-T}^{t} V^{\phi^p}(c^i, g^j, \lambda^h = M(c^i, g^j, t_c)) \quad (6)$$

$$V_{t-T,t}^{\phi^p,c^i,G^j} = \textstyle\sum_{g^l \in G^j} V_{t-T,t}^{\phi^p,c^i,g^l} \quad ; \quad V_{t-T,t}^{\phi^p,C^i,g^j} = \textstyle\sum_{c^l \in C^i} V_{t-T,t}^{\phi^p,c^l,g^j}$$

For example, in order to detect the civilians with a dangerous intention ϕ^p for themselves: the distribution gives a positive value when civilians move towards a safe area and a negative value when the groups move toward a dangerous area, and after all groups executing ϕ^p will have a higher positive *goal value*. According to the distribution of the intention $\phi^p \in \Phi$, the group c is moving toward the specific long-term goal $g^{\phi^p,c}$ is:

$$g_{t-T,t}^{\phi^p,c} = argmax_{\forall g^j \in G} V_{t-T,t}^{\phi^p,c,g^j} \quad (7)$$

According to the monitored situation and what we search to identify, a *valued goal-based group behavior* indicates the intention of the group on the long-term on two levels of observation: firstly, a general information about its specific goal, and secondly, if its intention is normal (friend) or abnormal (foe). Thus the ability to learn and recognize *valued goal-based group behaviors* using *HMM* methods in real-time is shown during the following experimentation section.

5. EXPERIMENTS

During the experiments, the proposed method of the *relative* states in *HMM* is compared with *CEP*. Our implementation in *java* communicates with *Virtual Battle Space 2* (VBS2) [1] a highly realistic simulated environment and can manage *HMM*s and *CEP*s. For the *HMM*s, it uses a *jahmm* [2] library available on-line (managing states, observations and standard algorithms relating to *HMM*s). It uses also *Drools* a *CEP* rule engine (leading java open source rule engine federated in *Jboss*). During a scenario in this software, it extracts essential information of agents to apply the proposed model of detection and it displays the results in real-time.

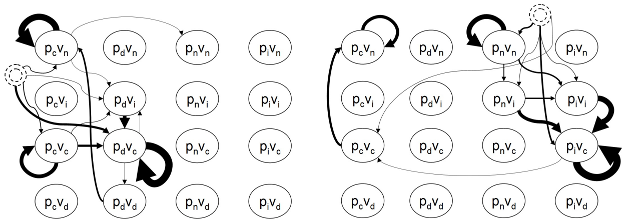
This section begins with the *HMM* learning of behaviors. The following subsection presents the results from the experiments in order to show how the proposed model can be used to help in a crisis management situation. This section also includes an experiment of a simulated crowd during a social gathering with an human carrying a bomb that moves toward a press reporter (in order to detect the abnormal behavior of the crowd).

5.1 The HMM learning of behaviors

For the proposed use case, between a group and a goal, the probability of four different *GbG* behaviors needs to be determined in real-time. These behaviors are: $\lambda^{c \rightarrow g}$ moving towards a goal, $\lambda^{c \leftarrow g}$ moving away from a goal, $\lambda^{c \circlearrowright g}$ moving around a goal and $\lambda^{c \nleftrightarrow g}$ not moving according to a goal.

In order to define *HMM* on a *goal-based group behavior*, a learning model on the observation data resulting from various scenarios is created using the *Baum-Welch* algorithm [23]. This algorithm enables us to maximize probabilities of each given observation sequence by updating the parameters A, B and Π (i.e. section 2).

During the simulated scenarios, the teams are composed of 4 to 16 civilians that move in group formation. There is some (mobile or not) goals (defined by other agents or specific positions). The environment has also few obstacles that the teams avoid. The observations of each position and speed of agents is obtained each second to compute their *DR* state sequences (i.e. the process is described in section 4.1).



(a) The team is moving towards the goal.

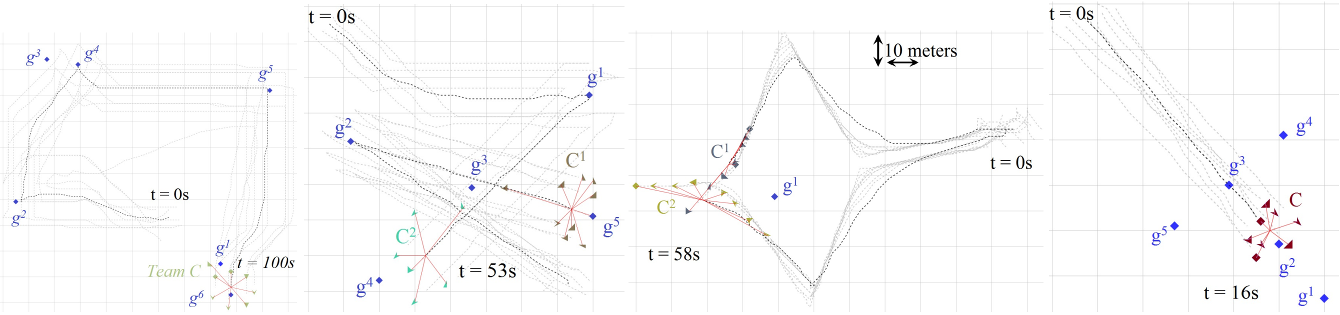
(b) The team is moving away from the goal.

Figure 3: Two HMM examples of *goal-based group behaviors* that have been learned (and they have their own transition probabilities between the discrete relative states).

Two learning results are shown on Figure 3. Each of 16 *DR* states (based on the relative positions and the relative speeds) has a

[1] Bohemia Interactive Simulations, Australia, 2011, http://vbs2.com

[2] J.M. Francois, Jahmm (*Java Hidden Markov Model*), 2006, http://code.google.comp/jahmm/



(a) A team has moved through four critical infrastructures and a security force followed it.

(b) Two teams have moved toward two different goals.

(c) Two teams have avoided a dangerous area.

(d) A team is moving toward a safe area while avoiding two dangerous goals.

Figure 4: Examples of monitored experimentations.

transition probability not null to go to another *DR* state. Edges represent their transition probabilities (with a proportional size only if they have a probability $A_{ij} > 1/|Q^{DR}| = 0.0625, \forall i,j \in Q^{DR}$). Each double circle indicates us the possible initial states of the behavior. The *GbG* behavior of Figures 3(a) and 3(b) are completely opposed and we can see that their initial and final *DR* state are almost at the opposite (as $\langle p_n, v_n \rangle$ that is the final *DR* state of 3(a) and the initial state of 3(b)). According to each behavior, only a subset of all *relative* states is used for learning and the transition probabilities are different even if the two behaviors overlap. So, we can learn various different *GbG* behaviors using our method.

5.2 HMM versus CEP

In order to compare the results of the proposed *discrete relative* states on *Hidden Markov Model*, the *Complex Event Processing* was used as an alternative method because it is often used for the complex event detection [2, 15]. The following experimentations of detection are produced on a set of scenarios where collective movements are simulated by a multi-agent system in *VBS2*.

Drools was used to write rules that take observations of agents in low level facts and extract high level facts of collective behaviors and goals of groups. These rules compute the same function (as explained in section 4.1 and 4.2) to create high level facts including the same result of the centroids, the *DR* state sequences and M_t^c (the most likely collective behavior for each goal for a group).

The illustrations of Figure 4 are examples of monitored situations used to test the precision of our detection method. The dotted lines show the paths used by the teams (black for the centroid of each team and grey for each member). Each group has its own color and a red line links each member of the centroid. For each group during the detection, note that the other entities (alone agents, groups or important positions) compose the set of its possible goals and are presented by blue diamonds.

During the experimentations, the behavior detection quality is measured by the percentages of the *precision* expressed by A/B: where A is a value that increases one by one point when a detection corresponds correctly to the observation, and B is a value that increases one by one point when the system obtain a new observation that must be detected.

With the four *GbG* behaviors $\langle \lambda^{c \rightarrow g}$, $\lambda^{c \circlearrowright g}$, $\lambda^{c \leftarrow g}$, $\lambda^{c \nleftrightarrow g} \rangle$, we added an inconsistent *GbG* behavior $\lambda^{c \leftrightsquigarrow g}$ that is a *HMM* by default without learning. If $\lambda^{c \leftrightsquigarrow g}$ has the highest probability then the observation sequences used during the behavior learning require further corrections.

Table 1 presents the tests for the *GbG* behavior learning and recognition on different lengths of observations sequences from 2 to ∞ (by an iterated method). The table presents results up to a

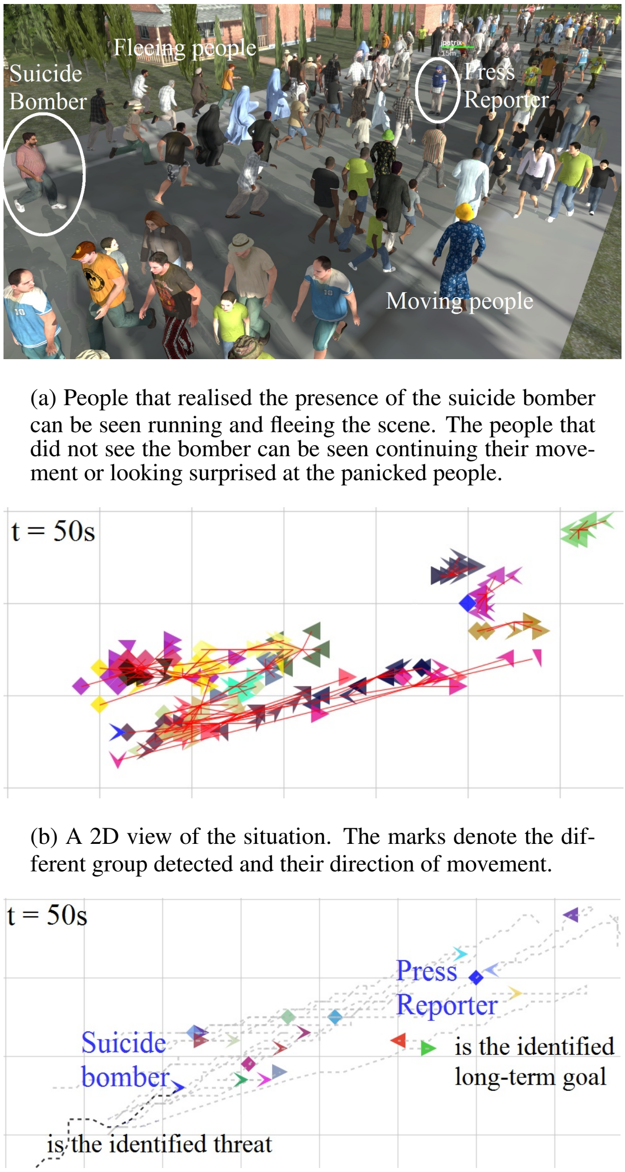
Length of recognized sequences							
1	CEP	58,06 %					
	HMM	Length of learned observation sequences					
	2	3	4	5	6	7	8
2	86.2%	84.0%	83.5%	83.5%	83.3%	83.6%	83.6%
3	86.2%	84.6%	84.2%	84.2%	83.7%	83.8%	83.8%
4	87.0%	85.4%	85.8%	86.2%	86.6%	85.3%	83.8%
5	**88.1%**	88.1%	85.8%	85.9%	85.9%	85.7%	86.1%
6	85.4%	85.1%	85.0%	85.0%	85.2%	82.7%	82.6%
7	81.8%	82.2%	81.3%	81.5%	82.2%	82.3%	82.3%
8	77.4%	78.7%	78.2%	78.3%	78.2%	78.6%	78.8%

Table 1: The precisions of Goal-based Group Behavior Assignment on multiple scenarios (i.e. Figure 4) using CEP and HMM (according to the observation sequences lengths used to learn and recognize them).

sequence length of 8 as we found that for higher length the precision decreases. In order to obtain the best detection quality, the last five observations (one per second) in the tested sequences (of these scenarios) always give the highest precision. The decomposition of sequences of *GbG* behaviors to learn them with only two observations give a better precision. The solutions with a short length of observation sequences (as 2,2 in the table 1) detects more quickly each new *GbG* behavior but each new noisy observation has more impact (e.g., when the group avoids an obstacle during their observed movements). The solution with a long length of observation sequences (as 5,5) detects the current *GbG* behaviors even with more noisy observations but each new *GbG* behavior takes longer to be detected (after two new observations). The majority of errors during the detection occurs when a group changes its current *GbG* behavior. In order to increase the precision more than 92%, we added each possible *GbG* behavior (*GbGB*) that overlaps the evolution of an old *GbGB* toward a new *GbGB*. But by increasing the size of the library of *GbGB* models, we add a difficulty for an human user to understand what is happening.

CEP updates instantaneously the current state of the *GbG* behavior but without taking account the uncertainty of observations. In order to help in a crisis management situation, the poor precision (58%) indicates that the behaviors of groups and their goals can not be detected by *CEP*. On the opposite, **the *Goal-based Group Behavior Assignment* reduces the impact of observations under uncertainty and always detects the nearest *goal-based group behavior* after two observations**.

As a conclusion the precision of our proposed method can be used to help in crisis management context, in particular with the *valued goal-based group behavior detection* used as demonstrated in the following experiment.



(a) People that realised the presence of the suicide bomber can be seen running and fleeing the scene. The people that did not see the bomber can be seen continuing their movement or looking surprised at the panicked people.

(b) A 2D view of the situation. The marks denote the different group detected and their direction of movement.

(c) It is easier to follow each centroid movement of groups. The *valued goal-based group detection* reveals that there are groups that avoid and move away from a single agent during its movement toward the player.

Figure 5: A situation with a high-level risk.

5.3 Threat detection on complex dynamic environments

In the framework of the *EUSAS project* in *VBS2*, we simulated a busy urban street. Two hundred people can be seen walking peacefully in a road until they realise that there is a suicide bomber among them. They can see a man wearing a concealed bomb under his shirt. Then, the people in crowd start to flee in panic. However, there are people that do not see the man, and without realising the imminent danger they continue moving towards him. At the scene there is also a reporter that observes the situation and tries to understand what is happening. A screenshot of the simulated street is shown in Figure 5(a).

The proposed methods are applied to the simulated scenario. Initially, the crowd was separated into groups according to their observed state of the group. Thus we applied the *valued goal-based group detection* (i.e. section 4.3) in order to detect the intention $\phi^{\Leftarrow g^j \rightrightarrows}$ of groups run away from a specific entity (that is a non-detectable threat) within the crowd and the intention $\phi^{c^i \longmapsto g^j}$ of a group than continue its movement along a long-term predefined goal. The result of the group detection is shown in Figure 5(b).

In order to identify $\phi^{\Leftarrow g^j \rightrightarrows}$, we verify for each goal, if its *goal value* $V_{t-T,t}^{\phi^{\Leftarrow g^j \rightrightarrows}, C^i}$ is not abnormaly high using a distribution that gives a positive value when groups move around or away from the goal, else it gives a negative value: $V(\{\lambda^{C^i \leftarrow g^j}, \lambda^{C^i \circlearrowleft g^j}, \lambda^{C^i \leftrightsquigarrow g^j}, \lambda^{C^i \nleftrightarrow g^j}, \lambda^{C^i \rightarrow g^j}\}) = \{1, \frac{1}{2}, 0, -\frac{1}{2}, -1\}$. And $\phi^{c^i \longmapsto g^j}$ the long-term goal g^j of the group c^i is identified by the highest *goal value* $V_{t-T,t}^{\phi^{c^i \longmapsto g^j}}$ using the distribution: $V(\{\lambda^{c^i \rightarrow g^j}, \lambda^{c^i \circlearrowleft g^j}, \lambda^{c^i \leftrightsquigarrow g^j}, \lambda^{c^i \nleftrightarrow g^j}, \lambda^{c^i \leftarrow g^j}\}) = \{1, \frac{1}{2}, 0, -\frac{1}{2}, -1\}$.

With these distributions, each *non-searched goal value* is near to zero over time, and the *searched goal value* is quickly higher than the T value (because of is computed from $t-T$ to T) and increased by the number of goals or groups (when the *goal value* is the sum of their *goal values*, i.e. Equation 6).

Figure 5(a) shows the panicked groups running away for the threat. Note that not everyone realised the threat and some people continue along their original path. Figure 5(c) shows the results of the detections. It presents an overview of the situation for the analyst. First the threat is detected based on the behavior of the people towards this entity. Also the groups that did not detect the threat and are in danger can also be detected. This enables the security forces to predict the groups in danger and from the group members predict the number of casualties.

Another important aspect of the method is that if the suicide bomber tries to hide within the group the system treats the entire group as dangerous, rather than allowing the threat to be hidden within the majority of an innocent group. Therefore the system can continue tracking the danger even if he tries to conceal himself by merging with another group. However, a difficulty that the method faces is to create a correct distribution of *goals values* according to the library of *GbG behaviors* because each abnormal intention executes specific behaviors that must be valued correctly in order to avoid fake detections. This provides a powerful tool to the security forces to understand the situation and predict threat in order to take better informed decisions on the developing situation.

This could involve resource allocation to investigate further the detected threat, or eliminating the threat once it is confirmed. Furthermore, it could provide information about the member of public in greater danger so that the security forces can focus on accessing them and protecting them. Finally, in case there are critical infrastructures and vulnerable entities, the system can be used to assess the intention of the target towards these assets, based on whether he is moving towards them or not.

6. CONCLUSION

We have shown that the use of *discrete relative* states in *Hidden Markov Model* may successfully detect, in real-time both multiple *collective behaviors and goals of groups* based on observations. We have compared the proposed approach with a different method called *Complex Event Processing*, and found the results are superior in terms of precision and tolerance to noises.

In future work on asymmetric threat environments, we plan to apply our approach on new abnormal collective behaviors such as panic and violent crowd behavior by the fusion of new types of information. The automatic identification and learning of new collective behaviors including agent roles is envisioned to obtain a fully automatic collective behavior detection system. This last objective could allow an analyst to make an informed decision in the case of an asymmetric threat, taking into account the detected results.

7. REFERENCES

[1] B. Banerjee, L. Kraemer, and J. Lyle. Multi-agent plan recognition: Formalization and algorithms. In *Proceedings of AAAI*, volume 10, pages 1059–1064, 2010.

[2] C. Bowman. The dual node network (dnn) data fusion & resource management (df&rm) architecture. In *AIAA Intelligent Systems Conference, Chicago*, 2004.

[3] H. Bui. Efficient approximate inference for online probabilistic plan recognition. In *AAAI Fall Symposium on Intent Inference for Users, Teams and Adversaries*, 2002.

[4] H. Bui, D. Phung, and S. Venkatesh. Hierarchical hidden markov models with general state hierarchy. In *Proceedings Of The National Conference On Artificial Iintelligence*, pages 324–329. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2004.

[5] H. Bui, S. Venkatesh, and G. West. Policy recognition in the abstract hidden markov model. *Journal of Artificial Intelligence Research*, 17(1):451–499, 2002.

[6] M. Chang, N. Krahnstoever, and W. Ge. Probabilistic group-level motion analysis and scenario recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 747–754. IEEE, 2011.

[7] P. Charniak Robert et al. A bayesian model of plan recognition. *Artificial Intelligence*, 64(1):53–79, 1993.

[8] J. Choi and K. Kim. Inverse reinforcement learning in partially observable environments. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence (IJCAI)*, pages 1028–1033, 2009.

[9] T. Duong, H. Bui, D. Phung, and S. Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 838–845. IEEE, 2005.

[10] K. Gaitanis, M. Gemo, O. Vybornova, D. Ruiz, R. Moncarey, and B. Macq. Cooperative team behaviour recognition for multimodal fusion. 2007.

[11] M. Giersich, P. Forbrig, G. Fuchs, T. Kirste, D. Reichart, and H. Schumann. Towards an integrated approach for task modeling and human behavior recognition. *Human-Computer Interaction. Interaction Design and Usability*, pages 1109–1118, 2007.

[12] K. Han and M. Veloso. Automated robot behavior recognition. In *Robotics Research-International Symposium-*, volume 9, pages 249–256. Citeseer, 2000.

[13] L. Liao, D. Fox, and H. Kautz. Hierarchical conditional random fields for gps-based activity recognition. *Robotics Research*, pages 487–506, 2007.

[14] X. Liu and C. Chua. Multi-agent activity recognition using observation decomposed hidden markov model. *Computer Vision Systems*, pages 247–256, 2003.

[15] D. Luckham. *The power of events: an introduction to complex event processing in distributed enterprise systems*. Addison-Wesley Longman Publishing Co., Inc., 2001.

[16] L. Luotsinen and L. Boloni. Role-based teamwork activity recognition in observations of embodied agent actions. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*, pages 567–574. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[17] N. Nguyen and S. Venkatesh. Discovery of activity structures using the hierarchical hidden markov model. In *British Machine Vision Conference*, pages 409–418, 2005.

[18] N. Nguyen, S. Venkatesh, and H. Bui. Recognising behaviours of multiple people with hierarchical probabilistic model and statistical data association. In *British Machine Vision Conference*, 2005.

[19] N. Nguyen, S. Venkatesh, G. West, and H. Bui. Learning people movement model from multiple cameras for behaviour recognition. *Structural, Syntactic, and Statistical Pattern Recognition*, pages 315–324, 2004.

[20] N. Oliver, E. Horvitz, and A. Garg. Layered representations for human activity recognition. In *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*, pages 3–8. IEEE, 2002.

[21] J. Patrix, A. Mouaddib, and S. Gatepaille. Detection of primitive collective behaviours in a crowd panic simulation based on multi-agent approach. *International Journal of Swarm Intelligence Research (IJSIR)*, 3(3):50–65, 2012.

[22] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[23] L. Rabiner and B. Juang. An introduction to hidden markov models. *ASSP Magazine, IEEE*, 3(1):4–16, 1986.

[24] S. Saria and S. Mahadevan. Probabilistic plan recognition in multiagent systems. In *Proceedings of International Conference on AI and Planning Systems*, 2004.

[25] Y. Shi, Y. Huang, D. Minnen, A. Bobick, and I. Essa. Propagation networks for recognition of partially ordered sequential action. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–862. IEEE, 2004.

[26] G. Sukthankar. *Activity recognition for agent teams*. PhD thesis, Citeseer, 2007.

[27] G. Sukthankar and K. Sycara. Automatic recognition of human team behaviors. In *Proceedings of Modeling Others from Observations, Workshop at the International Joint Conference on Artificial Intelligence (IJCAI)*, 2005.

[28] G. Sukthankar and K. Sycara. Simultaneous team assignment and behavior recognition from spatio-temporal agent traces. In *Proceedings of the National Conference on Artificial Intelligence*, volume 21, page 716. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[29] D. Tran, D. Phung, H. Bui, and S. Venkatesh. Factored state-abstract hidden markov models for activity recognition using pervasive multi-modal sensors. In *Intelligent Sensors, Sensor Networks and Information Processing Conference, 2005. Proceedings of the 2005 International Conference on*, pages 331–336. IEEE, 2005.

[30] D. Vail, M. Veloso, and J. Lafferty. Conditional random fields for activity recognition. In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pages 1–8. ACM, 2007.