

# Cooperation Issues and Distributed Sensing for Multi-Robot Systems

Enrico Pagello, *Member IEEE*, Antonio D'Angelo and Emanuele Menegatti *Member IEEE*

**Abstract**—The paper considers the properties a Multi-Robot System should exhibit to perform an assigned task cooperatively. Our experiments regard specifically the domain of RoboCup Middle-Size League (MSL) competitions. But the illustrated techniques can be usefully applied also to other service robotics fields like, for example, videosurveillance. Two issues are addressed in the paper. The former refers to the problem of dynamic role assignment in a team of robots. The latter concerns the problem of sharing the sensory information to cooperatively track moving objects. Both these problems have been extensively investigated over the past years by the MSL robot teams. In our proposal, each individual robot has been designed to become reactively aware of the environment configuration. In addition, a dynamic role assignment policy among teammates is activated, based on the knowledge about the best behavior that the team is able to acquire through the shared sensorial information. In the experiment section, we present the successful performance of Artisti Veneti robot team at the MSL Challenge competitions of RoboCup-2003 to show the effectiveness of our proposed hybrid architecture, as well as some tests run in laboratory to validate the Omnidirectional Distributed Vision System which allows to share the informations gathered by the omnidirectional cameras of our robots.

**Index Terms**—Multi-Robot Systems, RoboCup Middle-Size League Competitions, Cooperative Behaviors, Distributed Sensing, Distributed Vision System

## I. INTRODUCTION

A Multi-Robot System (MRS) is characterized by attributes like size, composition, communication topology and range [12], as well as agent redundancy and collective intelligence [20]. Thus, solving cooperatively complex tasks requires an intelligent multi-robot system to show dynamic group reconfigurability and communication among individuals. This can be achieved either through an *explicit* or an *implicit* approach, or through a combination of both, with the specification of whether each individual robot shares or not a common goal [24]. In the *explicit communication*, signals are intentionally shared between two or more individuals, while in the *implicit communication*, the robots observe other robots' actions. Contrary to what one might expect, intelligent cooperation does not necessarily require explicit communication among robots. Indeed, we have previously implemented collective behavior through implicit communication [32], [31] and [33]. In this paper, instead, we show how collective actions can be achieved through the exchange of roles that can be forced by an appropriate communication mechanism.

E. Pagello and E. Menegatti are with the Autonomous Systems Laboratory (AIS-Lab), Dept of Information Engineering, University of Padua, Italy.

A. D'Angelo is with the Dept. of Mathematics and Computer Science, Udine University, Italy

Cooperation abilities are crucial for an MRS that must operate in a dynamic environment. In particular, we want to force the emergence of cooperative abilities in the context of MRS that perform advanced tasks where flexibility and reliability are especially required. One such context is the MSL RoboCup competition, where individual robots are often engaged in collective actions. To improve team performance, omnidirectional vision and role-based coordination have been largely introduced by most teams over the years. Indeed, to effectively coordinate the team actions, each individual in the team must be able to manage appropriately its role and to exchange information with its teammates.

In our approach, we have addressed together both the role assignment problem and the distribution of sensor data. On one hand, we investigate under what conditions an MRS is able to perform a given task cooperatively by using a dynamic role assignment mechanism. On the other, we discuss the problem of developing a distributed sensing system based on omnidirectional vision sensors to cooperatively track and share the information about moving objects. The combination of these two approaches has proven very suitable in making each individual capable of developing a cooperative behavior.

As it is well known, a coordination mechanism for an MRS operating in a dynamic environment should provide flexibility and adaptability, where individual robots should be able to change dynamically their behaviors, in order to execute different types of cooperative tasks. To this aim, robots do not need to build a complete global state of the world. It has been shown that a role assignment mechanism is able to allow robots to change their role dynamically during the execution of a task, based on the partial information that each individual has about its own task and the operating environment [9]. In our approach, each robot has been designed to become aware of distinguishing configuration patterns in the environment by evaluating descriptive conditions as macroparameters at the reactive level. In parallel, the interaction with a deliberative level activates the dynamic role assignment among teammates on the basis of knowledge about the best behavior that the team should adopt.

When the environment is static, the agent can analyze its subcomponents, and store the acquired information in a sort of memory [22]. But if the environment is dynamic, this approach no longer works, because the information that can be retrieved from the memory of the agent is no longer up-to-date. This is one of the reasons why in dynamic environments mobile robots are increasingly fitted with omnidirectional vision systems [39]. These systems provide in one shot a complete view of the surrounding of a single robot. Nevertheless, in a highly



Fig. 1. Picture taken during experiments to test the Omnidirectional Distributed Vision System.

dynamic environment such as the RoboCup soccer fields, this is not enough. Since the evaluation process, can require gathering a large amount of sensor data, the omnidirectional vision by itself cannot be sufficient. Given its low resolution, it does not solve the problem of perceiving occluded or very distant objects<sup>1</sup>. However, since each individual is part of a robot team, the sensorial horizon of the single robot can be extended by using the information perceived by the teammates. The separately gathered information can be broadcast to all the teammates allowing every robot to fuse its own measurements with the information received from the others, thus constructing its own "vision of the world".

If the robot is part of a multi-robot team, the sensorial horizon of the single robot can be extended by using the information perceived by the teammates. In [26] we proposed an *Omnidirectional Distributed Vision System* (ODVS) capable of tracking moving objects in a highly dynamic environment by sharing the information gathered by every single robot.

Even though the ODVS was developed for the RoboCup domain, it could be used in more general situations, such as surveillance systems or intelligent space applications. Every time the application requires the monitoring of a large area that cannot be framed in the field of view of a single sensor, the cooperation of different sensors becomes extremely useful. Some examples are found in [19] where a Distributed Vision System composed of perspective cameras is able to drive a robot through a toy-scale model of a town, in [29] where multiple perspective cameras can track people moving from one room to another, and in [21] in which the Distributed Vision System is able to support the activity of robots and humans.

As such, the sharing mechanism among the vision sensors, in combination with a continuous role exchange, enhances the capabilities of the robot team. This approach is very general and can be used also in other applications where a team of robots has to perform complex dynamic tasks in any environment with multiple moving objects.

Paper sections are organized as described in the following. Section II recalls the fundamentals of behavior-based architec-

<sup>1</sup>Note that usually, the effective range for omnidirectional sensors is shorter than for perspective cameras due to the lower resolution.

tures to be adopted for each single robot of a team in order to clarify how to build compound behaviors from primitive ones. Section III illustrates the basic technique for using a multipart omnidirectional mirror in a perception module for Gaussian single-sensor observations. Section IV gives the details of the hybrid architecture implemented on each team robot, to allow it to develop the desired complex behaviors, and to coordinate its action with its teammates. Section V explains how multiple observations are fused among the robot of the same team, in order to enhance their capability in achieving coordinated actions. Section VI documents a set of experiments that have been carried on a RoboCup Middle-size League game field to validate our approach. Finally, the conclusions are presented in Section VII

## II. USING A BEHAVIOR-BASED APPROACH

**I**N this section, first we discuss how to build primitive behaviors to obtain sensorimotor coordination for a single robot. Then, in section IV, we show how compound behaviors can be constructed only by processing the information from the environment and from the other team robots in a suitable way.

### A. Implementing Schemas

The behavior-based approach [6] assumes a robot to be *situated* within its environment. Moreover, since robots are not merely information processing systems, their *embodiments* require that both all acquired information and all delivered effector commands must be transmitted through their physical structure. Different research areas like biology, ethology and psychology, have contributed to the design of robot control. Among them, *schema*-based theories have been adapted by Arbib [1] to build the basic blocks of robot behaviors. In this perspective, a *schema* is a generic template for doing some activity which is parameterized and created like a class (*schema instantiation*).

Schema-based methodologies are widely used in robotics. So, *motor schemas*, as they were proposed and developed by Arkin [3], are the *basic units of behavior from which complex actions can be constructed; they consist of the knowledge of how to act or perceive as well as the computational process by which they are enacted* [4]. His schemas are always active and produce outputs as action vectors which are summed up. Our implementation assumes only one schema to be active at a time, in a winner-take-all fashion. Moreover, the output is not a continuous signal but either a motor command to feed some servo or an evaluated condition affecting the activation/inhibition mechanism for another schema.

Following [2], we implemented a primitive *behavior* with one *motor schema*, representing the physical activity, and one *perceptual schema* which includes sensing. The resulting governor's unit of each individual robot is a hybrid architecture whose deliberative/reactive trade-off stems from the hierarchical organization of its behaviors. In this perspective, each behavior is implemented at some level and it can use perceptual schemas coming from the underlying level, eventually triggering one selected behavior at that level. Thus, the

<i>learning</i>	<b>6</b>
<i>deliberative level</i>	<b>5</b>
<i>dynamic role assignment</i>	<b>4</b>
<i>individual goal triggering</i>	<b>3</b>
<i>implicit coordination</i>	<b>2</b>
<i>reactive level</i>	<b>1</b>
<i>perceptron</i>	<b>0</b>

level

Fig. 2. The hierarchical levels of control for each individual robot, that represent the different levels of abstraction.

overall architecture is organized at many levels of abstraction, the lowest one being directly coupled with the environment by the robot servos. Each level is populated by a set of control units, which are schema-based *behaviors* receiving sensor information by monitoring the lower level and acting on some *releaser*. We can build a behavior at level  $k+1$  using perceptual information coming from the underlying level  $k$  and controlling one of these behaviors.

More explicitly, consider two basic behaviors like *defendArea* or *carryBall*. They can be implemented in C++ as motor schemas accessing directly robot effectors. On top of these, we can build two primitive behaviors like *playDefensive* and *chaseBall* by simply appending a perceptual schema to a motor schema, as explained by the following behavior constructing rules:

$$\begin{aligned} \textit{playDefensive} & : \textit{seeBall} \rightarrow \textit{defendArea} \\ \textit{chaseBall} & : \textit{haveBall} \rightarrow \textit{carryBall} \end{aligned}$$

The perceptual schemas *seeBall* and *haveBall*, also implemented in C++, allow to access virtual sensor devices like *senseBall* and *touchBall* which are fed by robot physical sensors. A behavior is fired by an activation-inhibition mechanism built on evaluating-condition patterns. Thus, a primitive behavior at reactive level results in appending just one perceptual schema to one motor schema in order to get the sensorimotor coordination that the individual robot is equipped with. The reactive level uses only information coming from sensors and feeds motors with the appropriate commands. Compound behaviors appear only at higher levels, where they receive more abstract information about the environment, filtered by lower behavior functioning.

As suggested by Fig. 2, the control structure of each robot has been organized into different layers, each of which represents a different level of abstraction such that an upper level results in a more abstract handling of the environment. So, the implicit coordination layer assumes that perceptual patterns represent events generated by other individuals, either opponents or teammates. Moreover, the corresponding schemas can control the underlying reactive behaviors but, at the same time, they are also triggered by the individual goals every robot should pursue. The higher layers refer to the cooperation capabilities that any robot could exhibit with its teammate while a cooperative behavior emerges. This is described in Section IV.

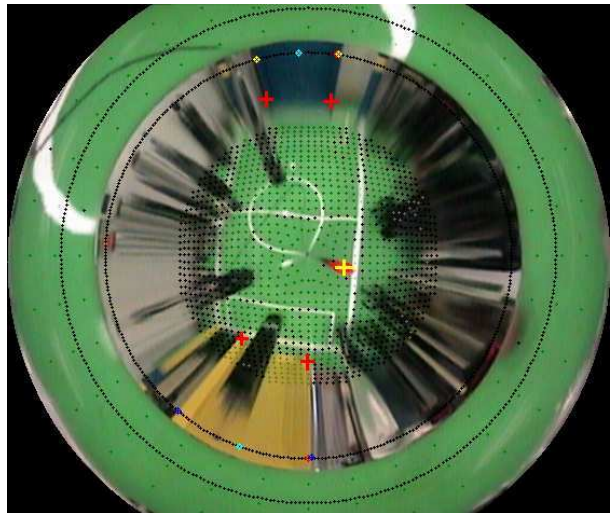


Fig. 3. An omnidirectional image processed by the Vision Agent of the perception module. Note that the ball has been detected as the red blob and marked with a yellow cross. The goals have been detected and marked with red crosses. The black dots are the sample grid used to process the image in a discrete fashion.

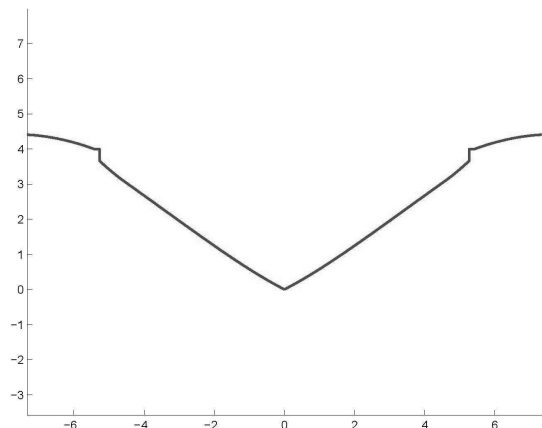


Fig. 4. Profile plot of the omnidirectional mirror used to grab the picture of Fig. 3. Note that the profile is generated point by point to achieve the desired resolution in the different parts of the image.

### III. SINGLE SENSOR OBSERVATION

Every robot of the team is fitted with a catadioptric omnidirectional vision system [18]. Every omnidirectional sensor mounts a mirror with a different profile, especially tailored for the task of the robot [25], Fig. 4. The assumptions are: the omnidirectional vision sensor is calibrated and the objects are assumed to lie on the floor. In Fig. 5, we sketched the *Perception Module* implemented inside our robots. The omnidirectional image is the input, on the left, of the image processing module, called *VA Module* (Vision Agent Module). The result of the image processing is sent to the so-called *Scene Module*, where all measurements are transformed in the common frame of reference of the field of play using the inputs of the encoders and of the localization module. The measures in the common frame of reference are sent to the other robots and to the *Distributed Vision Module* (DV), where they are fused with the measures received by the teammates.

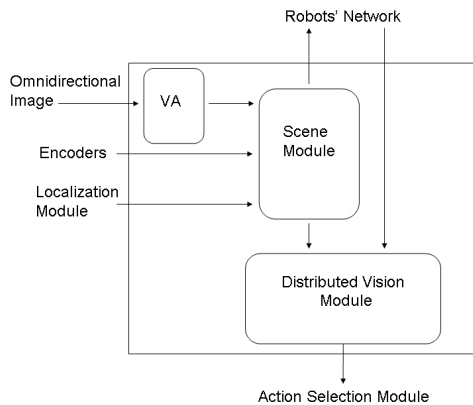


Fig. 5. Inside the *Perception Module* of each robot an input image is processed by the Vision Agent (VA) and the result is passed to the *Scene Module*. Then, the measures are sent to the other robots and to the DV

A description of the scene in the frame of reference of the field of play (i.e., the positions and speeds of the objects of interest) is reconstructed here using the data coming from the encoders and by the localization system. The measurements on the positions and velocities of the objects are then passed to the DV and broadcast to the other robots. Fig. 3 shows an example of the result of the image processing on one omnidirectional image taken in the RoboCup field of play. The strong distortion of the image is due to the custom profile of the mirror shown in Fig. 4. This is a three parts mirror, where the convex outer section is looking close to the body of the robot with a high resolution. This part produces the outer ring in the image (the one containing the single ring of dots and the field lines at high resolution in Fig. 3). The inner part presents a discontinuity in the vertex, so the self-reflection of the robot body does not appear in the picture. The blue goal (on the top of the image), the yellow goal (on the bottom) and the ball (in the middle-right) have been detected. The detection of other robots has not been enabled to not confuse the image.

A two dimensional Gaussian probability distribution is associated to every measurement. The centroid of the Gaussian is located at the estimated object position. The widths of the Gaussian along the principal axes ( $\sigma_r$ ,  $\sigma_\theta$ ) correspond to the uncertainty of the observation along those axes. Every measure is made in the reference frame of the robot and is then transformed into the reference frame of the field of play by the DV. This assumes that the robot knows perfectly its pose in the environment while it moves in the field of play. This is done using the self-localisation algorithm developed by our RoboCup team [28]. Due to the robustness of the self-localization algorithm, the assumption of an error-free localization is acceptable, even when the robots move in the playing field. The remaining localization error is taken into account by overestimating the error associated to the single measurements. We determined experimentally with 1000 measurements the width of the Gaussian along the two major axes, i.e. along the radial direction robot-object ( $\sigma_r$ ) and along the line orthogonal

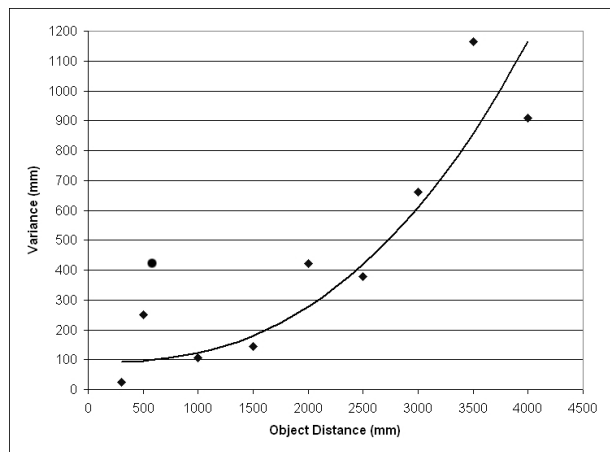


Fig. 6. The plot of the variances associated to distance measurements. The axis of abscissas represents the measured distance of an object from the robot (in mm). The ordinate represents the variance associated to the distance measure (in mm).

to this direction ( $\sigma_\theta$ ). The plot representing the measured variance  $\sigma_r$  for Robot 1 is displayed in Fig. 6. The data of the variance was interpolated with the curve of Eq. 1 and we obtained a function that associates to every measurement the correct variance along the radial axis<sup>2</sup>. As can be seen from the plot, the error increases more than linearly with the distance from the robot. This is because, with the mirror profile used in this experiment, the image resolution decreases with the distance from the robot. This procedure was repeated for each robot, since they mount heterogeneous vision systems and thus provide measurements with different accuracies.

$$y(x) = kx^a + q \quad (1)$$

Only the plot of the data about the distance object-robot is displayed, since the variance on the azimuth resulted to be so small that one could assume a zero error on azimuth. The zero value is not a valid one, though, as it would result in a degenerated Gaussian distribution. We therefore assumed a certain non-zero variance, increasing with the distance from the robot. This will also take into account the errors introduced by a non-perfect localization of the robot.

It should be noted that the objects observed by the robots are moving, not static. Assumptions about the time interval between two measurements cannot be made; in fact, we are working with robots with very different computational power, with vision systems working at different frame rates (from 10 fps to 25 fps). Even within the same robot, one cannot make any time assumptions regarding the timing of the measurements, as these are not guaranteed to be delivered at regular intervals. In fact, to fully exploit the low computational resources of our robots, we use a thread-scheduling system which allows a certain flexibility to the execution time of the threads, so measurements are made available at different time intervals (the typical value of variations in the image processing time in our robots is about  $\pm 20ms$ ). It is therefore

<sup>2</sup>For Robot 1 the constants of Eq. 1 are  $k = 0.0000009$ ,  $a = 2.52$ ,  $q = 90mm$ .

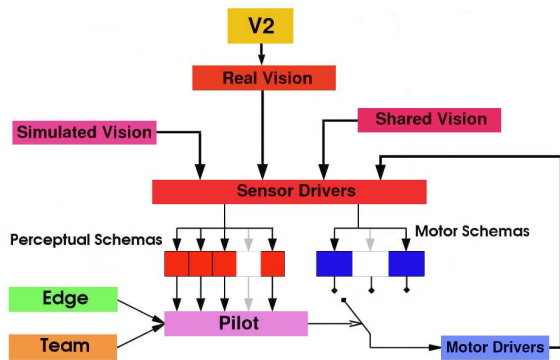


Fig. 7. The functional architecture of the governing unit for each single member of the Artisti Veneti robot team.

necessary to associate a precise timestamp to each and every measurement. The timestamp is that of the omnidirectional image that is processed to extract the measurements. As we will explain in the following, the measurements made of the position and speed of one particular object are not considered independent and are fused in a track that is used to increase the robustness of the observations.

#### IV. BUILDING A HYBRID ARCHITECTURE FOR COORDINATION

As previously stated, a schema is the building block of our architecture where perceptual components are organized into a hierarchy of abstraction levels. They feed motor schemas acting as either a control mechanism or a delivery device towards robot effectors, namely, the *wheel-driving motors* and the *kicker*. At the reactive level (see Fig. 2) schemas are true behaviors whereas at higher levels they work as triggering mechanisms to modulate the whole behavior of each individual. The actual implementation rearranges perceptual schemas in a network of and-or nodes, generated at start-up by executing appropriate scripts describing that hierarchy, and can be easily changed.

##### A. Integrating Deliberation

A pure reactive level would fail to provide a robot team with the required cooperation capabilities because of the lack of some sort of mechanism which allows the behavior of each individual robot to take into account the behavior of other robots. Generally, individual robot behaviors are triggered by *coordination* in such a way that some *actions that are a part of an agent's own goal-achieving behavior repertoire, but have effects in the world, help other agents to achieve their goals* [24]. Even a coordinated behavior among a group of robots based only on some *stigmergic*<sup>3</sup> property, could fail to exhibit collective behaviors, because stigmergy in itself does not guarantee cooperation.

The problem could thus be stated as follows: *how much deliberation should be implemented between agents to ensure the emergence of cooperative behavior?* The solution to the

<sup>3</sup>This term, commonly used in biological literature, refers to the animal capabilities to coordinate without explicit communication.

more general problem of making a collective behavior emerge from the individual behaviors of a group of robots depends on two different conditions that must be true at the same time. The first concerns the ability of any robot to recognize the circumstances under which it can be engaged in a collective behavior. The second requires that those circumstances become effective, to allow the group of robots to cooperate; the question of how to trigger, at the abstract level, the overall performance of the group while trying to exhibit a collective behavior has been previously elucidated [15]. Integrating deliberation within a behavior-based architecture is a current topic of research actively debated [16] because the *reactive/deliberative* trade-off depends on how many representational issues are implemented and how much reasoning process is made available to the system.

Since any deliberative process slows down the *decide-sense-adapt behavior cycle*, different priorities should be assigned to the different layers shown in Fig. 2. In the hybrid multi-level architecture that we have devised for our robot team (see Fig. 7), two intermediate levels have been provided to allow robot individuals to communicate. The lower implements stigmergy, whereas the higher deals with the dynamic role exchange, needed if we want an effective control on cooperation to be triggered by internal and external firing conditions.

##### B. Implementing Coordination

As previously stated, coordination has been implemented at two stages: the lower, dealing with the reactive level, provides the necessary conditions to be verified to start an activation cycle of cooperation. Such conditions are evaluated by acquiring information from the environment and testing for specified patterns. If we are looking for a better performance inside the group then role assignment is needed in what it can trigger the emergence of the required collective behavior.

Hence, though the general problem of coordinating a group of robots could be stated as an *optimization problem*, the solution can be searched heuristically simply adding some behavioral rules with the aim to force the activation of *complementary behaviors* within different teammates. Such a solution cannot be considered optimal, but the experimental evidence shows how this approach can address the problem. We have, therefore, added a higher layer, devoted to examine and schedule the behaviors which are the best candidates for cooperation. It uses a general but simple protocol to allow robots to assign proper roles, as explained below.

When an individual robot succeeds in recognizing a distinguishing configuration pattern in the environment, it tries to become a *master* of a collective action indexed by that pattern. This can occur because at reactive level some stigmergic condition forces the estimation of a given *utility function* to evaluate over a fixed threshold. Of course, different individual robots could evaluate positively the same stigmergic condition. Therefore, we have introduced a simple but effective method to acquire a master role on the basis of the *temporal ordering by which individuals try to notify the other teammates also wishing to become master*. In the case two robots try to simultaneously advocate a master role, and the utility functions

computed by both robots give the same value, then a random selection is done to choose which one must be the master.

Roles are played at different levels; let us call them *canbe*, *assume*, *acquire* and *advocate* where the first three refer to a *supporter* and the last one is committed to the master.

As an example, we will discuss the coordination task between two robots which try to carry the ball towards the opponent goal, passing and eventually defending it from opponents' attacks. A number of conditions must be continuously tested if we want such a cooperative task to become effective. Because the two robots are required to play well-specified roles, we assign the *master* role to the robot chasing the ball, whereas the other can be considered the *supporter*. We can then build a couple of complex behaviors for the master robot and for the supporter robot. Thanks to the *clampmaster* behavior, one robot is able to acquire and then to advocate a master role, showing a dominant role in the clamp action by chasing the ball. The other robot instead is committed to acquire a supporting role in the clamp action while it is approaching the ball.

```

behavior clampmaster
  haveBall(me)&¬haveBall(mate)
    → acquire(Master);
  acquire(Master)&Notify(Master)
    → advocate(Master);
behavior clampsupporter
  ¬acquire(Master)&canBe(Supporter)
    → assume(Supporter);
  assume(Supporter)&Notify(Supporter)
    → acquire(Supporter);

```

Since the role assignment depends on *ball possess*, we can use the condition *haveBall* to discriminate among robots which one is really carrying the ball. It can be understood as a *macroparameter*, in the style of our previous work [32], describing the characteristic of the environment, which can be evaluated by different robots. Moreover, it resynchronizes the activation of a new cooperation pattern.

*Clampmaster* and *Clampsupporter* are complementary behaviors that must be arbitrated. The basic rule is that a *master* role must be *advocated* whereas the *supporter* role should be *acquired*. To this aim, we require two reciprocity rules where a role is switched either from *acquire* to *advocate* or from *assume* to *acquire* provided that a *notification* is made to the referred teammate. Such rules imply a direct communication between teammates to assign the role on the *first notified/first advocated* basis. In this way, the robot carrying the ball advocates the master role for itself and commits the teammate to acquire the supporter role. By doing so, the former robot issues a behavior of *chaseBall* whereas the latter exhibits a behavior of *approachBall*.

### C. RoboCup MSL teams coordination policies

Starting from Stone and Veloso's pioneering work [36], since the beginning of RoboCup many teams have implemented some kind of role-based coordination. The first attempts were made on a *static basis*, where each robot takes a

fixed role within the team, but if we consider the *integrating* robot societies [35], characterized by a small number of heterogeneous and specialized members, it becomes important for each individual to develop the ability of modifying dynamically its behavior while performing an assigned task. Thus, a *dynamic role assignment* capability has become a key issue.

It can be stated as follow. *Given  $n$  robots,  $n$  prioritized single-robot roles, and some estimation of how well each robot is expected to play each role, assign robots to roles in order to maximize the overall expected performance* [15]. Role allocation in an MRS is a dynamic decision problem, changing over the time, according to the environmental changes. Gerkey and Mataric [15] showed that the role allocation problem in the RoboCup domain is similar to the task allocation problem for MRS that cooperatively achieve a goal, where a time-extended role concept replaces that of a transient task. They developed a formal discussion on role allocation issue in RoboCup, by showing that many allocation mechanisms are algorithmically equivalent to instances of the canonical greedy algorithm for the Optimal Assignment problem. Thus the dynamic role assignment problem is the natural evolution of an iterated assignment problem in the domain of multi-robot systems. However, MSL RoboCup teams used hand-coded methods, where the current allocation is re-evaluated periodically a few times per second.

ART Team [30], the ancestor of Artisti Veneti Team [34], ordered the roles in a descending priority, and then assigned each role to the available robot with the highest *utility* function [17]. Utility is a scalar quantity that estimates the cost of executing an action. Its value is a weighted sum of factors such as the distance from target or from the ball, defense-offense configurations, etc. It should be noted that such computations are always affected by sensor noise, general uncertainties, and environmental changes. A slightly different solution was adopted by CS Friburg Team [38] which used a distributed role allocation mechanism where two robots may exchange roles only if both agree, thus increasing their own utility values. Isocrob team follows a popular policy of distributing the sensor information among robots. In the ICRA2004 [23] Video Proceedings, its coordination policy is illustrated. Each robot team member achieves a local estimate of the environment and shares its evaluation through sensor fusion for distributing dynamically the suitable roles among the robots. Our Team, Artisti Veneti [34], at the IAS-Lab of the University of Padua, has developed its own control architecture starting from a behavior-based hand-coded software. At beginning, the choice of the appropriate role was obtained by considering collision avoidance issues and competitive behaviors [33]. Then, we have introduced a hybrid architecture [10], [34], where the deliberative component interacts with the reactive one and viceversa, as it has been described in the previous subsections,

Other teams used different concepts like the social benefit [14], or scheduling policies based on Petri Nets [40]. Eigen Team of Keio University, a RoboCup-MSL World-Champion on 2002, 2004 and 2005, achieves a cooperative ability through a continuous exchange of information among the team members. The evaluation of task achievement by the team is done

by each individual robot with regards to an individual-social satisfaction, that compares the evaluation of the achievement of its own particular sub-task with the evaluation of the achievement of the whole task done by all the other team members[13]. CoPs Stuttgart uses a special and simplified form of an Interaction Net to evolve team strategies and agent behavior fast and efficiently[40]. The robots are able to achieve a high degree of autonomy in deciding which strategy to use. They succeeded in showing a pass-play performance.

## V. FUSING MULTIPLE OBSERVATIONS

The measures of the position and speed of the tracked objects can come from two sources: the repeated observations of the single robot or the observations of the teammates. The work illustrated in [37], authors do not take into account the dynamics of the objects to be tracked and assume that the objects are instantaneously steady. They use a minimal variance estimation approach to fuse the measurements of the different robots assuming the measurements are made at the same instant. In the real world, measurements of different robots are never made at the same time and because the objects are moving, every robot will measure the object when it is in a different position. They solved the problem by discarding measurements too distant in time to be compatible.

On the contrary, the solution we adopted is similar to the one proposed in [11], with a significant difference: whereas they used the external *Global Sensor Integrator* to fuse the information, in our implementation every robot fuses all the received measurements. Every time a new measurement is received, independently of whether it comes from itself or from another robot, it is compared with the existing tracks of the objects. If it is compatible with an existing track, the measurement is added, otherwise a new track is initialized.

This is done with a classical Kalman filter for every track [5], [7], [8]. Every incoming measurement can reduce the variance of the Gaussian distribution, reducing the uncertainty on the position of the object, as detailed in [37]. This approach also allows the storage of multiple tracks for a single object; i.e., the creation of multi-modal distributions for every object. The real position of the object is decided to be the one with smallest variance, i.e., the one with smaller uncertainty. This solves also the problem of dealing with multiple instances of the same object. For instance, if for some reasons two balls are on the field of play, every robot will instantiate two tracks for the ball and will consider as the 'real' ball, the one with the smallest variance (probably the ball closer to it).

In the classical Kalman filtering approach, it is implicit that measurements arrive one after the other. In the next section we will see how we modified this classical approach to take into account the fact that measurements come from different robots and that we want to be able to accept measurements older than the last one (usually, only newer measurements are taken in account).

### A. Fusing Observations from Different Robots

Our system is designed to be totally independent from the number of robots active on the field. Every robot uses all the

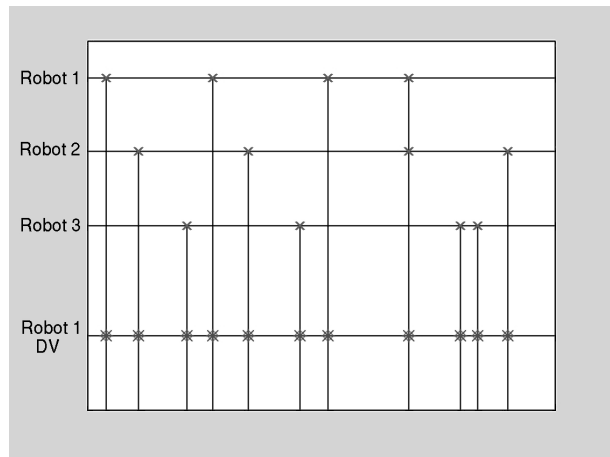


Fig. 8. A conceptual sketch of the fusion of measurements coming from different robots in the Distributed Vision module (DV) of Robot 1. Note that Robot 1 is measuring the object's position only 4 times, while the object's position is updated 11 times in this time interval.

measurements available, independently from the number of teammates. This is done to make the system robust to failures of single robots.

Fig. 8 depicts a conceptual sketch of the process of fusing measurements coming from different robots. Every time that the DV of Robot 1 receives a new measure from Robot 2 or Robot 3, it fuses this measure with its own measures. In Fig. 8, Robot 1 is measuring the position of the ball only 4 times, but the ball position estimate is updated 11 times in its DV. This means that the robot has a more reliable world model. However, there are several problems to solve when fusing measurements from different robots.

The first problem is that, in order to combine the different observations, all the robots must share the same spatio-temporal frame of reference. It is not enough to be able to refer all the measurements made in the spatial frame of reference of a single robot to the common spatial frame of reference of the field. The robots need also to be able to refer the time stamp associated to every measurement to a common temporal frame of reference. In other words, the internal clock of the robots needs to be precisely synchronized in order to know the time relation between the different measurements. The problem of the synchronization of the internal clocks of the single robots was solved using the well-known Network Time Protocol, developed by the Network Time Protocol Project<sup>4</sup>. Any robot can act as a server for synchronizing the clocks of the others.

A second problem is that when an agent is cooperating with other agents, it needs to trust the other agents. The amount of confidence in the measurements of the others influences the amount of cooperation. We made every robot more confident on its own measurements than on measurements received from the teammates. This is done by doubling the variances associated to the measures received from the teammates. This implies also that the single robot measurements are less affected by errors introduced by the teammates, due

<sup>4</sup>URL: <http://www.eecis.udel.edu/~ntp>

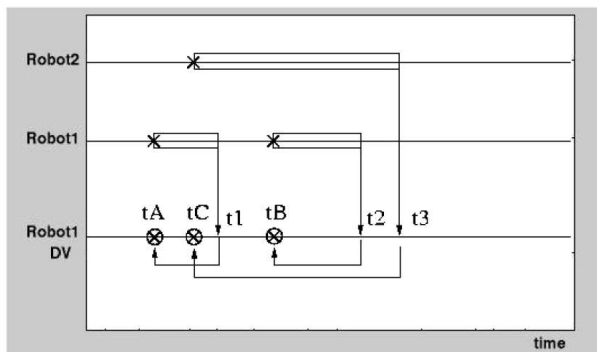


Fig. 9. A conceptual time diagram showing the process of fusion of measures older than the last one received. For our robots the image processing time is 40 ms for Robot 1 and 70 ms for Robot 2.

either to non-precise localization or to measurement errors, like for example in the case the errors in the perception of the ball position and the different weight of the information communicated by team-mates could make robots to have rather different positions of the ball. Thus, the robots can show a certain degree of coordination even in absence of perfect localization and coherent "map merging".

A third problem emerging when working with heterogeneous vision systems running at different speeds is that the measurements arrive at different instants in time. When a robot receives, from another robot, a measurement that is older than its own measurements, it cannot simply discard that measurement. Often this measurement can be carrying useful information even if it is old. As a practical example, consider the situation of a robot that is close to the object, but has a very slow vision system. Given its proximity to the object, this robot will report very accurate measurements which can improve the estimate generated by a robot with a faster, but less precise, vision system.

The solution we adopted is conceptually outlined in Fig. 9. Robot 1 makes two measurements at instants  $t_A$  and  $t_B$ ; these measurements are available at instants  $t_1$  and  $t_2$ , respectively. The boxes in Fig. 9 represent the image processing time required by the robot, in our case approximately 40 ms for Robot 1 and approximately 70 ms for Robot 2). At time  $t_2$ , the state of the system is estimated by the DV of Robot 1, using the measurements made at  $t_A$  and  $t_B$ . At time  $t_3$  the DV of Robot 1 receives a measure from Robot 2 referring to an instant  $t_C$  which precedes  $t_B$ . In order to take into account this new measurement, the DV retrieves the state of the system at instant  $t_A$ , reorders the available measurements and regenerates the system state from  $t_A$  fusing all the received measures in the correct time sequence. The maximum amount of delay allowed for a measure to trigger a reappraisal of the state of the object is 250 ms. Measures older than 250 ms are considered too old to carry useful information and would imply to reconsider too much past measures.

### B. Related Approaches

The idea for information-sharing in the sensing process comes from previous work on cooperative sensing, mainly,

Stroupe et al. [37] (using perspective cameras) and Gutmann et al. [11] (using perspective cameras and laser range finders).

Besides using omnidirectional vision systems, the major extensions of our proposal, as compared to [37], is the modification of their approach so as to integrate observations made at different instants in time, while also taking into account the speed of the objects being tracked. Compared with [11], the main difference is their use of an external computer running a so-called *Global Sensor Integrator* (GSI). This GSI receives the observations made by the different robots, integrates them by elaborating a merged vision of the world, and sends it back to every single robot. Every single robot then uses the information received by the GSI only to locate objects out of its field of view. In our system, on the contrary, every robot fuses the information coming from the teammates without the need of an external computer. At the same time, these measures are used to improve the ones made by the robot itself. However, we believe that a robot should not fully trust the information coming from teammates, since they may be in a misleading situation. So, in the previous SubSection we proposed a way to appropriately weigh the incoming measures.

## VI. EXPERIMENTAL RESULTS

As we have previously discussed, implicit communication is a necessary and sufficient tool for activating and tailoring cooperative behaviors. The problem is how many times the interaction patterns are needed to be detected by different robots to initiate a cooperation task. In the case of simulated soccer games, we have shown [33] that a continuous evaluation of environmental patterns could trigger a ball exchange between teammates. The number of successful cooperation can be kept high by increasing the circumstances of positive activation by a kind of *Brownian motion* among teammates. The situation becomes more difficult in the case of MSL real robot competitions, where the evolving dynamics of teammates cannot provide such a satisfactory number of active interactions. Dynamic role assignment thus becomes a very important feature to be implemented into a team.

The approach described in section IV was successfully tested during the MSL Challenge Competition of RoboCup2003 held in Padua. One of these challenges required that a team of two robots would be able to show a cooperative behavior by exchanging the ball between them before scoring it into a goal that is not defended by an adversary goalkeeping. Thanks to the described approach, our robot team succeeded to show a successful performance.

As shown in Fig. 10<sup>5</sup>, we developed a cooperative behavior of two companion robots, involved in a cooperative task, which results in carrying the ball towards the opponent goal to score after having exchanged the ball between them. As a matter of fact, the robots exchange the ball by swapping the role.

Fig. 10 shows two different robots, *A* and *B*. The first is chasing the ball, while its companion is approaching to protect it. The *approachBall* behavior shown by the latter is a consequence of the exchange of a low number of short messages and which result in the assignment of the *master*

<sup>5</sup>The full video is available at <http://www.dei.unipd.it/~robocup/video/tenaglia.avi>



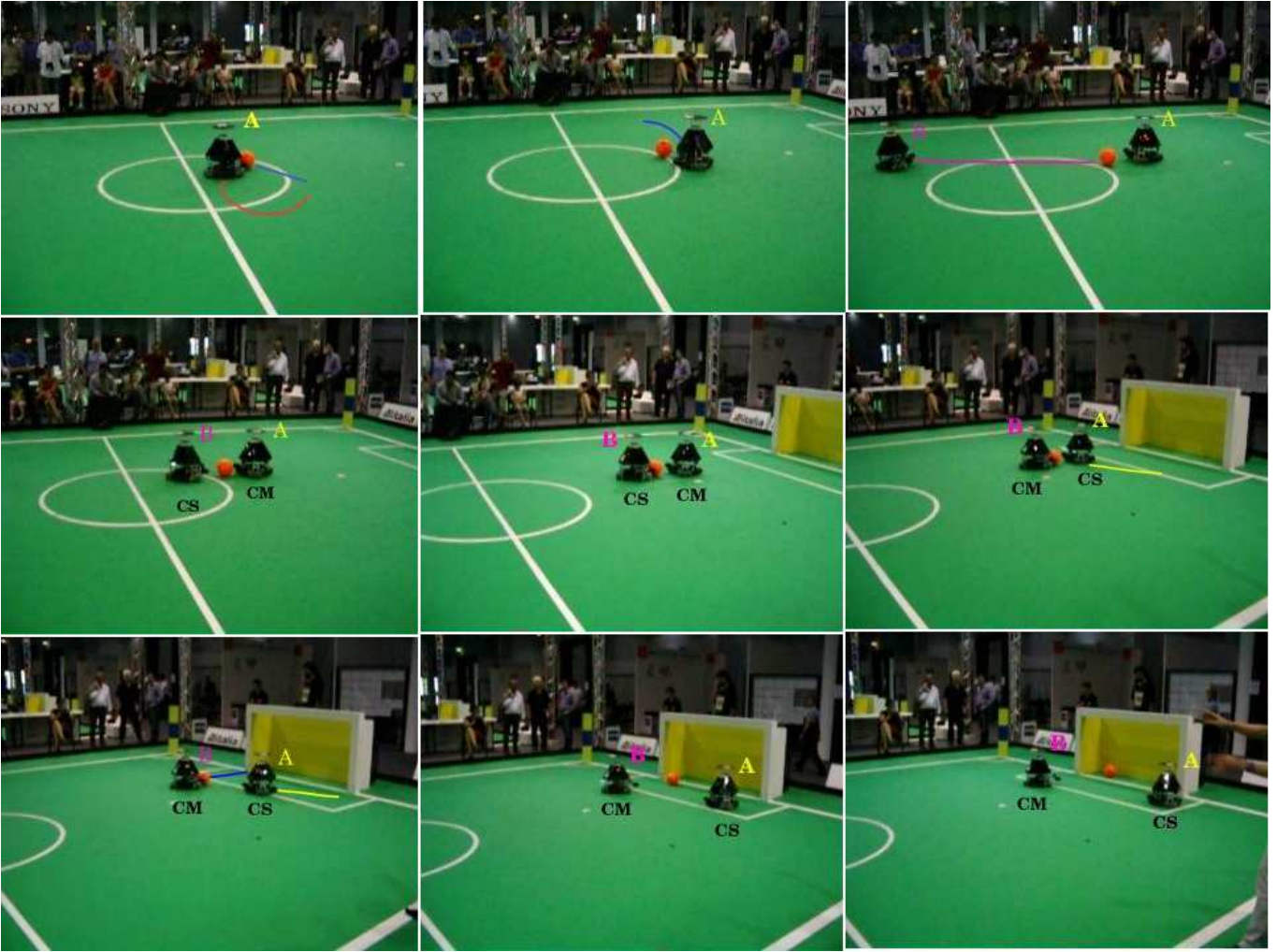


Fig. 10. Two attacking robots of the Artisti Veneti team show their ability in exchanging the ball during a clamp action at the MSL Challenge of RoboCup-2003 International Competition held in Padua, on July 2003.

role to *A* and the *supporter* role to *B*; this is indicated by the labels *CM* and *CS* in the figure. The two robots are moving in a strictly coordinate manner. The roles can be swapped, depending on the actual environmental conditions.

As the two robots approach the goal line, they swap the roles, and in such a way they exchange the ball because they have evaluated which one can score more easily. In the figure, the role labels are swapped to indicate how the roles have actually been exchanged between the two robots. The resulting emergent *exchangeBall* behavior emphasizes the aptitude of soccer robots to activate a cooperative cycle of actions.

Regarding implementation, all schemas are executed as threads in ADE, a runtime environment especially designed for real-time systems over a Unix/Linux kernel [34]. Also the arbitration module is executed as a thread; more precisely, three different threads have been committed to select a behavior for its execution.

Looking at Fig. 7, it can be easily understood how the governing unit operates to control robot behaviors. First of all, sensor information, coming from different sources, is piped towards the *sensor drivers* which work as input controllers. They provide all perceptual schemas with the required sens-

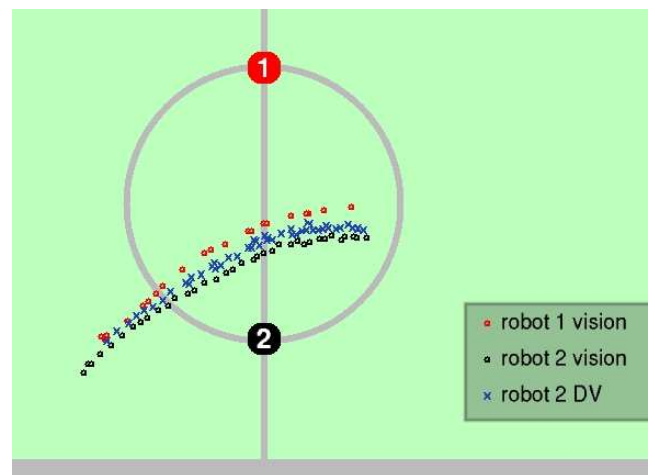


Fig. 11. A screenshot of the Omnidirectional Distributed Vision System visualization software with two robots steady and the ball moving. The red circles are the subsequent positions of the ball measured by Robot 1. The black circles are the positions of the ball measured by Robot 2. The blue crosses are the ball positions reconstructed by the Distribute Vision Module of Robot 2.

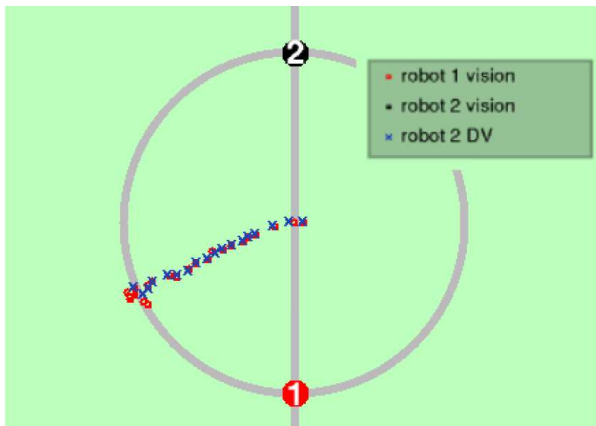


Fig. 12. In this experiment, Robot 2 cannot see the ball. Nevertheless, it is able to locate it by using the measures received from Robot 1 (the red circles) processed by its Distributed Vision Module (the blue crosses).

ing, also feeding the C-implemented motor schemas which demand immediate sensor data for triggering. The modules labelled *pilot*, *Edge* and *Team*, implemented as threads, are committed to select the most suitable motor schema to gain exclusive control of the robot. The thread *pilot* evaluates all the possible activating conditions. The *Edge* module affects robot behaviors from the external environment by adapting their execution to the constraints which stem from soccer play rules like for example avoiding, as far as possible, violating situations. Finally, the *Team* module provides the necessary coordination that a single teammate must exhibit to enable a collective behavior, like for example *passing the ball*, to emerge from the robot team.

Regarding the Omnidirectional Distributed Vision System, we used it during the RoboCup 2004 competitions in Lisbon (Portugal). In order to have controlled and repeatable test situations, we performed a series of experiments in our laboratory, reproducing partial real game situations. In the first experiment, depicted in Fig. 11, two robots are stationary at the center of the field (see Fig. 1) and the ball passes between them. Each robot measures the ball position and speed, sends these measurements to the other, and fuses its own measures with those received from the teammate. In Fig. 11, the red and black squares represent the measurements made by Robot 1 and Robot 2, respectively; the blue crosses represent the position of the ball, as calculated by Robot 2 while also taking into account the measurements received from Robot 1. Note that the frequencies of measurements made by the two robots are different: Robot 1 is measuring at 16.5 fps (frames per second), and Robot 2 is measuring at 25 fps.

In the second experiment, depicted in Fig. 12, one robot (Robot 2) cannot see the ball (because the color threshold of its vision system for ball recognition was expressly set to wrong values; the other color thresholds are set correctly, so the robot is able to self-localize). A second robot (Robot 1) can see the ball and sends its measurements to Robot 2 (depicted by red squares). Robot 2 is thus able to correctly estimate the position of the ball (depicted by blue crosses) using the measurements received from Robot 1.

The third experiment, depicted in Fig. 13, tests the robustness of the proposed *ball kidnapping* system, described in Section 3. In this experiment, the two robots are at the two sides of the playing field (approximately 6m apart) and the ball is *kidnapped* from the position on the left, where it was stationary, to a position on the right where it starts to move. Note that the DV considers the new ball position only after receiving 4 measurements; i.e., the blue crosses appear on the right-side only after 4 measurements. This is the time interval needed by the variance of the old track to grow larger than the variance associated to the new track.

## VII. CONCLUSIONS

**I**N this paper we have tried to understand how to enhance the cooperative capability of a robot team playing in the RoboCup MSL competitions. Our current work is a direct evolution of our past experience in designing behavior arbitration which triggers and is triggered by purely *stigmergic* mechanisms, namely, implicit communication [32], [33].

Considering the inherent difficulty of forcing coordination during fast and dynamic games, we tried to achieve a cooperative task using a dynamical role assignment, switching from an implicit team assessment, to an explicit *first notified/first advocated* arbitration. Our approach has been tested during the RoboCup2003 MSL Challenge competition, where our robot team, Artisti Veneti, showed an excellent coordination capability for exchanging the ball. This is documented by a movie, available on our Web site, which was judged as the best recorded cooperative action by the selection committee of the RoboCup-2004 MSL qualification procedure.

In the same frame, we have illustrated the implementation of an Omnidirectional Distributed Vision System used to share the information needed for planning the cooperation. To deal with the multiple problems arising when a multi-robot team works in a real environment, we introduced a series of new concepts. All the robots are capable of relating all the measurements to a common spatial frame of reference. In addition, the observations of all the team members are synchronized using the Network Time Protocol. The robot team is also able to take into account the heterogeneity of its different constituent robots. The state of the system can be recalculated using old measurements (i.e., older than the last one in the track), fusing data from different robots, and taking advantage of the redundancy of observations and observers.

The experiments carried out in our laboratory validate the approach and suggest its possible applications in other robotics domains. Although in this paper we showed how to fuse data from omnidirectional vision sensors only, our approach is very general and can be applied to fuse measurements coming from different sensors, such as perspective cameras, laser range finders, or other kind of sensors, as long as these ones provide data that can be represented as Gaussian distributions. In [27] we fused both video and audio sensor data to improve the detection of intruders. Thus, our approach, originally developed within the RoboCup field, can be effectively applied to service robotics applications, such as surveillance.

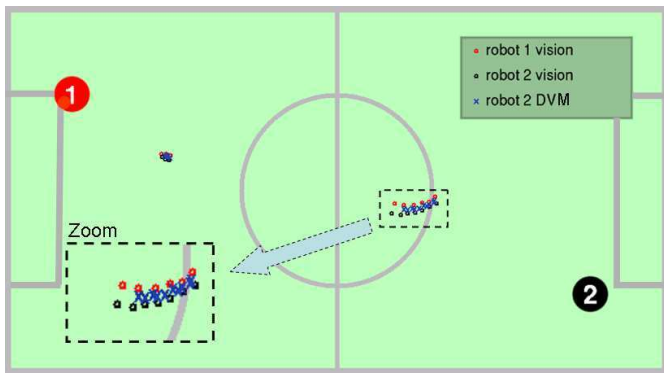


Fig. 13. In this experiment the ball is kidnapped from the position on the left, where it was stationary, to a position on the right, where it starts to move. The tracks on the right are blown-up in the dashed box.

#### ACKNOWLEDGMENTS

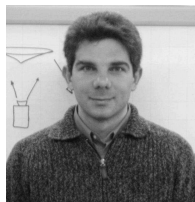
We wish to thank all the members of the past *ART Team*, active from 1998 to 2000, which provided the community where we started to develop and experiment many of the ideas illustrated in this paper. We also thank all the students who, with great enthusiasm and effort, continued the work initially started by the ART Team, carrying it into the *Artisti Veneti Team* from 2001 onwards. A special thank you goes to Alberto Scarpa who contributed to the development of the ODVS system, as well as to Enrico Ros and Arrigo Zanette, who gave the most important contribution to the design and implementation of the robot team architecture.

#### REFERENCES

- [1] M.A. Arbib. Perceptual structures and distributed motor control. In *Handbook of Physiology-The Nervous System II: Motor Control*, pages 1449–1480. 1981.
- [2] M.A. Arbib. Schema theory. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*, pages 830–834. MIT Press, Cambridge (MA), 1995.
- [3] R.C. Arkin. Motor schemas based navigation for a mobile robot. In *IEEE Conference on Robotics and Automation*, pages 264–271. 1987.
- [4] R.C. Arkin. Modeling neural function at the schema level. In *Biological Neural Networks in Invertebrate Neuroethology and Robotics*, pages 383–410. 1993.
- [5] Y. Bar-Shalom and T. Fortmann. *Tracking and Data Association*. Academic Press, 1988.
- [6] R. Brooks. A robust layered control system for a mobile robot. *IEEE Jour. of Robotics and Automation*, 2(1):14–23, 1986.
- [7] R. G. Brown and P. Y. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. Wiley & Sons, 1996.
- [8] A. Cai, T. Fukuda, and F. Arai. Information sharing among multiple robots for cooperation in cellular robotic system. In *Proc. of 1997 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 97)*, pages 1768–1773, 1997.
- [9] L. Chaimowicz, V. Kumar, and M. Campos. A paradigm for dynamic coordination of multiple robots. *Autonomous Robots*, 17:7–21, 2004.
- [10] A. D’Angelo, E. Menegatti, and E. Pagello. How a cooperative behaviour can emerge from a robot team. In R. Alami H. Asama and R. Chatila, editors, *DARS04*, pages 71–80, Toulouse (F), 23-25 June 2004.
- [11] M. Dietl, J.-S. Gutmann, and B. Nebel. Cooperative sensing in dynamic environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS’01)*, Maui, Hawaii, 2001.
- [12] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. A taxonomy for swarm robotics. *Autonomous Robots*, 3(4):375–397, 1996.
- [13] H. Fujii, D. Sakai, and K. Yoshida. Cooperative control method using evaluation information on objective achievements. In R. Alami H. Asama and R. Chatila, editors, *DARS04*, pages 201–210, Toulouse (F), June 2004.
- [14] H. Fujii, M. Kato, and K. Yoshida. Cooperative action control based on evaluating objective achievements. In I. Noda, A. Jacoff, A. Bredendfeld, and Y. Takahashi, editors, *to appear in RoboCup 2005: Robot Soccer World Cup IX*, LNAI, Osaka, 2006. Springer.
- [15] B.P. Gerkey and M.J. Mataric. On role allocation in robocup. In D. Polani A. Bonarini B. Browning and K. Yoshida, editors, *RoboCup 2003: Robot Soccer World Cup VII*. Springer, 2004.
- [16] M. Hannebauer, J. Wendler, and E. Pagello, editors. *Balancing Reactivity and Social Deliberation in Multi-Agent Systems. From RoboCup to Real-World Applications*, volume 2103 of *LNAI*. Springer, 2001.
- [17] L. Iocchi, M. Piaggio D. Nardi, and A. Sgorbissa. Distributed coordination in heterogeneous multi-robot systems. *Autonomous Robots*, 15:155–168, 2003.
- [18] H. Ishiguro. Development of low-cost compact omnidirectional vision sensors. In R. Benosman and S.B. Kang, editors, *Panoramic Vision*, chapter 3, pages 23–38. Springer, 2001.
- [19] H. Ishiguro. Distributed vision system: A perceptual information infrastructure for robot navigation. In *Proc. of the Int. Joint Conf. on Artificial Intelligence (IJCAI97)*, pages 36–43, 1997.
- [20] D. Kurabayashi. Toward realization of collective intelligence and emergent robotics. In *IEEE Int. Conf. on Sys., Man and Cyb. (IEEE/SMC)*, pages 748–753. Tokyo, Oct. 12-15 1999.
- [21] J. H. Lee, N. Ando, T. Yakushi, K. Nakajima, T. Kagoshima, and H. Hashimoto. Applying intelligent space to warehouse - the first step of intelligent space project. In *In IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, July 2001.
- [22] J. J. Leonard and H. J. S. Feder. Decoupled stochastic mapping. *IEEE Journal of Oceanic Engineering*, pages 561–571, October 2001.
- [23] P. Lima. Dynamics role exchange and cooperative object location for soccer robots. In *IEEE/ICRA-2004 Video Proceedings, CD-ROM*. 2004.
- [24] M. Mataric. Issues and approaches in the design of collective autonomous agents. *Robotics and Autonomous Systems*, 16(2-4):321–331, 1995.
- [25] E. Menegatti, F. Nori, E. Pagello, C. Pellizzari, and D. Spagnoli. Designing an omnidirectional vision system for a goalkeeper robot. In A. Birk, S. Coradeschi, and S. Tadokoro, editors, *RoboCup-2001: Robot Soccer World Cup V*, L. N. on A. I, pages 78–87. Springer, 2002.
- [26] E. Menegatti, A. Scarpa, D. Massarin, E. Ros, and E. Pagello. Omnidirectional distributed vision system for a team of heterogeneous robots. In *Proc. of IEEE Workshop on Omnidirectional Vision (Omnivis’03), in the CD-ROM of Computer Vision and Pattern Recognition (CVPR 2003)*, pages On CD-ROM only, June 2003.
- [27] E. Menegatti, E. Mumolo, M. Nolic, and E. Pagello. A surveillance system based on audio and video sensory agents cooperating with a mobile robot. In F. Groen, N. Amato, A. Bonarini, E. Yoshida, and B. Krose, editors, *Intelligent Autonomous Systems 8*, pages 335–343, Amsterdam, 2004. IOS Press.
- [28] E. Menegatti, A. Pretto, and E. Pagello. Testing omnidirectional vision-based Monte-Carlo localization under occlusion. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS04)*, pages 2487–2494, September 2004.
- [29] A. Nakazawa, H. Kato, S. Hiura, and S. Inokuchi. Tracking multiple people using distributed vision systems. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA2002)*, pages 2974–2981, May 2002.
- [30] D. Nardi et al. Art-99: Azzurra robot team. In E. Pagello M. Veloso and H. Kitano, editors, *RoboCup 1999: Robot Soccer World Cup III*, number 1856 in *Lecture Notes in Artificial Intelligence*. Springer, 2000.
- [31] E. Pagello C. Ferrari A. D’Angelo F. Montesello. Intelligent multirobot systems performing cooperative tasks. In *Special Session on Emergent Systems - Challenge for New System Paradigm, IEEE/SMC*, pages 754–760. Tokyo, Oct. 12-15 1999.
- [32] E. Pagello, A. D’Angelo, F. Montesello, F. Garelli, and C. Ferrari. Cooperative behaviors in multi-robot systems through implicit communication. *Robotics and Autonomous Systems*, 29(1):65–77, 1999.
- [33] E. Pagello, A. D’Angelo, C. Ferrari, R. Polesel, R. Rosati, and A. Speranzon. Emergent behaviors of a robot team performing cooperative tasks. *Advanced Robotics*, 17(1):3–19, 2003.
- [34] E. Pagello, E. Menegatti, A. Allegro, E. Avventi, N. Bellotto, T. Guseo, F. Favaro, M. Pasquotti, A. Pretto, E. Ros, A. Scarpa, A. Zanette, and A. D’Angelo. Enhancing the artisti veneti team for robocup 2003. In D. Polani, A. Bonarini, B. Browning, and K. Yoshida, editors, *Team*

*Description Papers Collection, CD-ROM attached to RoboCup-2003: Robot Soccer World Cup VII*, number 2377 in LNAI. Springer, 2004.

- [35] L. Parker. From social animals to teams of cooperating robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS'97)*, Workshop on Multirobot cooperation: Current trends and key issues. Grenoble (France), 1997.
- [36] P. Stone, M. Veloso. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. *Artificial Intelligence*, 110(2):241–273, 1999.
- [37] A. W. Stroupe, M. C. Martin, and T. Balch. Distributed sensor fusion for object position estimate by multi-robot systems. In *Proceedings of International Conference on Robotics and Automation (ICRA2001)*, May 2001.
- [38] T. Weigel, J.S. Gutmann, M. Dietl, A. Kleiner, and B. Nebel. Cs-freiburg: Coordinating robots for successful soccer playing. *IEEE Transactions on Robotics and Automation*, 18(5):685–699, 2002.
- [39] Y. Yagi. Omni directional sensing and its applications. *IEICE TRANS. INF. & SYST.*, VOL. E82-D(NO. 3):pp. 568–579, MARCH 1999.
- [40] O. Zweigle et al. Cooperative agent behavior based on special interaction nets. In T. Arai, R. Pfeifer, T. Balch, and H. Yokoi, editors, *Intelligent Autonomous Systems 9*. IOS Press, 2006.



**Emanuele Menegatti** Emanuele Menegatti received his "Laurea" in Physics at the University of Padua in 1998. He got an MSc in Artificial Intelligence from the University of Edinburgh (U.K.) in 2000, and a Ph.D. in Computer Science from the University of Padua in 2002. From 2003 to 2004 he was a Post-Doc Researcher at the Intelligent Autonomous Systems Laboratory (IAS-Lab) of the University of Padua where he is now an Assistant Professor of Computer Science. In 2004, he was a visiting researcher for some months at the Laboratory of

Prof. H. Ishiguro at Osaka University (Japan), and the Laboratory of Prof. Frank Dellaert at the Georgia Institute of Technology. His research interests are in the field of Robot Vision, particularly omnidirectional vision, to develop a Distributed Vision Systems for Autonomous Surveillance in which mobile robots can cooperate with static sensors.

**Enrico Pagello** Enrico Pagello received his "Laurea" on Electronic Engineering at the University of Padua (Italy) in 1973. From 1973 to 1983, he was a Research Associate at the Inst. of Biomedical Engineering of the National Research Council of Italy, where he is now a part-time collaborator. Since 1983 he belongs to the Faculty of Engineering of the University of Padua, where he is Professor of Computer Science at the Dept of Information Engineering. During 1977-78, he was a Visiting Scholar at the Lab. of Artificial Intelligence of Stanford University.



Since 1994, he has regularly visited the Dept of Precision Engineering of the Univ. of Tokyo, in the context of a joint scientific agreement between the Padua and Tokyo Universities and of the JSPS Senior Fellow Program. He was a General Chair of the Sixth Int. Conf. on Intelligent Autonomous Systems in July 2000, and a General Chairman of RoboCup-2003, in July 2003. He has been a member of the Editorial Board of the *IEEE/Trans. on Rob. and Aut.*, and he is currently a member of the Editorial Board of *RAS International Journal*. He is a President of the Intelligent Autonomous Systems International Society. His current research interests concern the application of Artificial Intelligence to Robotics with particular regards to the Multi-robot Systems domain.



**Antonio D'Angelo** Antonio D'Angelo is Assistant Professor of Robotics and Operating Systems at the University of Udine. He received the M.S. on Electrical Engineering from Padua University in 1981 and since 1984 he has been working at the Laboratory of Artificial Intelligence and Robotics at the Department of Mathematics and Computer Science at the University of Udine. His current research interests cover multiagent autonomous system coordination, behaviour-based planning and control including complex dynamical system models for

autonomous robots. In this framework he developed the robotic model to gain insight over coordination without communication.