Concepts for anchoring in robotics

Andrea Bonarini, Matteo Matteucci, and Marcello Restelli

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

Abstract. Anchoring is the activity an agent does to classify the objects it perceives and to track them. With the increasing abilities of mobile robots, anchoring is becoming a major research issue to make robots perform their task in unstructured and partially unknown environments. We propose a model to represent knowledge in an agent, possibly participating to a multi–agent system, showing that the anchoring problem can be successfully faced by well known AI techniques. We discuss how this model can reliably support the instantiation of concepts by aggregating percepts affected by uncertainty, sensed by several sensors, and obtained from different agents.

1 Introduction

Robotic agents operate in an environment containing other *physical objects*. These are perceived by *smart sensors* that translate some physical features acquired by a *physical sensor*, into information (*features*) represented by an internal formalism which can be understood by the other modules participating to the action selection for the agent. When this information is used to maintain a model of the environment, it is important that features referring to a physical object be collected in an internal representation of the same object, referred to as *perceptual image*. It is relevant to relate the perceptual image with a *concept* present in the agent knowledge, since this makes it possible to reason about the perceived object with categories describing the corresponding concept. An embodied agent *anchors* a physical object when it can instantiate a concept compatible with the perceived features, and maintains the relationship between such an instance and the physical object it refers to, during its activities.

Recently, the *anchoring problem* has been introduced [2][3] as a research topic relevant to implement autonomous embodied agents able to perform their tasks in unstructured environments, possibly interacting with other robots, smart sensors, and people. For non-trivial tasks, reactive behaviors (relying on signals from sensors) have to be complemented by behaviors relying on reasoning that can only be done on grounded symbols. The use of a symbolic model enables data fusion among different sensors and agents operating in related environments. Symbolic interpretations can be successfully shared, and make it possible to integrate in compact models distributed information. The anchoring problem still needs a formal definition. In the next section, we introduce a general model to represent concepts and instances, and a procedure to relate them to the corresponding physical objects through the respective perceptual image. In the third section, we discuss how our model applies to multi-agent systems, by focusing on sharing knowledge.

2 Knowledge representation and anchoring

Among the properties that can be obtained by basing anchoring on a knowledge model we mention here: *noise filtering*, *sensorial coherence*, *virtual sensing*, *consistency in time*, and *abstraction*.

The knowledge representation model we propose for anchoring 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.

Two properties p_1 and p_2 are *compatible*, $p_1 \sim p_2$, if they have the same label and domain or a mapping between the respective domains exists. A property p_1 includes p_2 if they are compatible and the restriction of domain $\rho_1(\mathbb{D})$ is included in the restriction of domain $\rho_2(\mathbb{D})$:

$$p_1 \subseteq p_2 \Leftrightarrow (p_1 \sim p_2) \wedge (\rho_1(\mathbb{D}) \subseteq \rho_2(\mathbb{D})) \tag{2}$$

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 3).

$$C \triangleq \{< p, \mathtt{x} >\} : \ \mathtt{x} \in \{\mathtt{S}, \mathtt{A}\}. \tag{3}$$

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 explain 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 they are the basis of instance formation, tracking, and model validation.

We call the set of properties for a concept i as P_i, and the partial function defining the type of property ϕ:

$$\phi : \bigcup_i (C_i \times P_i) \rightarrow \{\mathtt{A}, \mathtt{S}\}. \tag{4}$$

The *extension* $\epsilon(C)$ of the concept C is the set of all its possible instances. The *intension* $\iota(C)$ of the concept C is the set of all its substantial properties.

The fundamental structural relationships between concepts are *specialization* and *generalization*. We say that a concept C_2 specializes a concept C_1 (we denote that with $C_2 = \sigma(C_1)$) when it is defined by a superset of the C_1 properties and compatible properties are included, i.e., when:

$$(\iota(C_2) \supseteq \iota(C_1)) \wedge (\epsilon(C_2) \subseteq \epsilon(C_1)), \tag{5}$$

Concepts C_2 and C_3 that specialize C_1 do that in a *partial* way or a *total* way; we have a total specialization iff

$$\epsilon(C_2) \cup \epsilon(C_3) = \epsilon(C_1), \tag{6}$$

otherwise we have a partial specialization. Concepts C_2 and C_3 that specialize C_1 do that in an *overlapping* way or an *exclusive* way; we have an exclusive specialization iff

$$\epsilon(C_2) \cap \epsilon(C_3) = \emptyset, \tag{7}$$

otherwise we have a overlapping specialization.

Using concepts it is possible to describe the knowledge used by the agent during the anchoring process. We introduce the notion of *model*: given the set of the 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

2.1 The anchoring process

Sensors produce a description of perceptual images in terms of set of features. Each feature f is represented as a pair

$$f \triangleq < label, value >, \tag{8}$$

where *label* is the symbolic name of the property the feature refers to, and *value* is the value of the feature belonging to an appropriate set of possible values $\mathbb{D}$. This activity can be considered as the *symbol grounding* phase in the anchoring process: percepts are interpreted as symbolic features to be classified as concept instances and maintained in time.

When enough information (i.e., features) is collected for an object in the environment, it is possible to detect, by a *classification* process γ, a concept matching the perceived object and to generate an instance of it, with the related degree of reliability:

$$\gamma : \wp(\{f\}) \rightarrow \{\overline{C}\} \times \Re_{[0,1]}, \tag{9}$$

where $\wp(\{f\})$ is the power set of all the features. In doing this, we use as classification criteria the *degree of matching* of the features perceived by sensors about the object and the substantial properties of the concepts. After classification, all the substantial and accidental properties associated to the concept can be used to reason about the concept instance.

In this phase we notice how *sensor fusion*– the aggregation of features from different sensors referring to the same object in the environment – can be easily implemented using *clustering* techniques considering the distance between the features values, common subsets of features, or domain specific knowledge represented by relationships among concepts.

The matching degree among concepts and features describing an object in the environment can be computed by any *pattern matching* algorithm that takes into account that: only partial descriptions of perceptual images may be available, only substantial properties are relevant, and not all properties have the same expressive power in describing a concept. Pattern matching for a concept C is done on the intension $\iota(C)$ of the concepts, but only instances belonging to the extension $\epsilon(C)$ of the concepts can be accepted.

Pattern matching, as symbol grounding, has to deal with uncertainty, since the classification process can be affected by noisy features, partial descriptions of objects, and partial contradictions among concept properties or in relationships among instances. We call θ the function that associates a concept instance to its reliability value. Concept instances related to more intensive concepts are in general less reliable, since they require matching more features and matching problems are common in real world applications:

$$\iota(C_1) \subseteq \iota(C_2) \Rightarrow \theta(\overline{C_1}) \geq \theta(\overline{C_2}) \tag{10}$$

From the instances of concepts $\overline{C}_i$ and the model $\mathcal{M}_E$ it is possible to infer new concept instances using relationships between concepts representing specific knowledge for the application domain. We define as instance of the environment model $\overline{\mathcal{M}}_E$ 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:

$$\overline{\mathcal{M}}_E \equiv \{\overline{C} : C \in \mathcal{M}_E\}. \tag{11}$$

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.

3 Extension to Multi-Agent Systems

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 suppose 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 representation building process achieves fusion of concept instances through 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. The meaning of "similar" is given by the following partial function δ, defined only for compatible properties:

$$\delta : \rho_{p_1}(\mathbb{D}) \times \rho_{p_2}(\mathbb{D}) \mapsto \{\mathtt{true}, \mathtt{false}\}. \tag{12}$$

This function, given two values of compatible properties p_1 and p_2, returns `true` if the related concept instances can coexist in the same cluster, `false` otherwise.

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

1. their accidental properties are similar:

$$\forall p_i \in P_1, \forall p_j \in P_2 :$$
$$(p_i \sim p_j) \wedge (\phi(C_1, p_i) = \mathtt{A}) \wedge (\phi(C_2, p_j) = \mathtt{A}) \Rightarrow \delta(\overline{p_i}, \overline{p_j}) = \mathtt{true}$$

2. they have a different owner
3. the respective concepts are not mutually exclusive, i.e.,

$$\epsilon(C_1) \cap \epsilon(C_2) \neq \emptyset.$$

For instance, *person* and *man* are two compatible concepts, while *man* and *woman* cannot belong to the same cluster, since no man is a woman, but men are persons; moreover instances of concepts like *woman* and *soldier* can belong to the same cluster since some women are soldiers.

A new global concept instance ($\overline{GC}$) is extracted 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 $\overline{GC}$ can be an instance of any concept among those relative to the concept instances belonging to the cluster and its reliability is evaluated by combining their reliability values; clusters having more elements and less intensive concepts produce more reliable $\overline{GC}$s.

A global representation gives to the MAS some interesting qualities (that justify the flourishing of several recent works about this topic [4][5]): *robustness*, *extensive sensing*, *fault tolerance*, and *cooperation*

4 Conclusions

In this paper we have presented the anchoring problem and a model to face it. Although a model for anchoring has been already proposed with some applications [3][2], we propose an alternative approach to establish a general framework for anchoring, suitable for sensor fusion and multi–agent communication, too. The main advantages and novelties introduced by the proposed solution to the anchoring problem are:

- *general unifying model*: our approach puts in evidence the relationships between anchoring and classical AI notions such as symbol grounding, pattern recognition, state estimation, and clustering
- *separation of concerns*: we propose a framework for automatic instantiaton of a symbolic model of the environment from sensorial percepts given a sound knowledge formalization, thus separating the design of the sensorial apparatus from control architecture
- *integrability of domain knowledge*: the use of classic knowledge representation languages allows to consider domain knowledge deriving also from designers

We consider our model as a possible general formalization of the anchoring problem. It is also suitable for triggering special, active sensing behaviors, working as active anchoring processes [6]. We have also discussed how it can be easily applied to multi–agent systems increasing robustness, reliability, and fault tolerance. More details are given in [1].

References

1. A. Bonarini, M. Matteucci, and M. Restelli. Anchoring: do we need new solutions to an old problem or do we have old solutions for a new problem? In *Proc. of the AAAI Fall Symposium on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems*, page In press, Menlo Park, CA, 2001. AAAI Press.
2. S. Coradeschi and A. Saffiotti. Anchoring symbols to vision data by fuzzy logic. In *Lecture Notes in Computer Science*, volume 1638, pages 104–112. Springer-Verlag, Berlin, D, 1999.
3. S. Coradeschi and A. Saffiotti. Anchoring symbols to sensor data: Preliminary report. In *Proc. of AAAI*, pages 129–135, Cambridge, MA, 2000. MIT Press.
4. L. Hugues. Grounded representations for a robots team. In *Proc. of the 2000 IEEE/RSJ Int. Conf. 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. 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.