Discrete Relative States to Learn and Recognize Goals-based Behaviours of Group
Jérémy Patrix, Abdel-Illah Mouaddib, Simon Le Gloannec, Dafni Stampouli, Marc Contat

To cite this version:
Jérémy Patrix, Abdel-Illah Mouaddib, Simon Le Gloannec, Dafni Stampouli, Marc Contat. Discrete Relative States to Learn and Recognize Goals-based Behaviours of Group. AAMAS 2013, 2013, Saint Paul, Minnesota, United States. hal-00971734

HAL Id: hal-00971734
https://hal.science/hal-00971734
Submitted on 3 Apr 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Discrete Relative States to Learn and Recognize Goals-based Behaviors of Groups

Jérémy Patrix,* Abdel-Illah Mouaddib† Simon Le Gloannec‡
GREYC (UMPR 6072) GREYC (UMPR 6072) IPCC department,
University of Caen, France University of Caen Val-de-Reuil, France
jeremy.patrix@unicaen.fr abdel-illah.mouaddib@unicaen.fr
& Dafni Stampouli & Marc Contat
IPCC department, CASSIDIAN
CASSIDIAN, France IPCC department,
CASSIDIAN, France CASSIDIAN
jeremy.patrix@cassidian.com
{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
1.2 [Artificial Intelligence]: Distributed Artificial Intelligence; 1.5 [Pattern Recognition]: Models, Design Methodology; 1.6 [Simulation and Modeling]

General Terms
Algorithms, Measurement, Security, Human Factors

*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.

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
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 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 $a_{ij} = Pr(r(s_{t+1} = q_j | s_t = q_i)$ 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 $q_i$;
- $\Pi$ 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

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

and allows three main computations [23]:

1. The probability of the current state by the equation $Pr(s_t = q_i | O_t)$ that gives the most likely hidden state $q_i$ 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.

$$p_{x,y,t}^c = \frac{\sum_{i=1}^{|c|} p_{x,y,t}^{c,i}}{|c|}$$

At time $t$, the group position $p_{x,y,t}^c$ of the centroid state $s_t^i$ is computed by the average of positions $p_{x,y,t}^{c,i}$ of each agent $q_i$ of the group $c$. The centroid state $s_t^i$ 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 leaves 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 Hierarchical 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-JPDH [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, Multi-agent-AHMEM [10] allows a contextual, non-deterministic approach and a high level multimodal 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 HMMs 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 (CEs), (3) the aggregation rules for new CE 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 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 $\{o_t^{a'}, o_t^{a''}\}$ on the hidden states $\{s_t^{a'}, s_t^{a''}\}$ for each agent $a'$ 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 $(p_t, v_t)$.

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

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

$$\Delta s_t^c = \sqrt{[r_{x,t} - r_{x,t-1}]^2 + [r_{y,t} - r_{y,t-1}]^2}$$

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

3. We discretize the relative states $\{\Delta s_t^{c-1}, \Delta s_t^c\}$ on their difference from $t - 1$ and $t$ for each variable $\Delta r_t^c \in \Delta r_t^{c-1}$. The discrete relative variables $\{d_r, d_r, d_r, d_r\}$ are introduced to represent any DR state which are, in our case, the DR positions $\{p_{x,y,t}, p_{x,y,t}, p_{x,y,t}, p_{x,y,t}\}$ and the DR speeds $\{v_{x,y,t}, v_{x,y,t}, v_{x,y,t}, v_{x,y,t}\}$. These DR variables are given by:

$$d_{r} = \begin{cases}\text{if } \Delta r_t^c < \delta_{\text{min}} & \text{drd (decrease)} \\ \text{if } \Delta r_t^c > \delta_{\text{min}} + \delta_{\text{max}} & \text{drd (increase)} \\ \text{if } |\Delta r_t^c - \delta_{\text{min}}| < \delta_{\text{min}} & \text{drd (constant)} \end{cases}$$

The value $\delta_{\text{min}}$ indicates a minimal state transition between two successive times according to the variable $r \in \{x, y\}$, 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:

$$Q^{DR} = \{ (p_{x,y}, v_{x,y}) \}$$

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 $(p_{x,y}, v_{x,y})$ represents a null discrete relative position $p_{x,y}$ and a null discrete relative speed $v_{x,y}$ (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:

$$\Delta S_{0}^{c,g} = \{ (p_{x,y}, v_{x,y}) \}$$

$$\Delta S_{1}^{c,g} = \{ (p_{x,y}, v_{x,y}) \}$$

First during $\Delta S_{0}^{c,g}$, the group quitted the goal $g$ and has moved away from it, second during $\Delta S_{1}^{c,g}$, $c$ has moved around the goal at the same distance, and last during $\Delta S_{1}^{c,g}$, $c$ has moved towards $g$ and is now close to it.

4.2 Goal-based Group Behavior Assignment

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

$$O_t^{G} = \{ \{o_0^{a'}, o_0^{a''}, \ldots, o_T^{a'}, o_T^{a''}\} : \{o_0^{a'}, \ldots, o_T^{a''}\} \}$$

is an observation sequence of each agent of the group $c$ and each goal $G$ from $0$ to $t$, where $\{o_0^{a'}, o_T^{a''}\}$ is an observation of the $i^{th}$ agent and the $j^{th}$ possible goal that an observed group $c$ can seek to reach.

$\Lambda$ is a set of HMMs, 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 $\Delta S_{t-1}^{c}$ from an observation sequence according to HMM $\lambda^h$.

$$Pr(\lambda | O_t)$$

is the probability that the behavior $\lambda$ 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 $O_{t-T}^{g^j}$ in DR states sequence $\Delta S_{t-1}^{c}$ is given by:

$$M_t^c(\lambda^h, g^j) = Pr(\lambda | \Delta S_{t-1}^{c,g^j} = f^h(O_{t-T}^{g^j}))$$

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

$$M_t^c = \begin{bmatrix}
(M(\lambda^h, g^j) & \cdots & M(\lambda^h, g^j) & \cdots & M(\lambda^h, g^j) \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
M(\lambda^h, g^j) & \cdots & M(\lambda^h, g^j) & \cdots & M(\lambda^h, g^j) \\
M(\lambda^h, g^j) & \cdots & M(\lambda^h, g^j) & \cdots & M(\lambda^h, g^j)
\end{bmatrix}$$

We obtain the most likely tuple $\lambda^h, g^j$ by:

$$M(c, t) = \text{argmax}_{\lambda^h, g^j} \{ M_t^c(\lambda^h, g^j) \}$$

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 $\lambda^H$ for each goal $\forall g^j \in G$ by the result of:

$$M(c, g^j, t) = \arg\max_{\forall \lambda^h \in \Lambda} \{ M^j_t (\lambda^h; g^j) \}$$  \hspace{1cm} (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 $\lambda^i$ on $M^i$.

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:

- $V_{t-T, t}^{c, g^j}$ the intention of the group $c^i$ according to the goal $g^j$;
- $V_{t-T, t}^{G^j}$ 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 $\lambda^h$ of a group $c$ give the value $V(c, g, \lambda^h)$. A distribution is created according to the intention $\phi^p (\in \Phi$ 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:

$$V_{t-T, t}^{\phi^p, c^i, g^j} = \sum_{t-e=T}^{t} V_{t-T, t}^{\phi^p, c^i, g^j, \lambda^h} = M(c^i, g^j, t_e)$$  \hspace{1cm} (6)

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

For example, in order to detect the civilians with a dangerous intention $\phi^p$ 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 $\phi^p$ will have a higher positive goal value. According to the distribution of the intention $\phi^p \in \Phi$, the group $c$ is moving towards the specific long-term goal $g^p$ as:

$$g^p = \arg\max_{\forall c^i \in G^j} V_{t-T, t}^{\phi^p, c^i, g^j}$$  \hspace{1cm} (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 experiment 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 HMMs and CEPs. For the HMMs, it uses a jaHMM \(^2\) library available on-line (managing states, observations and standard algorithms relating to HMMs). 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: $\lambda^{go-g}$ moving towards a goal, $\lambda^{go-g}$ moving away from a goal, $\lambda^{go-g}$ moving around a goal and $\lambda^{go-g}$ 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 $\Pi$ (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).

![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).](http://code.google.com/p/jahmm/)

---

\(^1\)Bohemia Interactive Simulations, Australia, 2011, http://vbs2.com

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 $(p_{\text{in}}, t_{\text{in}})$ 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 $M_i^j$ (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 ($\lambda^{\text{ag}}$, $\lambda^{\text{cg}}$, $\lambda^{\text{bg}}$, $\lambda^{\text{gb}}$), we added an inconsistent GbG behavior $\lambda^{\text{cggb}}$ that is a HMM by default without learning. If $\lambda^{\text{cggb}}$ 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 $\infty$ (by an iterated method). The table presents results up to a 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, 3 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 4, 5, 6, 7, 8) 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 (GGB) that overlaps the evolution of an old GGB toward a new GGB. But by increasing the size of the library of GGB 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.

![Figure 4: Examples of monitored experimentations.](image)

<table>
<thead>
<tr>
<th>1</th>
<th>CEP</th>
<th>58.06 %</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>HMM</td>
<td>Length of learned observation sequences</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>86.2%</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>84.0%</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>83.5%</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>83.5%</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>83.5%</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>83.5%</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>83.5%</td>
</tr>
</tbody>
</table>

Table 1: The precision 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).
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 \( g_{i=1}^{g_{j}} \) of groups run away from a specific entity (that is a non-detectable threat) within the crowd and the intention \( g_{j=1}^{g_{j}} \) 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 \( g_{i=1}^{g_{j}} \), we verify for each goal, if its goal value \( V_{g_{i}^{g_{j}}}^{t} \) is not abnormally high using a distribution that gives a positive value when groups move around or away from the goal, else it gives a negative value: \( (\{X^{c_{1}^{g_{j}}} , X^{c_{2}^{g_{j}}} , X^{c_{3}^{g_{j}}}, X^{c_{4}^{g_{j}}} \}) = \{1 , \frac{1}{2}, 0, -\frac{1}{2}, -1 \} \). And \( g_{j=1}^{g_{j}} \) the long-term goal value \( g_{j=1}^{g_{j}} \) of the group \( c_{j} \) is identified by the highest goal value \( V_{g_{j}^{g_{j}}}^{t} \) using the distribution: \( (\{X^{c_{1}^{g_{j}}} , X^{c_{2}^{g_{j}}} , X^{c_{3}^{g_{j}}}, X^{c_{4}^{g_{j}}} \}) = \{1 , \frac{1}{2}, 0, -\frac{1}{2}, -1 \} \).

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 number 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 collec-
tive 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