A framework for robust sensing in multi–agent systems

Andrea Bonarini[1], Matteo Matteucci[1], and Marcello Restelli[1]

Department of Electronics and Information
Politecnico di Milano, Milan, Italy
{bonarini,matteucc,restelli}@elet.polimi.it

Abstract. We present the framework we have adopted to implement robust sensing in the Milan Robocup F–2000 Team. The main issue concerns the definition of symbolic models both for the single agent and for the whole multi–agent system. Information about physical objects is collected by intelligent sensors and anchored to symbolic concepts. These are used both to control the robots through a behavior–based system, and to improve the global knowledge about the environment.

1 Introduction

We present a framework for robust sensing in a multi–agent system (MAS), based on the cognitive reference model presented in [1]. We have implemented it in our F–2000 Robocup team, obtaining a versatile MAS, able to adapt to different perceptual and communication situations; the team exploit the effectiveness of knowledge sharing, when possible, and shows graceful degradation when each agent should operate on its own. We adopt a behavior–based control architecture, but we feed behaviors with symbolic concepts (implemented by fuzzy predicates) represented in the environment model we are presenting . In this paper, we present the reference framework we have defined to implement the *world modeler* MAP (*Map Anchors Percepts*), used to maintain a consistent instance of the model by anchoring [7] symbolic concepts to physical objects. We have adopted MAP not only in Robocup, but also in other embodied agents applications.

2 Sensorial integration in local maps

The world model we use in MAP considers not only geometrical features, but an integrated representation of objects considering their geometrical, dynamical and perceivable information (e.g., color and odor). The knowledge representation model we propose is based on the notion of *concept* and its *properties*. A property is a tuple

$$p \triangleq < label, \mathbb{D}, \rho >, \tag{1}$$

where *label* denotes the property, $\mathbb{D}$ is the set of all the possible values for that property given a specific representation code (e.g., for the colors we can use the set $\{red, green, blue, \ldots\}$ or the RGB space $\mathbb{N}^3_{[0,255]}$) and ρ represents a restriction of the domain $\mathbb{D}$ for that property in the specific concept.

A set of properties describes a *concept* C, which is used in our model to represent the knowledge about perceptual images of physical objects. Depending on the concept and on the specific domain, a property can be classified as *substantial* or *accidental*, respectively $\mathtt{S}$ and $\mathtt{A}$ in equation 2.

$$C \triangleq \{< p, \mathtt{x} >\} : \ \mathtt{x} \in \{\mathtt{S}, \mathtt{A}\}. \tag{2}$$

Substantial properties characterize the immutable part of a concept; for a given object, their values do not change over time, and they can be used for object recognition since they define the essence of the object they represent. Accidental properties are those properties that do not characterize a concept; their values for the specific instance can vary over time, they cannot be used for object recognition, but are the basis of instance formation, tracking, and model validation.

It is possible to describe the agent knowledge by using concepts. We introduce the notion of *model*: given the set of known domains $\mathcal{D}$, a model $\mathcal{M}_d$ is the set of all the concepts known by the agent referring to the specific domain $d \in \mathcal{D}$, linked by (structural and domain specific) relationships. A relationship between concepts may represent:

1. a *constraint* that must be satisfied by concept instances in order to belong to the model
2. a *function* that generates property values for a concept from property values of another (inference function)
3. a *structural constraint* to be used when reasoning about classification and uncertainty

The environment is perceived by a situated agent as a collection of concept instances. The property values for these instances are sensed by means of *intelligent sensors*, which analyze percepts and give them an interpretation at a higher level of abstraction, with the aid of basic domain knowledge.

From the instances of concepts $\overline{C}_i$ and a model $\mathcal{M}_d$ it is possible to infer new instances using relationships between concepts representing specific knowledge for the application domain. An instance of the environment model $\overline{\mathcal{M}}_E$ is the set of all concept instances either derived from the classification process or from inference on concept instances that are compatible with the relationships contained in the model itself:

$$\overline{\mathcal{M}}_E \equiv \{\overline{C} : C \in \mathcal{M}_E\}. \tag{3}$$

The *state* of the system represented by the model instance $\overline{\mathcal{M}}_E$ is the set of all values of accidental properties – time variant and not – of concept instances belonging to the model itself. The *tracking* phase of anchoring consists of maintaining in time a coherent state of $\overline{\mathcal{M}}_E$ and a correct classification of instances. In doing this, accidental properties have to be monitored during time, using state prediction techniques such as linear regression or Kalman filtering.

The model we are presenting in this section can be considered as the logical basis for anchoring, but it is also suitable for classical activities that an embodied agent has to accomplish:

- *sensor fusion*: features perceived by different sensors can be aggregated if they refer to the same object in the environment; this is done to collect as much information as possible about objects before classifying them, to avoid perceptual aliasing [8], and to reduce noise using redundancy in sensorial perception
- *self–localization*: we consider *self–localization* as the process of instantiating the environment model, thus obtaining $\overline{\mathcal{M}}_E$. This definition is a generalization of the common notion of self–localization [3] since it enables reasoning about the own position not only in terms of a geometrical model, but also in terms of more general knowledge (features)
- *virtual sensing*: the instantiation of a model of the environment can be used to produce new information applying *state estimation* techniques to knowledge about the model. This new information can be seen by the agent as new *virtual* features produced by sensors looking at the model of the environment instead than considering the environment itself.

3 Extension to MAS

So far, we have dealt with world modelling processes in a single–agent architecture. It is expected that in a multi–agent context each agent could take advantage of data perceived by its teammates. Having the opportunity to combine different local representations, it is possible to build a shared viewpoint of the common environment, that we call *global representation*. In doing this, we consider that each agent shares the same ontology containing *global concepts* (GC).

The global representation builder receives as input the instances of models produced by the local processes. Each model instance contains a set of instances of concepts (e.g., wall, robot, person, etc.). The agent having those instances in its $\overline{\mathcal{M}}_E$ is the *owner* and specifies a reliability value associated to the anchoring process, considering reliability of sensors in the operating conditions, pattern matching, and so on.

The global model building process achieves fusion of concept instances by a clustering process. We define *cluster* a set of concept instances related to concepts whose extensions have a non-null intersection and "similar" values for the accidental properties, where the meaning of "similar" changes according to the property.

Two concept instances $\overline{C}_1$ and $\overline{C}_2$ can belong to the same cluster if:

1. their accidental properties are similar
2. they have a different owner
3. the respective concepts are not mutually exclusive.

For instance, in RoboCup, *robot* and *opponent* are two compatible concepts, while *opponent* and *teammate* cannot belong to the same cluster; moreover, instances of concepts like *opponent* and *goalkeeper* can belong to the same cluster since among opponents there is a goalkeeper.

A new global concept instance ($\overline{GC}$) is generated for each cluster, and its accidental properties are deduced from the accidental properties of the cluster elements by a fusion process that takes into consideration also their reliability values.

A global representation gives to the MAS some interesting qualities (that justify the flourishing of several recent works about this topic [4][6][5]):

- *robustness*: the information coming from several agents that are working together in a given environment can be referred to the same physical objects
- *extensive sensing*: a MAS is comparable to a *super-agent* able to sense and act at the same time in different places, and the agents of a MAS can be considered as virtual sensors
- *fault tolerance*: the global representation can be used to identify and correct sensorial faults
- *cooperation*: it is easier to achieve coordination by reasoning on a global model shared by the agents

It is important to point out that the global representation is not built to substitute the local representation, but to supplement this by providing for lacking information and by recovering possible errors. Each agent weights its own way of integrating the information coming from the global representation, just like it does with information coming from any sensor; for this reason we refer also to the global representation as a *virtual sensor*. In this way, we exploit both the accuracy and the autonomy supplied by the local representation and the completeness and robustness of the global one.

4 The Robocup application

We have applied the described framework to our agents operating in the Robocup F-2000 league, where a team of four robots play soccer against another four. This domain can be classified as *loosely connected* since robots cannot always perceive everything is happening on the field, because of image resolution, and partial occlusions due to other robots. Each robot maintains its local map, which provides enough information for reactive behaviors; moreover, it exchanges with teammates information aimed at the maintenance of a distributed global map. The behaviors, implemented in our behavior manager BRIAN [1], are activated by evaluating sets of fuzzy predicates which represent concept instances. When information is not available for complex activity, possibly because of poor communication, the agents can still operate with the local map. If a global map can be instantiated, our MAS architecture [2] can assign *jobs* within *schemata* to perform coordinated actions.

In this environment each robot should recognize at least the ball, other robots (possibly distinguishing teammates from opponents), goals and walls. The mentioned objects may be enough for reactive robots, with limited coordination ability, to perform selfish behaviors such as most of those seen till year 2000 in the F-2000 Robocup league: a robot is able to go to the ball, eventually bring it towards the opponent's goal, finally possibly kicking the ball to the goal. This is an important achievement, but it is possible to improve this behavior. Deliberative robots, able to plan complex cooperative behaviors, need to self-localize in order to share their position and their local maps. We use a global map also to overcome problems of limited perception. For instance, if a robot cannot see a relevant object (such as the ball) because it is covered by another robot, it may be informed about the ball position by a teammate, and act consequently. This requires that both the robots are self-localized, and that both share the concept of ball, and its instance. This virtual sensing ability gives to our robots the possibility to perform interesting operating schemata, such as ball passing or coordinated blocking. In the first, a robot can go to a suitable position to receive a passage even if it cannot see the ball; in the second, a robot may intercept an opponent, even if it cannot see the ball, while another teammate is covering the defense area, waiting for the incoming opponent.

Fig. 1. The robot in front cannot see the ball that its teammate can see.

Another situation when global anchoring is crucial is when a robot has lost its sensing capability. We have made experiments where a robot has been purposely blinded and it is localized by other robots, which inform it about its position and that of ball, goal and obstacles. Of course, the performance of the blind robot could not be optimal, but it was able to intercept the ball when playing as a goal keeper, and to kick the ball in goal when playing as an attacker. This has been implemented as a failure recovery mechanism, to maintain on the field a robot with perception problems, but also to face problems temporarily occurring in specific situations. Our more recent robots can exploit a rotating movement that in certain conditions is fast enough to make the ball disappearing for some instants from the omnidirectional image due to the low frame acquisition rate (PAL standard: 25 frames/sec). They can decide to continue the rotation relying on information present in their local map, but also to consider relevant the information possibly provided by teammates, integrated by global anchoring. As their speed reduces, they directly anchor the ball again, and may consider as less relevant the information coming from other, more distant – and thus, less reliable – virtual sensors of the MAS architecture. Some of the mentioned applications of our approach to world modelling provide information to agents that cannot access to it. Most of the functionalities rely on self-localization, and its improvement is one of the main achievement of our approach. We provide here some data showing the precision obtained in two characteristic situations: self localization of one of the robots of the team by considering information coming from all the others, and ball positioning when the ball is covered by an opponent (Figure 1).

5 Conclusion

We have discussed as anchoring symbolic concepts to physical objects can give robustness to the model of the environment where a robot operates. Reasoning with anchors and established knowledge makes it possible to obtain complex behaviors which cannot be implemented by reactive modules only. Finally, sharing anchors gives the possibility to implement a global map maintained by the MAS as a system; this map may be considered as a virtual sensor which can support local deficiencies, rising the fault tolerance capabilities of the single agent and of the whole system.

References

1. A. Bonarini, G. Invernizzi, T. H. Labella, and M. Matteucci. A fuzzy architecture to coordinate robot behaviors. *Fuzzy Sets and Systems*, Submitted.
2. A. Bonarini and M. Restelli. An architecture for multi-agent systems. *IEEE Transactions on Robotics and Automation*, Submitted.
3. J. Borenstein, H. R. Everett, and L. Feng. *Navigating Mobile Robots: Systems and Techniques*. A. K. Peters, Ltd., Wellesley, MA, 1996.
4. L. Hugues. Griunded representations for a robots team. In *in Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2248–2253, 2000.
5. D. Jung and A. Zelinsky. Grounded symbolic communication between heterogeneous cooperating robots. *Autonomous Robots*, 8(3):269–292, 2000.
6. T. Nakamura, A. Ebina, M. Imai, T. Ogasawara, and H. Ishiguro. Real-time estimating spatial configuration between multiple robots by triangle and enumeration costraints. In *in Proceedings of 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2048–2054, 2000.
7. A. Saffiotti and K. LeBlanc. Active perceptual anchoring of robot behavior in a dynamic environment. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 3796–3802, San Francisco, CA, 2000.
8. S. D. Whitehead and D. H. Ballard. Learning to perceive and act by trial and error. *Machine Learning*, 7:45–83, 1991.