3

Tracking behaviours of cooperative robots within multi-agent domains

Fernando Ramos[1] and Huberto Ayanegui[2]
[1]*Tecnologico de Monterrey Campus Cuernavaca*
[2]*Universidad Autónoma de Tlaxcala*
Mexico

1. Introduction

The most important concerns in multi-agent cooperative systems are focused on the construction of models related with the communication, the interaction and the behavior of agents participating in a task. This chapter deals with behavioral aspects of cooperative agents while they are evolving in a task. Behavioral aspects of cooperative agents may give us precious information about individual, relational and functional roles that the agents assume during the different steps of a task. For instance, in competitive domains, such as robotic soccer, teams of agents dispute common resources to reach a goal. The importance of knowing about behavioral aspects of a team of agents under observation could give us valuable information to generate counter strategies or tactics to obtain the resources being disputed.

The behavioral aspects concern strategic and tactical behaviors. The former implies long term actions where the whole team is involved, while the latter is related with short term actions where two or more agents are involved. It is important to point out that tactical behaviors should be submitted to strategic behaviors.

The domains of cooperative agents are commonly complex due to the dynamic conditions and the multiple interactions between agents. Based on the precedent statements, a particular interest in this chapter is focused on the analysis of problems that can difficult the construction of models of behaviors.

Relevant works related with the study of behaviors in multi-agent domains, applied to soccer robotics, are exposed. We aim to illustrate the problematic, the advantages and drawbacks of different approaches that have been proposed to model the behavior of soccer-agents while they are evolving in a task.

Finally, we expose a model able to discover behaviors and tracking patterns in the soccer domains, which was tested in real games extracted from several matches belonging to different Robot-Cup Tournaments. The results obtained by applying this model have shown that it is able to discover satisfactory behaviors of strategic and tactical patterns as well as tracking the behaviors while the robots are evolving in a competitive complex task.

2. Cooperative Agents

Cooperative agents are focused on how a loosely-coupled network of problem solvers can work together to solve problems that are beyond their individual capabilities. Each problem-solving node in the network is capable of sophisticated problem-solving and can work independently, but the problems faced by the nodes cannot be completed without cooperation. Cooperation is necessary because no single node has sufficient expertise, resources, and information to solve a problem, and different nodes might have expertise for solving different parts of the problem (Durfee et al., 1989b).

Multi-agent systems research is concerned with the wider problems of designing societies of autonomous agents, such as why and how agents cooperate (Wooldridge & Jennings, 1994); how agents can recognize and resolve conflicts (Adler et al., 1989; Galliers, 1988b; Galliers, 1990; Klein & Baskin, 1991; Lander et al, 1991); how agents can negotiate or compromise in situations where they are apparently at loggerheads (Ephrati & Rosenschein, 1993; Rosenschein and Zlotkin, 1994); and so on.

An important concern is to design an appropriate organization of a multi-agent system for a particular domain and environment, such as described in the survey by (Horling & Lesser, 2004). In this work, advantages and disadvantages of these organizations are discussed. Such organizations can be hierarchies, holarchies, coalitions, teams, congregations, societies, federations, markets, and matrix organizations.

2.1 Coordination of multi-agent systems

Coordination of tasks is an important aspect directly associated with the success of a plan that supports a strategy and/or tactic. In addition, the coordination plays an important role in the correct execution of cooperative actions, in such a way that conflictive situations could be avoided. In (Chernova & Veloso, 2008) a teacher instructs multiple robots to work together in coordination through a demonstration of the desired behavior, using a communication system and a sharing information system. Another distributed approach that share information, which help to facilitate the coordination and solve problems such as collisions is described in (Jansen & Sturtevant, 2008). One of the most important complex dynamic domains of application of multi-agent systems is the soccer robot systems. In (Candea et al., 2001), aspects of coordination of soccer robots within the framework of RoboCup are treated, doing emphasis in behavior based techniques. The communication and distributed coordination is addressed in this work.

Multi-agent teamwork is critical in a large number of agent applications, including training, education, virtual enterprises and collective robotics (Nair et al., 2004). However, in multi-agent domains, agent interactions become the domain highly complex for the analysis of agent-team behaviors, such is the case of robotic soccer. We consider “complex domains“ to be those with enormous state action spaces, dynamic environment, competitive and real time. Obviously, when the multiple interactions of both teams are considered the task of analysis for modeling behaviors is even more complex.

2.2 Behaviors in multi-agent systems

Machine Learning techniques are used to model the robot behaviors from training instances generated during a play by imitating the human behavior derived from the interaction of a human that control a robot soccer agent during a play (Aler et al., 2009). Low-level behaviors take place: looking for the ball, conducting the ball towards the goal or scoring in the presence of opponent players. Lin et al. (2009), uses color features and hybrid systems compose of multi-layer perceptrons and genetic algorithms for the recognition of human behaviors based on trajectory patterns. Social insects provide a rich source of traceable social behavior for testing multi-agent tracking, prediction and modeling algorithms (Balch et al., 2001).

The use of predictive systems for studying agent opponent behaviors driven by the recognition of team actions, such as usual paths performed by an agent, plays performed by two agents or preferences from bid exchanges, can serve to build behavior patterns that can serve as guide to recognize behaviors of several agents participating in a cooperative task. An approach based on HMM serves to determine spatio-temporal behavior agent patterns through the recognition of team actions (Luotsinen & Bölöni, 2008). Meanwhile, Bayesian Networks can be used as learning system to characterize behaviors of the opponent. In particular in (Hindriks & Tykhonov, 2008), Bayesian Networks are used to learn opponent preferences from bid exchanges by making some assumptions about the preference structure and rationality of the bidding process. Based on the observations of agent behaviors the recognition of tactical enemy plans is made in military applications (Mulder & Voorbraak, 2003; Henniger & Madhavan, 2004) compared the performance of three predictive models all developed for the same, well-defined modeling task. Specifically, this paper compares the performance of an Extended Kalman Filter (EKF) based model, a neural network based model and a Newtonian-based dead-reckoning model, all used to predict an agent's trajectory and position.

2.3 Tracking behaviors

Multi-Agent systems, such as soccer robots, change constantly or apparently change due to the dynamic conditions and multiple interactions between agents. Due to this problems, the tracking of agents becomes very difficult and, by the way, the behaviors assumed by the agents risk of being quite different even in similar environment conditions. Tracking behaviors of multi-agent systems is very important in the study of opponent team attitudes in the design of counter strategies in soccer-agents worlds.

Tambe and Rosenbloom enhance the importance of agent tracking (Tambe & Rosenbloom, 1996). They argue that agent tracking is a key capability required for interactions in multi-agent environments. It involves monitoring other agents' observable behaviors and inferring their unobserved behaviors or high-level goals and plans. Their work examines the implications of such an agent tracking capability for agent architectures. It specifically focuses on real-time and dynamic environments, where an intelligent agent is faced with the challenge of tracking the highly flexible mix of goal-driven and reactive behaviors of other agents, in real-time. This support takes the form of an architectural capability to execute the other agent's models, enabling mental simulation of their behaviors. They have implemented an agent architecture, an experimental variant of the Soar integrated architecture, that conforms to all of these requirements. Agents based on this architecture have been implemented to execute two different tasks in a real-time, dynamic, multi-agent domain.

They propose some of the key requirements for agent tracking in real-time, dynamic environments. This analysis is based on tasks in a real-world, multi-agent environment and assumes that an agent is situated in the environment, as it tracks other agents while simultaneously interacting with them. Key requirements revealed by this analysis include:

1. Tracking other agents' highly flexible mix of goal-driven and reactive behaviors.
2. Recursively tracking its own actions from the perspective of other agents, so as to understand their impact on the other agents' behaviors.
3. Tracking groups of other agents, possibly acting in coordination.
4. Simultaneously tracking and reacting to other agents' actions.
5. Tracking other agents' activities in real-time, while resolving ambiguities.

For an illustrative example of agent tracking in pilot agents for a combat simulation environment, consider first the air-to-air combat scenario in Figure 1, involving fighter jets. The pilot agent **L** in the light-shaded aircraft is engaged in a combat with pilot agents **D** and **E** in the dark-shaded aircraft. Since the aircraft are far apart, **L** can only see its opponents' actions on radar (and vice versa). In Figure 1-a, **L** observes its opponents turning their aircraft in a coordinated fashion to a collision course heading, i.e., with this heading, they will collide with **L** at the point shown by **x**. Since the collision course maneuver is often used to approach one's opponent, L infers that its opponents are aware of its (**L**'s) presence, and are trying to get closer. Given a highly hostile environment, **L** may also infer that opponents are closing into fire their missiles. However, **L** has a missile with a longer range, so L reaches first its missile range. **L** then turns its aircraft to point straight at **D**'s aircraft and fires a radar-guided missile at **D** (Figure 1-b). Subsequently, **L** executes a 35^o *fpole* turn away from **D**'s aircraft (Figure 1-c), to provide radar guidance to its missile, while slowing its rate of approach to the enemy aircraft.

While neither **D** nor **E** can observe this missile on their radar, they do observe **L**'s pointing turn followed by its fpole turn. They track these to be part of L's missile firing behavior, and infer a missile firing. Therefore, they attempt to evade his missile by executing a 90^o beam turn (Figure 1-d). This causes their aircraft to become invisible to **L**'s radar. Deprived of radar guidance, **L**'s missile is rendered harmless. Meanwhile, in Figure 1-d, **L** tracks its opponents' coordinated beam turn (even while not seeing the complete turn). **L** then prepares counter-measures in anticipation of the likely loss of both its missile and radar contact.

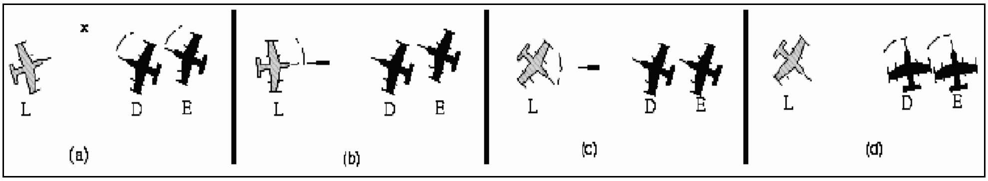


Fig. 1. Pilot agents **D** and **E** are engaged in combat with **L**. An arc on an aircraft's nose shows its turn direction.

Finally, Tambe and Rosenbloom argue that if agents are to successfully inhabit complex, dynamic social worlds, they must obtain architectural support for agent tracking –an important capability required for agent interactions. Their approach has been used in a large-scale operational military exercise (Tambe et al., 1995).

The use of relevant parameters helps to increase the robustness of tracking by using also predictive models. Such is the case of (Muñoz, 2008; Muñoz et al., 2008; Muñoz et al., 2009), where color information associated with clothes of people, and predictive models based on Kalman Filter and Bayesian Networks, has helped to reduce the errors of the tracking system.

The use of classification algorithms, such as K-means, serves to categorize animal tracking data into various classes of behaviors even in the absence of biological factors that should be considered (Schwager et al., 2007). An automatic video tracking system is used to study behaviors, movements and interactions, between insects such as beetles, fruit flies, soil insects, ticks and spiders (Noldus et al., 2002).

Ukita and Matsuyama propose a real time cooperative multi-target tracking based on Active Vision Agents that interact dynamically between them. An architecture compose of three layers that use parallel processes through which the information is exchanged for an effective cooperation (Ukita & Matsuyama, 2005).

Most of the behaviors in soccer agent systems depend on the strategies to be performed. Strategies conditioned also tactical and individual plays. Associated with the strategies specific structures, formations of players, determine importantly the correct execution of strategies and tactics.

2.4 Formations

Formation for multi-agent systems with large population of members is the main concern of the work described in (Xiao et al., 2009), where formation information is divided into two parts: some agents are responsible of global formation information to carry out the navigation of the whole team. Meanwhile, the other agents regulate their positions delaing with local information in distributed manner. In (Porfiri et al., 2007) is described a tracking and formation control for an agent team within a dynamic environment by using shared information to control their trajectories in a cooperative manner.

Most of the research involved in multi-agent modeling is based on building models considering partial aspects and non relevant aspects of the team. Nevertheless, relevant aspects associated with any team should be taken into account in order to model its behaviors. These aspects include: individual actions (individual aspect), relationships between agents (tactical aspect) and formation behaviors (strategy aspect). This chapter emphasizes on the fact of having an expressive representation model which takes into account different aspects exhibited in a team of agents. The adequate representation of these aspects enables the discovery of behavior patterns at different levels of abstraction in a complex domain. Thus, we argue that an expressive representation model at different levels of abstraction facilitates the discovery of behavior patterns in a complex domain such as robotic soccer. Some of the most important behaviors are related with strategic and tactical plays (Ramos & Ayanegui, 2008a) . On the one hand, a team that presumes to play by following certain strategies should play under the context of formations to assure order, discipline and organization during a match (Kuhlmann et al., 2005). On the other hand, tactical plays occur, most of the time, under the context of formations. The discovery of tactical or team behaviors needs the tracking of both the positions of players at any instant of the game and relevant relations able to represent particular interactions between players. Nevertheless, the tracking task becomes very complex because the dynamic conditions of the game brings about drastic changes of positions and interactions between players, which difficult the construction of models capable of recognizing and discovering behaviors of teams playing soccer matches (Lattner et al., 2005). In (Ramos & Ayanegui, 2008b), we proposed a model able to manage the constant changes occurring in the game, which consists in building topological structures based on triangular planar graphs. Thus, based on this model tactical behavior patterns have been discovered and tracked in spite of the dynamic conditions. The test domain for this research was simulated robotic soccer, specifically, the Soccer Server System (Noda & Frank, 1998), used in the Robot World Cup Initiative (Kitano et al., 1997), an international AI and robotics research initiative. A total of 10 matches was analyzed. The results obtained have been shown that the model has been able of recognizing and discovering behaviors satisfactory.

3. Related works with soccer agents

Raines et al. (2000) developed a system called ISAAC, a tool that helps humans to analyze, evaluate and understand agent and multi-agent behavior. ISAAC analyzes soccer games off-line after its end using data from the agents observable behavior traces. An impressive wide range of behaviors of the individual agent, of agent interactions and of team success or failure are analyzed (see Figure 2)

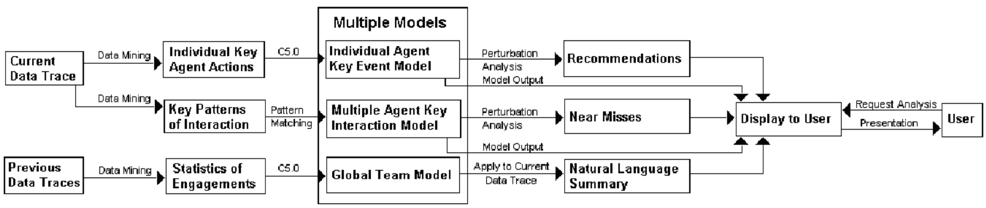


Fig. 2 Flow chart for ISAAC model generation and analysis

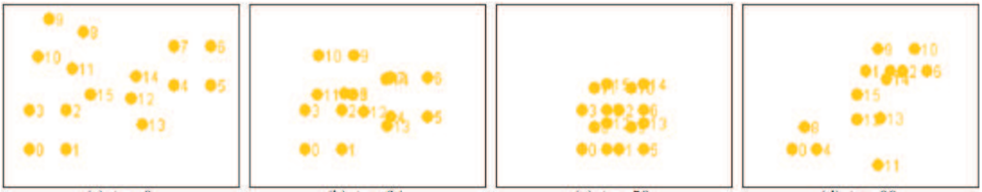
Data traces are matched against generic interaction pattern only tofigure out the success or failure of the interaction behavior. This information is statistically processed and presented to the human observer. Authors propose an analysis of the events leading up to key events, such as shots on goal in the case of the RoboCup soccer simulation. ISAAC analyses the situations when the defence of the goal succeeds or fails with respect to a number of variables, such as the distance of the closest defender, the angle of the closest defender with respect to the goal, and the angle of the attacker from the centre of the field, the angle of the shot on goal and the force of the kick. The user is able to do a perturbation analysis to determine which changes in a rule will increase the goal success rate (e.g. changing the angle at goal, increasing the force of the kick). This enables analysing teams to seek improvements. ISAAC produces learned rules to explain the performance of the team as well as predict future outcomes. However, there is no automated way for these learned rules to be used by the agent team; rather, the team designer analyzes the rules and decides how to modify the team.

Without a priori knowledge of current team assignments, the behavior recognition problem is challenging since behaviors are characterized by the aggregate motion of the entire team and cannot generally be determined by observing the movements of a single agent in isolation. To handle this problem, Sukthankar and Sycara (2006) introduce the algorithm STABR (Simultaneous Team Assignment and Behavior Recognition), that generates behavior annotations from spatio-temporal agent traces. STABR completely annotates agent traces with (1) the correct sequence of low-level actions performed by each agent and (2) an assignment of agents to teams over time. Such algorithm employs a randomized search strategy (Fischler & Bolles, 1981) to identify candidate team assignments at selected time steps; these hypotheses are evaluated using dynamic programming to derive a parsimonious explanation for the entire observed spatio-temporal sequence. To prune the number of hypotheses, potential team assignments are fitted to a parameterized team behavior model; poorly-fitting hypotheses are eliminated before the dynamic programming phase. The proposed approach is able to perform accurate team behavior recognition without exhaustive search over the partition set of potential team assignments, as demonstrated on several scenarios of simulated military maneuvers.

3.1 Problem Formulation

The formulation of the problem is: Let $\mathbf{A} = \{ a_0, a_1, \ldots, a_{N-1} \}$ be the set of agents in the scenario. A team consists of a subset of agents, and we require that an agent only participate in one team at any given time; thus a team assignment is a set partition on **A**. An agent that is not currently a member of any team is known as a singleton, and is unrestricted in its motion. By contrast, the agents in a team are constrained to move according to a set of team behaviors, **B**. The subset of behaviors available to a given team is specified by the domain and can depend on the number of agents in the formation and their relative configurations. For instance, the domain could specify that four agents in a square formation may execute a "wheel" (formation advances in an arc by rotating about a corner), but not a "pivot" (formation rotates about its center), which may be restricted to teams of three agents. In the course of a scenario, agents (either singletons or subsets of disbanding teams) can assemble into new teams; similarly, teams can disband to enable their members to form new teams or to operate as singletons. Thus the team assignment is expected to change over time during the course of a scenario. The team assignments over time and the behavior executed by each team are hidden. Assume that the input consists only of a *spatio-temporal traces*, which is a sequence of noisy observations of the 2D position of each agent through time, $a_i(t) \in R^2$.

To illustrate this with an example, Figure 3 shows several frames from a scenario with 16 agents. In Figure 3(a), 12 of the agents are arrayed in three teams of four agents in a square formation, ($\{a_0, \ldots, a_3\}$, $\{a_4, \ldots, a_7\}$, $\{a_8, \ldots, a_{11}\}$), with the remaining four agents as singletons. In Figure 3(b), the squares are converging towards the central area and the formations are starting to interleave. In Figure 3(c), the squares are disbanding and those are regrouping into four groups of three, arrayed as triangles. Finally, in Figure 3(d), the triangles are moving away from the central area.



(a) $t = 0$ (b) $t = 24$ (c) $t = 50$ (d) $t = 80$

Fig. 3. (a) An example scenario with three teams of 4 agents, (($\{a_0, \ldots, a_3\}$, $\{a_4, \ldots, a_7\}$, $\{a_8, \ldots, a_{11}\}$) and four singleton agents ($a_{12}, \ldots, a_{15}$); (b) teams maneuver while maintaining formation and converge to central area; (c) the three teams disband and regroup into four teams of 3 agents; (d) the various teams scatter as units. The interleaving of agent formations, the presence of singletons and observation noise (suppressed here) makes the team assignment and behavior recognition challenging.

The goal is to recover a team and a behavior assignment for every agent $a_i \in A$ at every time-step *t*. It is important to note that one cannot, in general, infer the behavior of a team by examining the motion trace of any single agent. Similarly, one cannot assign an agent to a team without confirming that the behavior of the team is legal.

Ideally, one may wish to consider every legal agent-to-team assignment and team-to-behavior assignment at every time-step and then select the sequence that best matches the observed data. However, a straightforward implementation of this idea is computationally infeasible.

STABR analyzes spatio-temporal traces in three stages.

- First, it performs a static analysis of agent positions at each time-step to identify potential agent configurations that may correspond to known formations; these are used as an initial set of agent-to-team assignment hypotheses in later stages. STABR maintains multiple potentially-conflicting assignments for an agent, if there is spatial support.
- Second, STABR examines hypothesized team assignments in isolation and determines whether they have sufficient local spatio-temporal support. Pruning unlikely hypotheses at this stage is crucial since it greatly affects the performance of the last stage. This analysis also enables STABR to determine plausible behavior assignments for each of the surviving hypotheses.

- Third, these agent-to-team hypotheses are used to generate complete partitions over the agents. In the worst case, this state space could be exponential in the number of surviving hypotheses, underscoring the benefits of pruning. STABR then organizes the states (partitions) over the spatio-temporal sequence in the form of a lattice and employs dynamic programming to identify minimal cost solutions. These correspond to agent-to-team and team-to-behavior assignments that are a good fit to the observed sequence.

Experiments on several simulated military maneuvers demonstrated that STABR is accurate at both team assignment and behavior recognition.

Riley used a set of predefined movement models and compare these with the actual movement of the players in set play situation (Riley et al., 2002). In new set play situations the coach then uses the gathered information to predict the opponent agent's behavior and to generate a plan for his own players. The main drawback of Riley’s model is that it is built based on individual movements of players without taking into account the relationships between agents. Raines and colleagues (Nair et al., 2004) presented a system called ISAAC which analyzes a game in mode off-line to generate rules about the success of players. ISAAC used the individual and relational models in an independent way. It tries to discover patterns in each level based on events that affect directly the result of the game. Two key differences between ISAAC and our approach are: we build a model of a team based on behavior patterns, independently of success or failures events; ISAAC is unable to discover the strategic behavior of a team. Bezek and colleagues (Bezek et al., 2006) presented a method to discover pass patterns incorporating domain knowledge and providing a graphic representation for detected strategies. Although their approach obtains tactical behavior patterns, they only consider the players involved in the passes without taking into account the notion of team behaviors. Visser and colleagues (Visser et al., 2001) recognized the formation of the opponent team using a neural networks model. The output was a predefined set of formations. The main difference with our approach is that Visser and colleagues did not represent relations between players. As Visser mentioned in his work, his approach is unable of tracking the changes of formations. This is because the lack of structures due to the absence of relations between players.

4. Multi-Level representation model

We emphasize on the fact of having an expressive representation model which takes into account different aspects exhibited in a team of agents. The adequate representation of these aspects enables the discovering behavior patterns at different levels of abstraction in a complex domain. In this work, we present an expressive representation model able to discover behavior patterns by taking into account various aspects such as individual aspects about agents, relationships between agents (tactical aspect) and formation behaviors (strategy aspect). In order to facilitate the discovery of behavior patterns, we need to have a representation model able to express relevant aspects at different abstraction levels. Such model should endow a reasoning system, through an expressive representation model, with the capacity of discovering strategic, tactical and individual behavior patterns.

The different levels of abstraction, each one representing a different aspect of the team, are built in a bottom-up mode, that is, higher levels are constructed based on lower levels. For instance, the representation of formations of a team is based on the relational level, which is composed of relations between zones. At the same time, each zone represents a relationship between individuals (players). As an example, the formation 4:3:3 represents four defenders, three midfielders and three forwards. The proposed multi-layered representation model is shown in Figure 4.

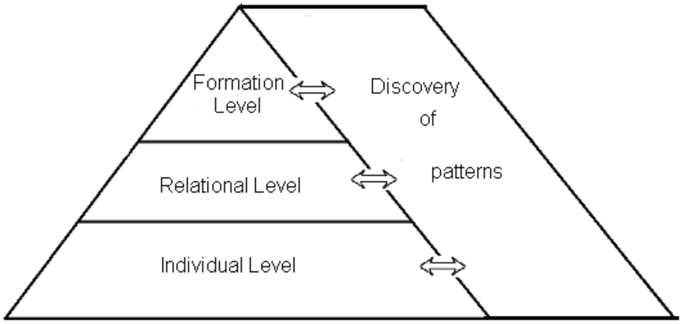


Fig. 4. Representation Model

Individual level. It represents the individual information of the objects in the field, such as players and ball. Such information can be acquired from the Soccer Simulator System directly.

Relational level. It represents the relationship between players.

Formation level. A formation represents the relation among defenses, midfielders and forwards of a team. The formation reveals part of the general strategy of a team. Formations are the way a soccer team lines up its defense, midfield, and attack line during a match. When talking about formations, defenders are listed first and then midfielders and forwards. For example, a code 5:3:2 represents a formation composed by five defenders, three midfielders, and two forwards (see Figure 5). As in the real soccer game, the goalkeeper is not considered as part of the formation. Usually, teams playing in strategic and organized ways search for respecting predefined structures or formations.

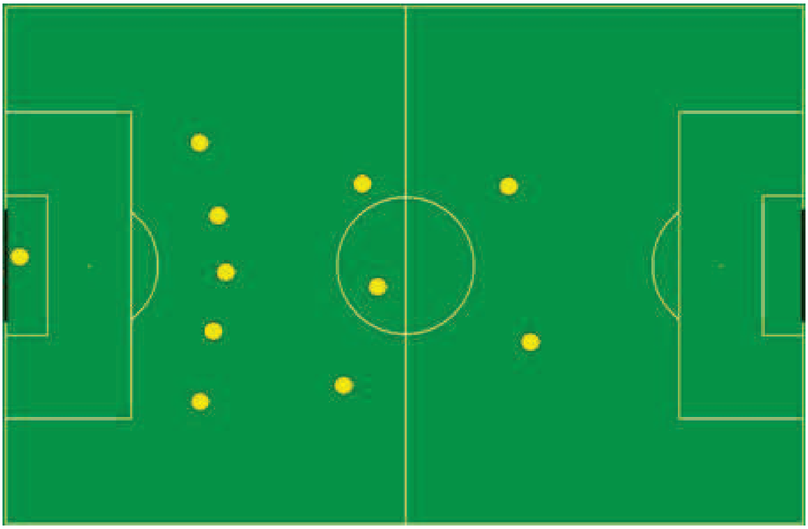


Fig. 5. A 5:3:2 formation

4.1 Recognition of formations

The focus of this work is on teams that play following patterns of high level of abstraction (formations) based on a distribution of zones named Defensive (D), Middle (M) and Attack (A), as in classic soccer game. These patterns will be represented as follows: D:M:A. Due to the dynamic conditions of the soccer game, the players are in constant movement and temporally breaking the alignment of players belonging to a zone. To handle the constant changes without an expressive representation of the relations between players can result in an inefficient way of recognizing formations submitted to a dynamic environment. In the next section will be explained how the zones and the players belonging to them are recognized in this work

4.1.1 Recognition of team zones

As in human soccer domains the players in robotic soccer should tend to be organized. That is, each player has a strategic position that defines its movement range in the soccer field. The role of a player is quite related with a predefined area within which an individual player can play basically in the field. Any behaviors of a player depend on its current role. According to the position of the player, roles in robotic soccer can be divided into four types: goalkeeper, defenders, midfielders and forwards. Different roles are associated with different positions and different behaviors that players assume. However, due to the dynamic changing conditions of a match, a defender could become a forward temporarily as his team is trying to attack. So the roles of a player are dynamically changing. Consequently, the recognition of formation patterns is difficult due to the dynamic and real time conditions of the environment. In a first step, we will discover what players belong to what zone. For this, the clustering algorithm, K-means (MacQueen, 1967), is applied. K-means classifies a given data set through a certain number of clusters (assume *k* clusters) fixed a priori. In this work, k=3 such that three zones will be defined: defensive, middle and attack zones. From the log file (game film), the data from one team is extracted and K-means is applied in each simulation cycle of the game. The positions of each player, with respect to the x axis, are taken as the input of the clustering algorithm and the output of clustering is the classification, according to their x position, of all players of the team in the three clusters. Clustering algorithm is useful to determine the three zones of a team but it is not able to represent the multiple relations between players of each zone. Given that patterns of formations are based on relations that determine structures then an additional model is crucial for the recognition of formation patterns. The next section describes an adequate representation model able to facilitate the recognition of formation patterns.

4.2 Topological Structure Model

A formation is represented by a set of relations between players. Thus, the relations represent the structure that supports a formation. So, a change of relations between players entails a change of formation. It is needed at least the change of one relation to transform one structure into another one. Constant changes of relations could occur because the multiple relations in a formation and the dynamic nature of a match. Figure 6(a) illustrates the relations of each one of the players with the rest of their teammates. A total of 90 relations are obtained by $\boldsymbol{n(n-1)}$, where n represents the number of players. This formula considers two relations by each pair of players. Thus, one relation is represented by the link from player A to player B and the second one from player B to player A. For practical reasons, just one of these relations is considered. Thus, the total of relations is 45. Figure 6(b) illustrates these 45 relations.

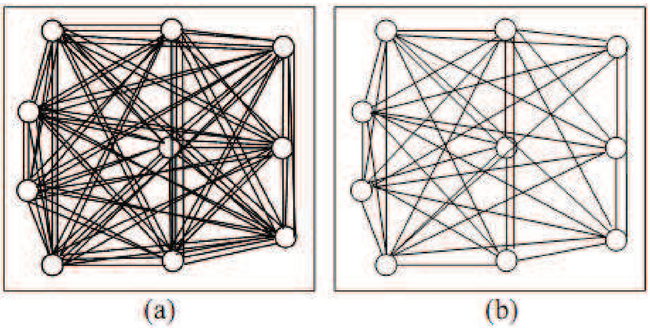
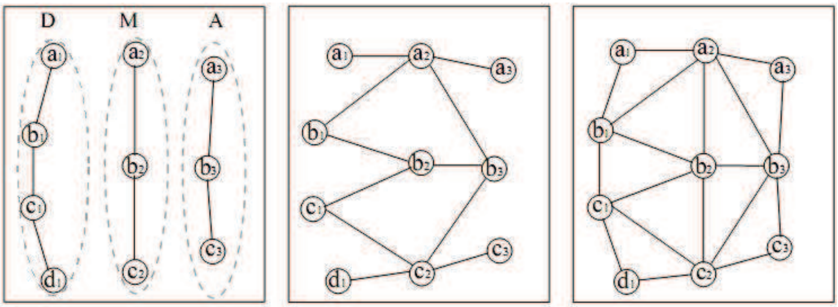


Fig. 6. All possible relations between players of a soccer team. (a) 90 relations and (b) 45 relations.

On the one hand, the control of such number of relations becomes very difficult to be managed because any change of relations would produce a change of structure. In addition, it could happen that several changes of relations occur at the same time then the problem of detecting what relations are provoking changes of structures becomes much more difficult to be managed. On the other hand, the 45 relations are not relevant in a real match, because a relevant relation is the one in which a player uses to exchange passes and positions in a strategic way. In this work, the goal is to build a simple but robust structure based on relevant relations modeled by a planar graph.

A graph *G* is planar if it can be represented on a plane in such a way that the nodes represent different points and two edges should be encountered only at their ends. The intersection of two edges out of their ends breaks the planar property of the graph G. This graph *G* is also named as planar topological graph (Berge, 1983). Two or more graphs are topologically the same if they can be transformed by elastic deformations until their form coincides.

The relevant relations used to build the topological structure are related with the notion of neighborhood. Thus, an agent remains related with his closer neighbor belonging to his zone (defensive (D), medium (M) or attack (A)), and his closer neighbor belonging to the neighbor zone as illustrated in Figure 7(a) and Figure 7(b). Figure 7(c) shows the integration of both kinds of relations for a 4:3:3 formation.



(a) (b) (c)

Fig. 7. (a) Step 1. Neighbor nodes of the same zone are linked. (b) Step 2. Neighbor nodes of neighbor zones are linked. (c) Planar graph obtained from step 1 and step 2.

Figure 7(c) shows the planar graph represented by triangular sub-graphs as result of applying the previous two steps. The total number of relations of a graph, which has been built based on the method described above, is given by N_m +15; where N_m is the number of nodes of the middle zone (Due to the lack of space the deduction of this formula is not described in this work). For instance, for a formation 4:4:2, the number of relations will be 19, because $N_m = 4$. The advantages of this method that the number of relations has been reduced from 45 to 19 for the formation 4:4:2. Then, 26 relations have been eliminated. Triangular graphs are able to assume a topological behavior (Ramos & Ayanegui, 2008a). That is, even if a structure is deformed because positional changes of nodes, the topological property of the triangular graphs helps to preserve the structure.

4.3 Pattern Recognition Process

Figure 8 shows the process to recognize patterns of formations and changes of structures that support the formations. The first module serves to determine the zones by using a clustering algorithm; the second module builds the multiple relations which are expressed by a topological graph and finally in the third module the changes of structures are detected if topological properties of a defined structure have been broken.

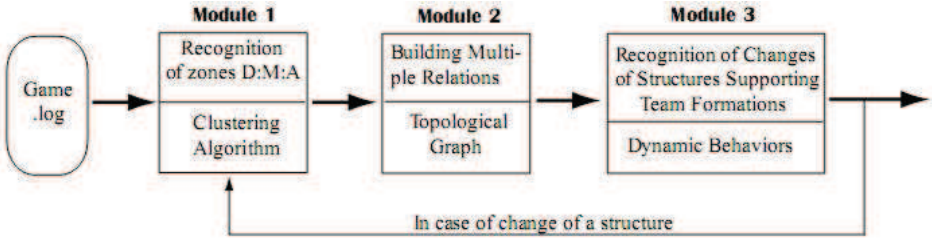


Fig. 8. Process to recognize pattern formations

Module 1. *Recognition of team zones*. The algorithm of clustering is performed during the first cycles of the match and it is stopped until the number of players in each group does not change. In this way, the three zones of a team, defensive, middle and attack zones are recognized.
Module 2. *Building multiple relations and a topological graph*. Based on the three zones recognized by the clustering algorithm and relevant multiple relations a topological planar graph is built.
Module 3. *Recognition of Changes of Structures that support Team Formations*. Changes of structures are detected if topological properties of a defined structure have been broken. A topological graph is, by definition, a planar graph (Berge, 1983). In a planar graph any pair of nodes belonging to the graph can be linked without any intersection of links. Otherwise, if the topological property of the graph has been broken then another structure supporting a formation should be built. Intersections occur when players change their roles in order to build a new formation or due to reactive behavior in response to the opponent. If intersections of links occur, clustering algorithm should redefine the zones and a new topological graph should be built.

5. Discovering of Tactical Behaviour Patterns

The process to discover tactical behavior patterns is illustrated in Figure 9. The following six steps describe such process:

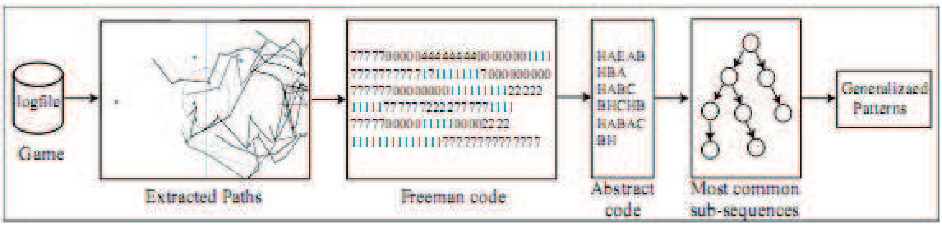


Fig. 9. The steps to discover the tactical behavior patterns

Step 1.Read *logfile*. Input data mainly related with players and ball positions;

Step 2.Extraction of similar paths. A set of ball's paths occurring under similar contexts are extracted. The extracted paths in Figure9 shows paths starting from the middle zone of the field and then distributed either to the right or to the left side until ball reach a zone close to the goal;

Step 3.First Freeman codification. The set of extracted paths are coded to be represented by a sequence of orientations using a Freeman codification (Freeman, 1973) which is composed of eight orientations.

Step 4.Second Freeman codification. The sequence of step 3 is recoded to obtain a more abstract code. Let A,B,...,H be the new abstract segments where each one represents a freeman code sequence with the same orientation, such that, A represents the sequence of 0's, B represents the sequence of 1's, and so on. Thus, a path coded as 7-7-7-1-1-1 can be represented by the code HB;

Step 5.Identification of most frequent sub-sequences. A method based on a generalization of a tree is applied to discover the general behavior patterns representing the paths of tactical plays. For instance, let's take two paths: BAH and ABA. Let's suppose that the trie is empty. It will first insert BAH into it. It will then insert the two remaining suffixes of BAH: {AH, H}. Next, it will then insert the next path and its suffixes: {ABA, BA, A}, into the trie. The most common single sub-sequence is A, the most common two subsequences is BA.

Finally, the players and zones are associated to the generalized paths. The topological structures used to track formations have been a very good support to determine the players participating in tactical plays, as well as the zones through which the plays have taken place. Thanks to the topological graph, we are able to know at each instant of the game the players and their relations participating in a play.

6. Experimental Results

In this section, important experimental results are analyzed. They are derived from two teams: The TsinghuAeolus soccer team, who won the Simulation RoboCup Championship in 2002. It is presented an analysis of the match between TsinguAeolus vs. Everest; and theWrightEagle team, who won the second place in the same competition that held in 2007. The model has been proven in nine matches, but for the relevance of the teams, we present the analysis of results of two matches, one for the TsinguAeolus and one for the WrightEagle. Figure 10 shows a sequence simulation cycles that represent the structures involving the soccer-agents in a path of a tactical play. Because of the lack of space it is shown some of the sub-graphs that compose the total sequence of sub-graphs representing the path (in fact, there are approximately 50 sub-graphs for this tactical play). As can be seen, the shadowed sub-graphs contain the soccer agents involved in the tactical plays. They are in this case: the middle center, the right middle, the right forward, the center forward and the left forward.

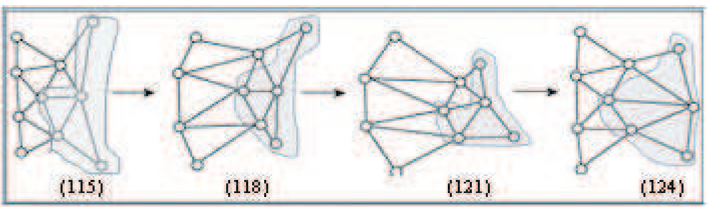


Fig. 10. Sequence of sub-graphs representing a tactical play incorporating agents and field zones

As first step, the paths of the ball were extracted to be analyzed and coded by the code of Freeman. In this way the set of paths can be compared numerically by measuring the similarity between them. Another advantage of this codification is that we can have an idea about how long the paths are. However, what is interesting in this analysis is not exactly how long a path is, but, from the point of view of behavior, the form adopted by the path and obviously the properties associated with the intention or purpose of it, in this case to get close to a position of shooting to the goal. Due to these reasons, it is proposed in this work a more abstract representation. Then the paths coded by the code of Freeman have been recoded to obtain a more abstract code. The paths represented by abstract codes have facilitated the application of the model to discover behavior patterns related with tactical plays. It is important to point out that similar paths are not necessary those to end in a goal, but those that assume a similar behavior from the start of the path to the final objective. Figure 11 illustrates two shapes of generalized paths of tactical behaviors played through the right and left side of the terrain. These generalized paths correspond to the TsinghuAeolus team.

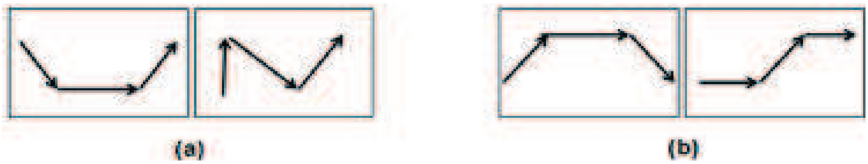


Fig. 11. Generalized paths of tactical behaviors: a) Attacks by right side and b) Attacks by left side

For the case of the WrightEagle team, they played in the right side, Figure 12 shows the extracted paths that get close to the opposite goal and Figure 13 shows two shapes of discovered generalized paths. Based on the results obtained, it is observed that the model to obtain the paths representing the tactical plays do not depend on the analyzed team. The topological structures used to track formations have been a very good support to determine the players participating in tactical plays, as well as the zones through which the plays have taken place.

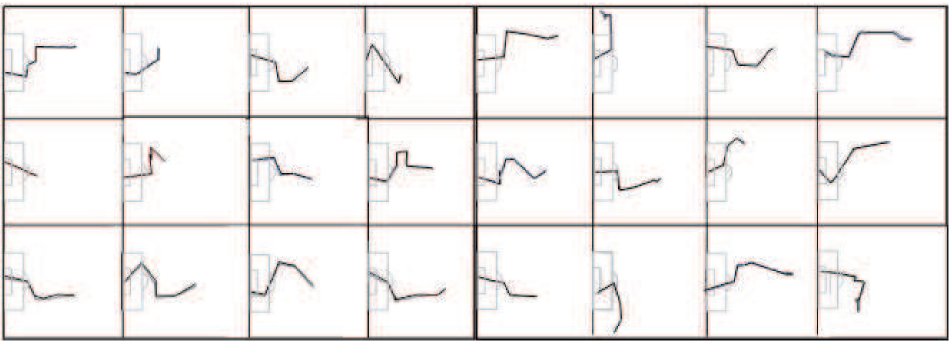


Fig. 12. Extracted paths that get close to the opposite goal. The team is attacking from left to right side.

Fig. 13. Two shapes of generalized paths of tactical behaviors

7. Conclusions

This chapter has been focused on the analysis of relevant aspects related with interaction and behavior models of cooperative multi-agent systems participating in a task. Our main interest in these aspects concerns the models of tracking of multi-agent behaviors. Along with the concept of tracking and behaviors, the concepts of coordination, prediction and opponent models have been revised in order to have a general panorama around of this important area in multi-agent systems applied to soccer agent robotics.

Behavioral aspects of cooperative agents may give us precious information about individual, relational and functional roles that the agents assume during the different steps of a task. The discovery of tactical plays and the recognition of formations supporting strategies of team represent relevant information to implement counter strategies or tactics to reduce the performance of the opposite team or, in the best of cases, to beat it. Nevertheless, the dynamic nature of soccer matches along with the multiple interactions between players difficult enormously the task of discovery. The model based on topological graphs has contributed importantly to manage the difficulties due to the dynamic nature of the soccer game. It can facilitate the tracking of formations. In addition, it provided the algorithm of discovery tactical plays with important information concerning the players participating in such plays.

An original idea described in this chapter is related with the double codification of the paths, which has facilitated the interpretation of paths to implement the algorithm described in section 4.3. The discovered paths can be considered as generalized because they were obtained from a set of paths by applying the generalization algorithm described in section 4.3. This chapter dealt with offensive actions to be modeled as an opponent model. However, a richer spectrum of team behaviors should take into account also defensive strategies and tactics, which is an important line of research.

8. References

Nair, R.; Tambe, M.; Marsella S.; Raines, T. (2004). Automated assistants for analyzing team behaviors, *Journal of Autonomous Agents and Multi-Agent Systems*, 8(1).

Ramos, F. & Ayanegui, H. (2008a): Discovering Tactical Behavior Patterns Supported by Topological Structures in Soccer-Agent Domains. In Proceedings of the *7th International Joint Conference on Autonomous Agents and Multi-agent Systems*, Padgham, Parkes, Muller and Parson (eds.), May 12-16, Estoril, Portugal, pp. 1421-1424.

Kuhlmann, G.; Stone, P.; Lallinger, J. (2005). The ut austin villa 2003 champion simulator coach: A machine learning approach. In *RoboCup 2004,* Nardi, D., Riedmiller, M., Sammut, C.,Santos-Victor, J. (eds.). LNCS (LNAI), vol. 3276, pp. 636–644. Springer, Heidelberg.

Lattner, A., D.; Miene, A.; Visser, U. & Herzog, O. (2005). Sequential pattern mining for situation and behavior prediction in simulated robotic soccer. In *RoboCup 2005,* A. Bredenfeld, A. Jacoff editors, volume 4020 of LNCS, pages 118–129. Springer, Heidelberg.

Noda, I. & Frank, I. (1998). Investigating the complex with virtual soccer. In Heudin, J.-C., editor, *Virtual Worlds*, volume 1434 of Lecture Notes in Computer Science, pages 241–253. Springer.

Kitano, H.; Tambe, M.; Stone, P. & Veloso, M. (1997). The robocup synthetic agent challenge 97. In Kitano, H., editor, *RoboCup 1997*, volume 1395 of LNCS, pages 62–73. Springer.

Riley, P.; Veloso, M. & Kaminka, G. (2002). An empírical study of coaching. *On proceedings of Distribuited Autonomous Robotic Systems* 6, Spring-Verlag.

Bezek, A.; Gams, M. & Bratko, I. (2006). Multi-agent strategic modeling in a robotic soccer domain. In Nakashima, H., Wellman, M.P., Weiss, G., Stone, P. (eds.) *Autonomous Agents and Multi-Agent Systems*, pp. 457–464. ACM Press, New York

Visser, U.; Drcker, C.; Hbner, S.; Schmidt, E. & Weland, H.-G. (2001). Recognizing formations in opponent teams. In Stone, P., Balch, T.R., Kraetzschmar, G.K. (eds.) *RoboCup 2000*. LNCS (LNAI), vol. 2019, pp. 391–396. Springer, Heidelberg

MacQueen, J.B. (1967). Some Methods for classification and Analysis of Multivariate Observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, vol. 1, pp. 281–29. University of California Press (1967).

Berge, C. (1983). Graphes. *Guathier-Villars*.

Freeman, H. (1973). On the encoding of arbitrary geometric configurations. *IRE Transactions*.

Durfee, E. H,. Lesser, V.R. and Corkill, D.D. (1989b). Trends in cooperative distributed problem solveing. IEEE Trnsactions on Knowledge and Data Engineering, 1(1), 63-83.

Wooldrige, M. & Jennings, N.R. (1994). Formalizing the cooperative problem solving process. In Proceedings of the 13th International Workshop on Distributed Artificial Intelligence (IWDAI-94), Lake Quinalt, WA, pp. 403-417. Reprinted in Huhns and Singh (1998).

Adler, M.R. et al. (1989). Conflict resolution strategies for non hierarchical distributed agents. In Distributed Artificial Intelligence (eds. L. Gasser and M. Huhns), Volume 2, pp. 139-162. Pitman, London and Morgan Kaufmann, San Mateo, Ca.

Galliers, J.R. (1988b). A theoretical frameqork for computer models of cooperative dialogue, acknowledging multi-agent conflict. PhD thesis, The Open University, UK

Galliers, J.R. (1990). The positive role of conflict in cooperative multi-agent systems. In Descentralized A.I. Proceedings of the First European Workshop Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-89) (eds. Y. Demazeau and J.P. Miller), pp-33-48. Elsevier, Amsterdam.

Klein, M. & Baskin, A.B. (1991). A computational model for conflict resolution in cooperative design systems. In CKBS-90. Proceedings of the International Working Conference on Cooperative Knowledge Based Systems (ed. S.M. Deen), pp. 201-222. Springer, Berlin.

Lander, S., Lesser, V.R. & Connel, M.E. (1991). Conflict resolution strategies for cooperating expert agents. In CKBS-90. Proceedings of the International Working Conference on Cooperating Knowledge Based Systems (ed. S.M. Deen), pp. 183-200. Springer, Berlin.

Ephrati, E. & Rosenschein, J.S. (1993). Multi-agent planning as a dynamic search for social consensus. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-93). Chambéry, France, pp. 423-429.

Rosenschein, J.S. and Zlotkin, G. (1994). Rules of Encounter: Designing Conventions for Automated Negotiation among Computers. MIT Press, Cambridge, MA.

Horling, B. & Lesser, V. (2004). A survey of multi-agent organizational paradigms. *Knowledge Engineering Review,* Vol. 19, Issue 4, pp. 281-316., ISSN: 0269-8889. Cambridge University Press, New York, NY, USA.

Chernova, S. & Veloso, M. (2008). Teaching multi-robot coordination using demonstration of communication and state sharing. In *Proceedings of the 7th international Joint Conference on Autonomous Agents and Multi-agent Systems - Volume 3* (Estoril, Portugal). International Conference on Autonomous Agents. International Foundation for Autonomous Agents and Multi-agent Systems, Richland, SC, 1183-1186.

Jansen, R. & Sturtevant, N. 2008. A new approach to cooperative pathfinding. In *Proceedings of the 7th international Joint Conference on Autonomous Agents and Multi-agent Systems - Volume 3* (Estoril, Portugal). International Conference on Autonomous Agents. International Foundation for Autonomous Agents and Multi-agent Systems, Richland, SC, 1401-1404.

Candea, C.; Hu, H.; Iocchi, L.; Nardi, D. & Piaggio, M. (2001). Coordination in Multi-Agent RoboCup Teams. In *Robotics and Autonomous Systems* , Vol. 36 (2001) , p. 67-86.

Aler, R.; Valls, J.; Camacho, D. & Lopez, A. (2009). Programming Robosoccer agents by modeling human behavior. *Expert Systems with Applications*, Volume 36, Issue 2, Part 1, pp. 1850-1859.

Lin, L., Seo, Y., Gen, M., and Cheng, R. (2009). Unusual human behavior recognition using evolutionary technique. C*omputers & Industrial Engineering*, *Volume 56, Issue 3*, *April 2009*, *pp. 1137-1153.*

Balch, T.; Khan, Z. & Veloso, M. (2001). Automatically Tracking and Analyzing the Behavior of Live Insect Colonies. *AGENTS'01, May 28-June 1, 2001, Montreal, Quebec, Canada.*

Luotsinen, L. J. & Bölöni, L. (2008). Role-based teamwork activity recognition in observations of embodied agent actions. In *Proceedings of the 7th international Joint Conference on Autonomous Agents and Multi-agent Systems - Volume 2* (Estoril, Portugal, May 12 - 16, 2008). International Conference on Autonomous Agents. International Foundation for Autonomous Agents and Multi-agent Systems, Richland, SC, 567-574.

Hindriks, K. & Tykhonov, D. (2008). Opponent modelling in automated multi-issue negotiation using Bayesian learning. In *Proceedings of the 7th international Joint Conference on Autonomous Agents and Multi-agent Systems - Volume 1* (Estoril, Portugal, May 12 - 16, 2008). International Conference on Autonomous Agents. International Foundation for Autonomous Agents and Multi-agent Systems, Richland, SC, 331-338.

Mulder, F. & Voorbraak, F. (2003). A formal description of tactical plan recognition. *Information Fusion*, Volume 4, Issue 1, March 2003, pp. 47-61.

Henniger & Madhavan. (2004). *Robotics and Autonomous Systems*, Volume 49, Issues 1-2, 30 November 2004, pp. 91-103.

Tambe, M. & Rosenbloom, P.S. (1996). Architectures for Agents that Track Other Agents in Multi-agent Worlds, *Intelligent Agents II*, Springer Verlag Lecture Notes in Artificial Intelligence, LNAI 1037.

Tambe, M.; Johnson, W. L.; Jones, R.; Koss, F.; Laird, J. E.; Ronsenbloom, P.S. & Schwamb, K. (1995). Intelligent agents for interactive simulation environment. *AI Magazine (16) 1,* Spring 1995.

Muñoz, R. (2008). A Bayesian plan-view map based approach for multiple-person detection and tracking. *Pattern Recognition*, *Volume 41, Issue 12*, *December 2008*, *Pages 3665-3676*.

Muñoz, R.; García, M. & Medina, R. (2008). Adaptive multi-modal stereo people tracking without background modelling. *Journal of Visual Communication and Image Representation*, Volume 19, Issue 2, February 2008, Pages 75-91.

Muñoz, R.; Aguirre, E.; García, M. (2007). People detection and tracking using stereo vision and color. *Image and Vision Computing*, Volume 25, Issue 6, 1 June 2007, pp. 995-1007.

Schwager, M.; Anderson, D. M.; Butler, Z. & Rus, D. (2007). Robust classification of animal tracking data. *Computers and Electronics in Agriculture*, Volume 56, Issue 1, March 2007, pp. 46-59.

Noldus, L.; Spink, A. & Tegelenbosch, R. (2002). Computerised video tracking, movement analysis and behavior recognition in insects. *Computers and Electronics in Agriculture*, Volume 35, Issues 2-3, August 2002, pp. 201-227.

Ukita, N. & Matsuyama, T. (2005). Real-time cooperative multi-target tracking by communicating active vision agents. *Computer Vision and Image Understanding*, Volume 97, Issue 2, February 2005, pp. 137-179.

Xiao, F.; Wang, L.; Chen, J. & Gao, Y. (2009). Finite-time formation control for multi-agent systems. *Automatica*, In Press, Corrected Proof, Available online 19 August 2009.

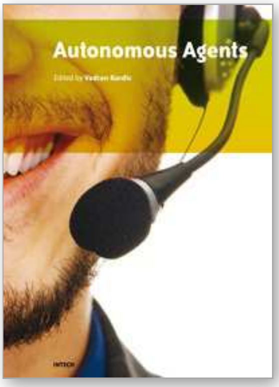
Porfiri, M.; Roberson, G. & Stilwell, D. (2007). Tracking and formation control of multiple autonomous agents: A two-level consensus approach. *Automatica*, Volume 43, Issue 8, August 2007, pp. 1318-1328.

Ramos, F. & Ayanegui, H. (2008b). Discovering Behavior Patterns in Muti-agent Teams. In *Agent and Multi-Agent Systems: Technologies and Applications*, Second KES International Symposium, KES-AMSTA 2008, Incheon, Korea, March 26-28, 2008. pp. 391-400.

Raines, T.; Tambe, M. & Marsella, S. (2000). Automated assistant to aid humans in understanding team behaviors. In *Agents-2000.*

Sukthankar, G. & Sycara, K. (2006). Simultaneous team assignment and behavior recognition from spatio-temporal agent traces. In *Proceedings of 21st national conference on artificial intelligence AAAI 2006*.

Fischler, M. & Bolles, R. (1981). Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography. *Communications of the ACM* 24 (6).



Autonomous Agents

Edited by Vedran Kordic

ISBN 978-953-307-089-6

Hard cover, 130 pages

Publisher InTech

Published online 01, June, 2010

Published in print edition June, 2010

Multi agent systems involve a team of agents working together socially to accomplish a task. An agent can be social in many ways. One is when an agent helps others in solving complex problems. The field of multi agent systems investigates the process underlying distributed problem solving and designs some protocols and mechanisms involved in this process. This book presents a combination of different research issues which are pursued by researchers in the domain of multi agent systems.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Fernando Ramos and Huberto Ayanegui (2010). Tracking behaviours of cooperative robots within multi-agent domains, Autonomous Agents, Vedran Kordic (Ed.), ISBN: 978-953-307-089-6, InTech, Available from: http://www.intechopen.com/books/autonomous-agents/tracking-behaviours-of-cooperative-robots-within-multi-agent-domains

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri

Slavka Krautzeka 83/A

51000 Rijeka, Croatia

Phone: +385 (51) 770 447

Fax: +385 (51) 686 166

www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai

No.65, Yan An Road (West), Shanghai, 200040, China

中国上海市延安西路65号上海国际贵都大饭店办公楼405单元

Phone: +86-21-62489820

Fax: +86-21-62489821