An Introduction to the Anchoring Problem

Silvia Coradeschi and Alessandro Saffiotti

Center for Applied Autonomous Sensor Systems
Dept. of Technology, Örebro University
Fakultetsgatan 1, S-70182 Örebro, Sweden
{*silvia.coradeschi,alessandro.saffiotti*}*@aass.oru.se*

Abstract

Anchoring is the problem of connecting, inside an artificial system, symbols and sensor data that refer to the same physical objects in the external world. This problem needs to be solved in any robotic system that incorporates a symbolic component. However, it is only recently that the anchoring problem has started to be addressed as a problem *per se*, and a few general solutions have begun to appear in the literature. This paper introduces the special issue on *perceptual anchoring* of the *Robotics and Autonomous Systems* journal. Our goal is to provide a general overview of the anchoring problem, and to highlight some of its subtle points.

> *For things to exist there are two essential conditions, that a man should see them and be able to give them a name.* [20, p. 53]

1 Introduction

> You are at a friend's house and your host asks you to go to the cellar and fetch the bottle of Barolo wine stored at the top of the green rack. You go down to the cellar, look around in order to identify the green rack, and visually scan the top of the rack to find a bottle-like object with a Barolo label. When you see it, you reach out your hand to grasp it, and bring it upstairs.

This vignette illustrates a mechanism that we constantly employ in our everyday life: the use of words to refer to objects in the physical world, and to communicate a specific reference to another agent. This example presents one peculiar instance of this mechanism, one in which the first agent "knows" which object he wants but cannot see it, while the second agent only has an incomplete description of the object but can see it. Put crudely, the two agents

Preprint submitted to Elsevier Science *23 January 2003*

Published in: *Robotics and Autonomous Systems* 43(2-3):85-96, 2003.

that embody two different types of processes: one that reasons about abstract representations of objects, and one that has access to perceptual data. One of the prerequisites for the successful cooperation between these processes is that they agree about the objects they talk about, that is, that there is a correspondence between the abstract representations and the perceptual data which refer to the same physical objects. In other words, there must be a correspondence between the names of things and their perceptual image. We call *anchoring* the process of establishing and maintaining this correspondence [4,18].

Not unlike our example, autonomous systems embedded in the physical world typically incorporate two different types of processes: high-level cognitive processes, that perform abstract reasoning and generate plans for actions; and sensory-motoric processes, that observe the physical world and execute actions in it — see Fig. 1. The crucial observation here is that these processes have different ways of referring to the same physical objects in the environment. Cognitive processes typically (although not necessarily) use symbols to denote objects, while sensory-motoric processes typically operate from sensor data that originate from observing these objects. If the overall system has to successfully perform its task, it needs to make sure that these processes "talk about" the same physical objects: that is, it has to perform anchoring.

Suppose for concreteness that a robot's planner has generated the action `PickUp(bottle-22)`, where the symbol `bottle-22` denotes an object known by the planner to be a bottle and to contain Barolo wine. In order to execute this action, the robot might start a `PickUp` operator implemented by visual-servoing the robot's arm with respect to a given region in the camera input. But *which* region? Intuitively, the robot must make sure that the region used for controlling the arm is precisely the one generated by observing the object that the planner calls `bottle-22`. That is, the robot must *anchor* the symbol `bottle-22` to the right sensor data. How the "right" data can be identified from the sensor stream is part of the anchoring problem.

The above considerations suggest that anchoring must necessarily take place in any robotic system that comprises a symbolic reasoning component. Until recently, however, the anchoring problem was typically solved on a system-by-system basis, often using techniques from the pattern recognition or object tracking domains, and the solution was hidden in the code. The situation is now changing, and the field of autonomous robots is showing a tendency to engage in the study of the anchoring problem *per se* — see for instance [5]. This study would allow us to develop a set of common principles and techniques for anchoring that can be easily applied across different systems and domains. From a more general perspective, a study of the anchoring problem would increase our understanding of the delicate issue of integration between symbolic reasoning and physical embodiment. The papers in this Special Issue

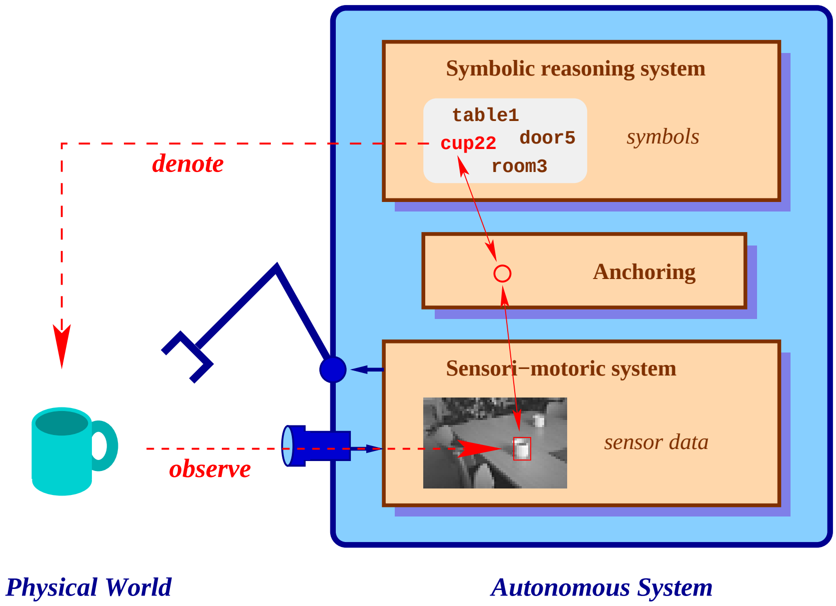

Fig. 1. Graphical illustration of the anchoring problem.

discuss possible solutions to the anchoring problem in its different facets and in different application domains.

2 The Anchoring Problem

Having recognized the existence of the anchoring problem, the next step is to define it in a more precise way. This is an obvious prerequisite to being able to devise general theories and techniques to address it. We give the following definition.

Definition 1 *We call* anchoring *the process of creating and maintaining the correspondence between symbols and sensor data that refer to the same physical objects. The* anchoring problem *is the problem of how to perform anchoring in an artificial system.*

This definition clearly covers the informal account given in the Introduction, but in fact it defines the anchoring problem in more general terms. In the rest of this section, we discuss this definition by highlighting the assumptions that it makes and those that it does not make.

The first thing to note is that the definition does not make any assumption about the direction of the anchoring process. In our introductory example, we were concerned with the top-down problem of identifying the "right" object to be used for a given task, and of allowing the sensory-motoric subsystem in the robot to operate on that specific object. Anchoring, however, can be performed top-down, bottom-up, or in both directions simultaneously. For example, in some systems the flow of sensor data determines, in a bottom-up fashion, which anchoring processes are initiated. An example in this issue can be found in the paper by Steels and Baillie [22], which focuses on the interpretation of scenes using linguistic terms.

A second observation is that our definition does not make any assumption about the type of architecture used in the robotic agent. In the Introduction, we have considered an agent endowed with a specific architecture. However, the definition simply assumes an agent that uses *symbols* to denote individual physical objects, and that has access to *sensor data* that refer to those objects.

As for the assumptions that our definition does make, the main one is that anchoring concerns *physical objects*. Anchoring concerns the grounding of the name for an object, say ‘car-22’, to the perceptual data that originates from the observation of that specific object, say a region in an image. In particular, anchoring as defined above does not concern the perceptual grounding of properties, like ‘red’. Grounding of properties is of course an important problem. Moreover, as we shall shortly see, it is a prerequisite to the perceptual grounding of physical objects, since objects can only be identified by their properties. However, the assumption to deal with individual physical objects has important consequences that differentiate anchoring from generic symbol grounding.

Physical objects persist in time and space, and some of their properties are preserved across time or evolve in predictable ways. The anchoring process must take this temporal dimension into account: anchoring cannot be modeled as a one-shot process, but it must take into account the flow of continuously changing sensor input. That is why our definition explicitly mentions the aspect of *maintenance*.

One way of taking object persistence into account is to include in the anchoring process a persistent internal representation that reifies the correspondence between symbols and sensor data. This representation can contain memory of the past and it can support prediction of the future. It can be used to track the object and to reacquire an object which has been out of sight. In our terminology, we refer to this representation as an *anchor*. An anchor can be seen as an internal model of a physical object that links together the symbol-level and sensor-level representations of that object. Many contributions in this issue include internal representations that play a role similar to anchors. For instance, Khoo and Horswill [14] use markers, Shapiro and Ismail [21] use PML-descriptions, and Fritsch *et al.* [12] use a hierarchy of anchors.

An important aspect of anchors is that they can be shared across different sub-systems of the agent in order to provide them with a *common handle* to refer to a specific physical object. In the example given in the Introduction, when the agent sees a bottle that matches the given linguistic description, it acquires perceptual properties like its size and position. These properties are then used to control the motion of the arm. In our terminology, the agent has created an anchor for the bottle. The anchor has persistence: if the agent momentarily loses sight of the bottle, e.g., while looking elsewhere, it can still move its arm using the internally stored position of the bottle. Anchors can be used for more than controlling motion: in the systems presented in this special issue, similar representations are used to coordinate task execution [14], to engage in communicative actions [21], to achieve a shared language [22,23], and to enable human-robot interaction [2].

The focus on individual objects has a second, important consequence: individual objects should be perceptually detected as such. In other words, our definition assumes as a prerequisite for anchoring that the available sensor data can be segmented to isolate *percepts* that correspond to individual objects. This assumption is not free of cost: the figure-ground segmentation is known to be a difficult problem, which is highly domain specific [15]. Moreover, the notion of "individual object" crucially depends on the sensory apparatus available to the agent, and it does not necessarily correspond to our intuitive, human-centered notion. For example, for a robot equipped with only sonar sensors the individual objects may be the different "places" in the environment which it is able to discriminate, and these are therefore the referents of the anchoring process for that robot.

The focus on individual objects does not exclude the possibility that these objects may be composed of several other objects, possibly in a complex structure. In this special issue, the paper by Chella *et al.* [3] considers a robotic finger as a composite object consisting of the different phalanxes; and the paper by Fritsch *et al.* [12] considers anchoring a person by anchoring a face and two legs, which are perceived by different sensors. Anchoring of groups of objects can be done as a group, or on an individual basis. In both cases the relations among objects probably need to be taken into account in the anchoring process.

Finally, some authors have applied the notion of anchoring to more general entities. In particular, some of the articles in this issue consider the correspondence between symbols and sensor data that refer to individual *actions* and *events* [3,22]. Interestingly, these authors can use similar principles to deal with the anchoring of physical objects and of these more abstract entities: it would constitute an interesting development to understand the differences and the similarities between these two types of anchoring.

3 The Challenges of Anchoring

Anchoring is a problem that can be studied from a number of different perspectives and within several disciplines. Philosophy, linguistics, and cognitive science are the ones that first come to mind. A study of the anchoring problem can raise a number of very challenging issues from each of these perspectives. While this suggests that a complete study of the anchoring problem can be an extraordinarily difficult task, we nonetheless need to develop practical, albeit partial solutions to this problem if we want to build working systems. In this section, we discuss those challenges that constitute, in our opinion, the most practical concerns that need to be addressed if one wants to build a robotic system where anchoring is present.

A first challenge is represented by the presence of uncertainty and ambiguity. Uncertainty and ambiguity obviously arise when anchoring is performed using real sensors, which have intrinsic limitations, and in an environment which cannot be optimized in order to reduce these limitations. The anchoring process might incorporate provisions to deal with these limitations, for instance by managing multiple hypotheses. Alternatively, it can rely on the perceptual system to filter out the uncertainty, or it can delegate the resolution of ambiguities to the symbolic level. In general, it may be difficult to assess what is properly handled at the anchoring level and what should be resolved at the symbolic or sensor level.

In addition to the limitations of sensing, there are aspects of uncertainty and ambiguity which are inherent to the anchoring problem itself. Vagueness of symbolic descriptions is a first example. Symbolic properties often do not have a precise definition in terms of measurable attributes, especially those used in natural language like 'red', and the matching between sensor data and symbolic descriptions is usually better described in terms of similarity than identity. A second aspect is the possibility of a mismatch between what we would like to discriminate at the symbolic level, e.g., colored objects, and what can be actually discriminated by the sensors, e.g., a black and white camera. A third aspect is that at the symbolic level we can refer to objects with a specific identity, like '`cup-22`', while the perceptual system is not in general able to perceive the identity of an object but only some of its properties. Although these factors may all end up in the same problem —uncertainty about the identity of perceived objects— their treatment in the anchoring process should probably be differentiated.

Another challenge of anchoring is that, at the symbolic level, there are several ways to refer to objects. An important distinction is between definite and indefinite symbolic descriptions. A definite description implies the existence of a unique object satisfying the description in the current context. For instance, '*the* cup belonging to Silvia' can denote an unique object in the office, even if Silvia can own many more cups at home. An indefinite description denotes an object having a number of properties, without any assumption about its uniqueness. For instance, '*a* red cup' is an indefinite description satisfied by any red cup in the current context. The importance of this distinction appears mainly when more than one object satisfies the description: this can be a problem in the case of definite description, but not in the case of indefinite ones. One may consider several more types of descriptions, for instance, descriptions that use functional properties like 'something to hold water'. The many ways of giving a reference brings about the problem of how the anchoring process should treat different kinds of descriptions.

In Fig. 1 just one object and one observer are present. This is clearly a simplified case. In general, it may be necessary to anchor several objects at the same time and to identify objects on the basis of the relations among them. Moreover an agent could observe an object with different sensors and/or from different points of view, and then need to integrate this information to be able to establish an anchor. We have an example in this issue in the paper by Fritsch *et al.* [12]. A similar problem arises if robots with different sensors need to exchange information about the objects in the environment. A robot could anchor an object on the basis of properties that cannot be discriminated by another one.

Difficult issues of communication and negotiation may arise if several robots need not only to anchor symbols internally, but also to exchange information among them and to agree on a shared language. Common agreement about the meaning of the symbols used to refer to objects in the environment is also needed for efficient human-robot cooperation. Some of the papers in this special issue deal with systems that involve communication among multiple robots [14,22,23] or between robots and humans [2].

Finally, a fundamental challenge of the anchoring problem is to investigate the formal properties of the anchoring process. Intuitively one may feel that some correspondences between the symbols and the sensor data are correct while some are not. How to express this formally, and how to prove the correctness of a specific system are open problems. Engaging in this study would probably require the ability to model both the anchoring system and the physical environment in the same formal system, in which we can define and prove formal properties.

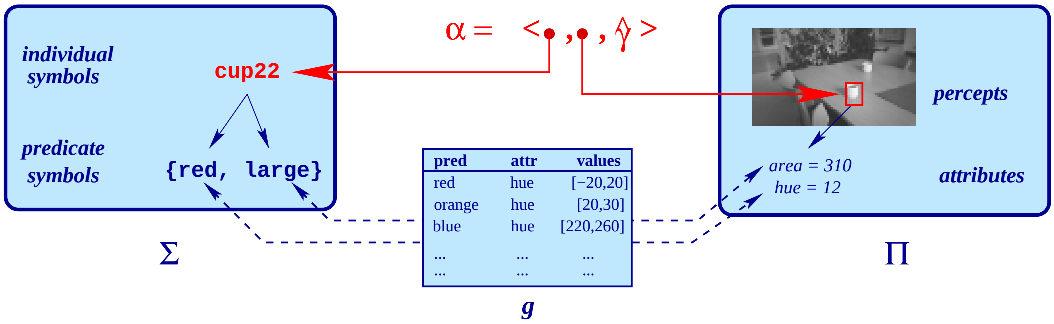

Fig. 2. The ingredients of anchoring in our framework. α is the anchor.

4 Anchoring in Practice

In order to get a better understanding of how the general concept of anchoring can be instantiated in different tasks and different domains, we present below a few implemented systems that perform anchoring. First, however, we need to outline the main ingredients of the framework for anchoring which is used in all the examples. A detailed description of this framework and of the examples can be found in [4,6,7].

4.1 Ingredients and functionalities of anchoring

According to our framework the anchoring process is performed in an intelligent embedded system that comprises a *symbol system* Σ and a *perceptual system* Π (see Fig. 2). The symbol system manipulates individual symbols, like '`x`' and '`cup22`', which are meant to denote physical objects. It also associates each individual symbol with a set of symbolic predicates, like '`red`', that assert properties of the corresponding object. The perceptual system generates percepts,[1] like a region in an image, from the observation of physical objects. It also associates each percept with the observed values of a set of measurable attributes, like the average hue values of a region.

The model further assumes that a *predicate grounding relation* g is given, which encodes the correspondence between predicate symbols and admissible values of observable attributes. How the admissible values are encoded may differ across different applications. For instance, they may be represented by ranges, or by fuzzy sets. No assumption is made about the origin of the g relation: it can be hand-coded by the designer, learnt from samples, or other.

The task of anchoring is to use the g relation to connect individual symbols in

[1] We take here a percept to be a structured collection of measurements that are assumed to originate from the same physical object.

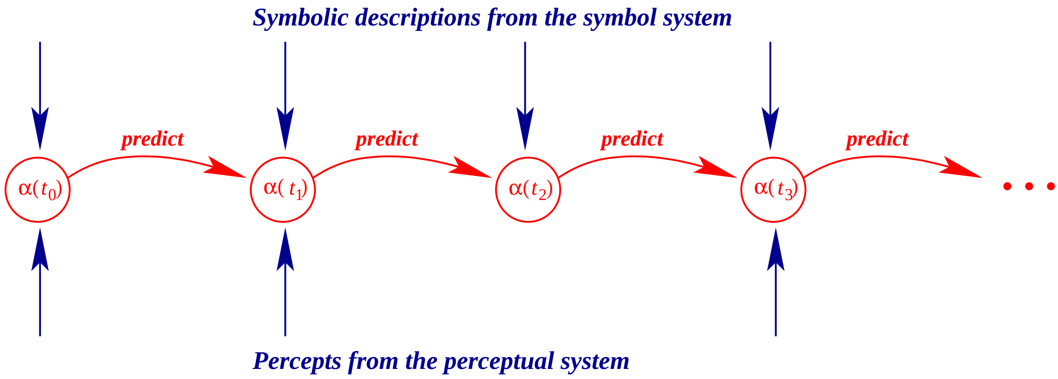

Fig. 3. Anchor dynamics. The anchor α is created by the **Find** functionality, and then maintained by the **Track** and **Reacquire** functionalities.

Σ and percepts in Π. For instance, suppose that `red` is predicated of the symbol `cup22`, and that the hue values of a given region in an image are compatible with the predicate `red` according to g. Then that region could be anchored to the symbol `cup22`. The correspondence between symbols and percepts is reified in a data structure called *anchor*, denoted by α in the figure. The anchor contains pointers to the corresponding symbols and percepts, together with an estimate of the current values of some of the attributes of the object which it refers to, called *signature* and denoted by $\hat{\gamma}$. The values in the signature, like the object's position, can be used both for acting on the object and for re-identifying it later on. They can also be used to operate on the object when this is not directly visible. An anchor can be considered as a model of a physical object that reflects the persistence of the object, and which can be shared across different sub-systems of the agent.

The anchoring process is defined in our framework by three abstract functionalities that manage anchors: **Find**, **Track**, and **Reacquire**. These functionalities have been found adequate to capture top-down anchoring in several applications. Additional functionalities will probably be needed for different types of anchoring processes, for instance, bottom-up anchoring.

The **Find** functionality corresponds to the initial creation of an anchor for an object given a symbolic description (set of predicates) provided by Σ. This functionality selects a percept from the perceptual stream provided by Π using the g predicate grounding relation to match predicates to observed attribute values. The initial creation of an anchor resembles a structural pattern recognition process.

Once an anchor has been created, it must be continuously updated to account for changes in the object's attributes, e.g., its position. This is done by the **Track** functionality using a combination of prediction and new observations, as illustrated in Fig. 3. Prediction is used to make sure that the new percepts used to update the anchor are compatible with the previous observations, that is, that we are still tracking the same object. Moreover, comparison with the symbolic descriptor is used to make sure that the updated anchor still satisfies the predicates, that is, that the object still has the properties that make it "the right one" from the point of view of the symbol system. The use of abstract symbolic information inside the tracking cycle differentiates anchor maintenance from the usual predict-measure-update cycle of recursive estimators like Kalman filters. The second example below illustrates a case where this information is crucial to a correct anchoring.

The Track functionality assumes that the object is kept under constant observation. The **Reacquire** functionality takes care of the case in which the object is re-observed after some time. For instance, every morning I tell my robot to go and pick up my cup. The robot knows what my cup looks like and where it has seen it last time, and it can use this information to find it again. The Reacquire functionality can be considered a combination of Find and Track: it is similar to a Find, with the addition that information from previously observed attributes can also be used as in the Track functionality.

4.2 Anchoring in an office navigation domain

The aim of this first example is illustrate a simple case of anchoring We consider a Nomad 200 robot equipped with an array of sonar sensors and controlled by an architecture similar to the one reported in [19], which includes a simple STRIPS-like planner. All the perceptual and prior information about the robot's surroundings is maintained in a blackboard-like structure called Local Perceptual Space (LPS). In terms of our framework, the *symbol system* is given by the planner; individual symbols denote rooms, corridors, and doors. The *perceptual system* extracts features from histories of sonar measurements; percepts include walls and doors. The *predicate grounding relation* is hand-coded, and it maps predicates like `narrow_door` to ranges of values for the observed door width, like $[60, 80]$. Finally, an *anchor* contains pointers to the appropriate symbols and percepts, plus a signature. Symbolic descriptions, percepts, and anchors are all Lisp structures stored in the LPS.

The task that we consider in this example is navigation in an office environment, as shown in Fig. 4. Anchoring arises when the planner gives direction to the robot in terms of names of rooms and corridors, for instance `Follow(corr4)`. The robot needs to anchor the symbol `corr4` to the sonar data corresponding to the walls of the actual corridor denoted by `corr4`. At time t_0 the planner puts the symbolic description of `corr4` into the LPS based on map information (shown by thick lines in the figure). At t_1 this descriptor is anchored to wall percepts (shown by thin segments) using **Find**. **Track** is then used to keep it anchored to the new wall percepts. The signature in the

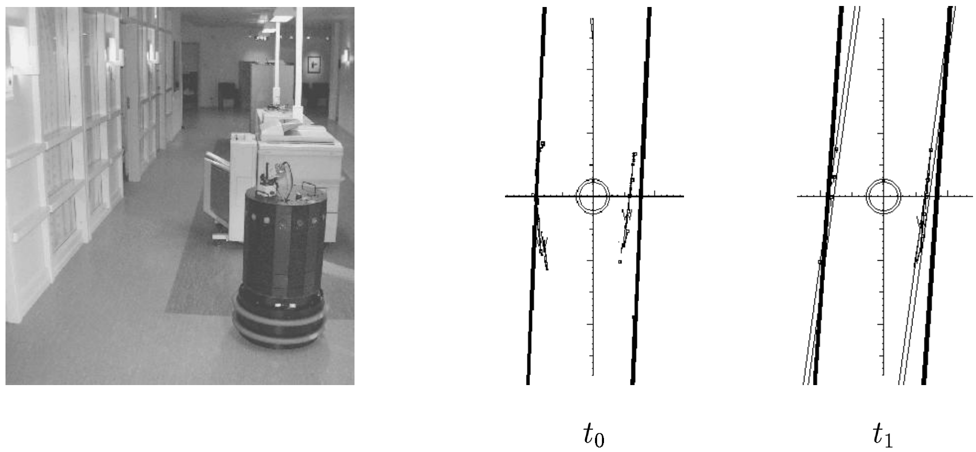

Fig. 4. Anchoring a corridor. t_0: before anchoring. t_1 after anchoring.

anchor (shown by double lines) is used by the `Follow` behavior to control the movement of the robot along the intended corridor.

4.3 Anchoring in an aerial surveillance domain

The next example emphasizes the dynamic aspect of anchoring and the use of symbolic information in predicting the next position of an object. The domain is an unmanned aerial vehicle (UAV) performing autonomous surveillance tasks in a simulated environment developed within the WITAS project [9]. The UAV system integrates a planner, a reactive plan executor, a vision system and a control system.

In terms of our framework, the *symbol system* consists of the planner; individual symbols denote cars and elements of the road network. The *perceptual system* is a reconfigurable active vision system able to extract information about car-like objects in aerial images; percepts are regions in the image, and they have attributes like position, width, and color. The *predicate grounding relation* is given as a hand-coded table that associates each predicate symbol with a fuzzy set of admissible values for the corresponding attribute. An *anchor* is a Lisp structure that stores an individual symbol, the index of a region, and an association list recording the current estimates of the values of the object's attributes (signature). The signature in the anchor is used to configure the vision system, to control the camera, and to control the UAV.

In the example shown in Fig. 5, the task of the UAV is to follow a specific car that was previously anchored using the **Find** functionality. At time t_0 two identical cars are present in the image, one traveling along a road which makes a bend under a bridge, and one traveling on the bridge. The UAV is

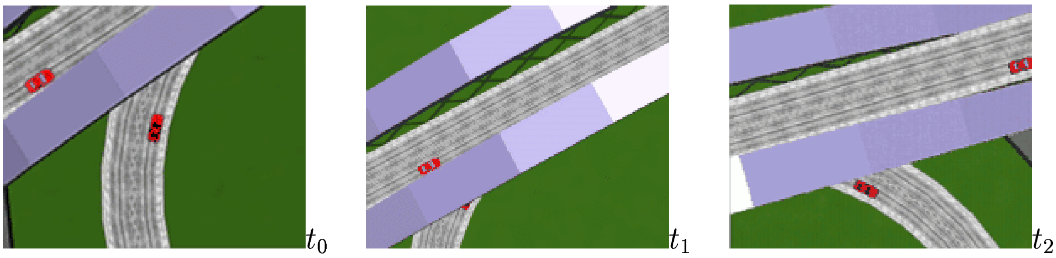

Fig. 5. Anchoring a moving object. The followed car disappears under a bridge and a similar car appears at its place over the bridge.

keeping under observation the car traveling along the road using the **Track** functionality. At t_1 this car disappears under the bridge and the second car is almost in the position in the image where the first one was expected to be. The Track functionality has access to the symbolic information about road topology and can therefore recognize that this car cannot be the car previously tracked. The **Reacquire** functionality is then invoked in order to find again the tracked car. Reacquire uses high-level knowledge to infer the presence of the occluding bridge, and to predict the next visible position of the car. This position is stored in the signature of the anchor, and it is used to direct the UAV and the camera toward the end of the bridge. When the car reappears from under the bridge at t_2, a percept is generated by the vision system that is compatible with the signature in the anchor. Normal tracking is then resumed.

4.4 Anchoring an indefinite description

Our last example is intended to illustrate some of the subtleties of the anchoring problem in the case of an indefinite reference and of multiple identical objects [7]. The task is one of the three "technical challenges" of the RoboCup 2002 competition in the Sony 4-legged robot league. A Sony AIBO robot is in a soccer field and 10 identical balls are placed in the field. The task is to score all the balls. When a ball is scored, it is removed from the field — see Fig. 6.

With respect to anchoring, the problem can be described as follows. The robot is given an indefinite description of a ball, for instance, '$x : \mathrm{Ball}(x) \wedge \mathrm{Red}(x)$'. Any of the 10 balls is suitable for the task. The **Find** functionality selects the first ball to act upon, for instance the nearest one, and anchors the symbol x to it. The created anchor includes in its signature the relative position of this ball, which is used by the motion and kicking routines. While the robot moves, the **Track** functionality updates the anchor regularly.

The Track functionality has an implicit definite reference: the ball which the robot is currently acting on. In a sense, anchoring has made the robot committed to that specific ball. However the anchoring process must remember

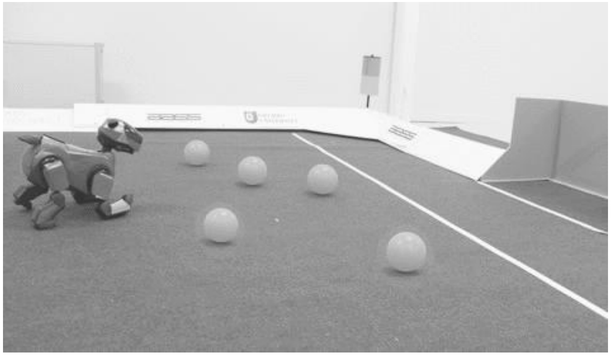

Fig. 6. Anchoring "a red ball" to perform a ball collection task.

that the original description was an indefinite one, and that another ball can also be suitable for the task. For instance, when the current ball is removed from the field the robot must try to **Reacquire** and then Track another ball, since the task was to score an arbitrary ball. Smarter anchoring strategies can be devised for this task. For instance, the robot should not remain committed to a ball if another ball is in a better position according to some specified criteria. In our implementation of this example, the robot tracks one specific ball and acts on it, but if it sees another ball which is in the same direction and closer, it starts tracking and acting on this other one.

5 Related Problems

The problem of connecting linguistic descriptions of objects to their physical referents has been largely considered in the fields of **philosophy** and **linguistics**. In fact, we have borrowed the term *anchor* from situation semantics.[2] Most of the aspects of anchoring discussed in this paper have also been studied in these fields. For example, the distinction between definite and indefinite references and the semantical problems associated with definite references have been addressed, among others, by Russell [17] and Frege [11]. While the anchoring problem could certainly belong to the philosophical and linguistic debate, the perspective taken here is more pragmatic. We are interested in ways of stating and solving this problem that can lead to implementing practical solutions in robotic systems. Even with this difference in perspective, the reflections done in the linguistic and philosophical fields undoubtedly provide a rich source of inspiration for the study of the different aspects of the anchoring problem. Two book reviews in this special issue introduce examples of the

[2] Situation semantics [1] is a semantics of natural language that tries to find meanings of sentences in the external world and in relations between situations rather than in truth values as in logic based semantics. In the terminology of situation semantics, an anchor is an assignment of individuals, relations and locations to abstract objects.

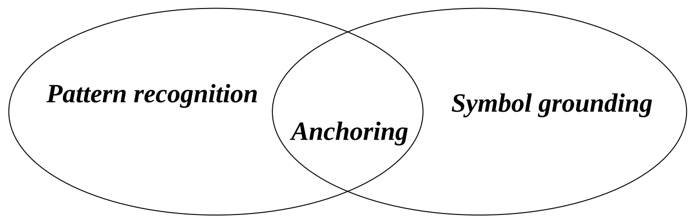

Fig. 7. Relations among Anchoring, Symbol Grounding and Pattern Recognition.

work done in the philosophical community which is relevant to anchoring.

From a more practical point of view, there are two research problems in the fields of robotics and of AI which are related to the anchoring problem: pattern recognition and symbol grounding. **Pattern recognition** can be defined as the problem of interpreting data provided by sensors by assigning them to predefined categories [10,16]. Taking pattern recognition in its most general sense, anchoring can be considered a subproblem of pattern recognition. However, the anchoring problem emphasizes several peculiar aspects, which are not usually the focus of pattern recognition. First, the presence of symbols is an essential aspect of anchoring, while this is not the case in pattern recognition. Second, a goal of anchoring is the dynamic maintenance of the anchor in time, while pattern recognition is mostly used in applications where this dynamic aspect is not relevant. Finally, anchoring focuses on the creation and maintenance of the anchor as a shared representation to link several sub-systems of the agent, such as motor control, sensor processing, and reasoning.

Symbol grounding can be defined as the problem of finding a semantics for a symbolic system that it is not in its turn a symbolic system [13]. Symbol grounding is a more general problem than anchoring. It concerns the philosophical issues related to the meaning of symbols in general. Anchoring is concerned with the practical problem of connecting symbols referring to physical objects to the sensor data originating from those physical objects in a implemented robotic system. In particular, anchoring focuses on perceivable physical objects, while symbol grounding needs to consider all kind of symbols, including ones like 'justice' and 'peace'. For these kinds of symbols it would be difficult to find appropriate sensor measurements, while the presence of sensor measurements is essential in anchoring.

Fig. 7 shows a simplified view of the relation among anchoring, symbol grounding and pattern recognition. Anchoring is included in the intersection between the other two problems and can represent a bridge between them. One can in fact find numerous cases of pattern recognition where no symbols are present, and one can study the symbol grounding problem without taking measurements in consideration. Anchoring by contrast implies the presence of both symbols and measurements and the possibility of establishing a connection between the two.

An important aspect of anchoring is that the referents are individual physical objects. In this respect, anchoring is related to the problem of **object tracking**. In object tracking an object is first found, and then kept under observation for some time. Some instances of the anchoring problem can also be considered instances of the object tracking problem, like the car tracking example discussed in the UAV scenario above. As shown in that example, the presence of symbols and the possibility of performing symbolic reasoning is a distinctive aspect of anchoring, which is usually not considered in object tracking. Other instances of the anchoring problem would not be easily expressed in terms of object tracking. An example is the case where a robot finds an object in a room and some days later is asked to reacquire the object that can or cannot be still present in the room. This is a clear case of anchoring, but it could hardly be regarded as object tracking. The paper by Steels and Baillie in this issue [22], which focuses on the interpretation of scenes using linguistic terms, provides another example.

We can summarize the above considerations as follows. Although specific instances of the anchoring problem can be also seen as instances of other problems studied in AI and in robotics, like symbol grounding, pattern recognition, and object tracking, the general anchoring problem has nonetheless several distinctive aspects that make it worth studying as a problem *per se*. Practical solutions to the anchoring problem will, of course, draw from the wide set of techniques developed to address these other problems, as well as from the debate about the relation between symbols, perception and reality which has animated the fields of philosophy, linguistics and psychology.

6 About the Papers in this Issue

Most of the papers contained in this special issue present specific systems that address the anchoring problem as defined in this introduction, although some of them deal with anchoring intended in a somewhat wider sense.

Shapiro and Ismail [21] consider how the anchoring problem is addressed in GLAIR, a three-level architecture for cognitive robots. The robot used in the experiments interacts with humans using natural language, and in order to answer the user's queries it needs to connect its visual input to the linguistics terms used by the human. The robot uses abstract knowledge of objects and persons to make this connection. An example is the dialog where the robot is asked to find Bill, looks for a blue block and when it finds it, it answers that it has found Bill.

Khoo and Horswill [14] present a system which uses reactive plans, expressed in a rule-based format, to perform cooperative tasks involving two robots. The variables used in the reactive rules are anchored to objects in the environment by means of color trackers that are attached to specific objects in a camera image. The two robots exchange information about objects using messages in which the anchored objects are associated to fixed positions in a bit string. The authors demonstrate their approach on two tasks involving co-operative office navigation: find an object, and visit all locations in the environment.

Fritsch *et al.* [12] deal with the problem of anchoring a composite object from the data provided by several sensors, each one of which can only observe part of the object. The authors consider the case of anchoring a human by aggregating the two anchors separately created for the face and for the legs. Face recognition is based on image data, while leg recognition relies on data from a laser range finder. Their system can be seen as a special case of cooperative anchoring, in which a common anchor must be established between two perceptual systems.

Vogt's paper [23] is based on the concept of semiotic symbol. A semiotic symbol is defined by a triadic relation among form, meaning, and referent and it therefore implicitly includes an anchoring relation between the form, symbol in the traditional sense, and the referent object. The approach considered is bottom-up from sensor data to names, and the experiment presented involves two robots sensing light sources and developing a lexicon to name the light sources.

Among the papers that deal with the anchoring problem intended in a wider sense, the paper by Steels and Baillie [22] considers the anchoring not only of objects but also of events. The system anchors objects seen in the images bottom-up, and keeps track of them over time. On the basis of this information events are recognized. This work is in the context of a language game between two robotic systems with the aim of learning a shared language. One of the systems sees a event, like a ball rolling, through a static camera in an otherwise static environment. It then formulates a sentence describing the event. The other system hears the sentence and interprets it. If the interpretation is considered appropriate with respect to one of the events recently seen the game succeeds.

Chella *et al.* [3] deal with the problem of recognizing motion events of a robotic finger observed by an external camera. They propose a framework based on Gardenförs' theory of *conceptual spaces*. In their system, each element (phalanx) of the robotic finger is anchored bottom-up to a point in a conceptual space, or *knoxel*. The knoxels that correspond to the different phalanxes of a finger are aggregated into a new knoxel which provides an anchor for the full finger object. In addition to anchoring individual physical objects, Chella *et al.* also deal with anchoring symbols that denote actions and fluents by considering the dynamic evolution of (sets of) knoxels. For example, the fluent "in_motion" is anchored to a set of knoxels that correspond to a given evolution in time of the finger. The fluents and actions so anchored are used in a logical system, formalized in situation calculus, where higher-level event recognition takes place.

Bredeche *et al.* [2] present a robotic system capable of learning the association between symbols like 'human' and 'fire extinguisher' and visual percepts. The robot takes snapshots of the environment that are then labeled by a supervisor. The aim is that the robot, after a number of label-percept associations, should be able to label autonomously a new environment. The authors focus more on the learning of basic concepts like 'human' than on the anchoring of a specific individual, a specific human. The learning of the association between concepts and sensor data does not cover the whole anchoring problem, but it is an essential ingredient in the process of connecting an individual symbol, like Silvia, to the sensor data associated with the specific human being known by the robot as Silvia.

This special issue is completed by the reviews of two books which may provide interesting insights on the anchoring problem from a philosophical perspective. The first one is *The Varieties of Reference*, by Gareth Evans. The second book is *Conceptual Spaces*, by Peter Gardenförs. The book reviews highlight the relevance of these two works to the problem of perceptual anchoring.

7 Conclusions

As robots are moving toward more complex tasks and environments, the field of robotics is looking more and more to ways of including higher level representations and reasoning into robotic systems. In many cases, the higher level is built around a symbol system. The claim made in this paper is that any physically embedded robotic system which includes a symbolic component needs to perform anchoring.

Anchoring is a difficult problem. It involves concepts which have interested philosophers for centuries and are still far from being fully understood. Nonetheless, we have to provide practical solutions to the anchoring problem if we want to build robotic systems that include a symbolic component. The papers in this special issue provide examples of such solutions. In the longer term, a research program on anchoring should bring a deeper theoretical analysis of the anchoring problem, together with general practical solutions that can be re-used in different systems and domains.

Acknowledgements

This work was funded by the Swedish KK Foundation and by Vetenskapsrådet. We thank Isabelle Bloch, Antonio Chella, Lars Karlsson, Amy Loutfi, and Stuart Shapiro for providing helpful comments to this paper.

References

[1] J. Barwise and J. Perry. *Situations and Attitudes.* The MIT Press, 1983.

[2] N. Bredeche, Y. Chevaleyre, J. D. Zucker, A. Drogoul, and G. Sabah. A meta-learning approach to ground symbols from visual percepts. *In [8]*, 2003.

[3] A. Chella, M. Frixione, and S. Gaglio. Anchoring symbols on conceptual spaces: the case of dynamic scenarios. *In [8]*, 2003.

[4] S. Coradeschi and A. Saffiotti. Anchoring symbols to sensor data: preliminary report. In *Proc. of the 17th Nat. Conf. on Artificial Intelligence (AAAI)*, pages 129–135, Austin, 2000. Online at http://www.aass.oru.se/Research/Robots/.

[5] S. Coradeschi and A. Saffiotti, editors. *AAAI Fall Symposium on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems*, AAAI Technical Report FS-01-01, Menlo Park, CA, 2001. AAAI. Online at http://www.aass.oru.se/Agora/FSS01/.

[6] S. Coradeschi and A. Saffiotti. Perceptual anchoring of symbols for action. In *Proc. of the Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 407–412, Seattle, USA, 2001. Online at http://www.aass.oru.se/Research/Robots/.

[7] S. Coradeschi and A. Saffiotti. Perceptual anchoring with indefinite descriptions. In *Proc. of the First Joint SAIS-SSLS Workshop*, Örebro, Sweden, 2003. Online at http://www.aass.oru.se/Research/Robots/.

[8] S. Coradeschi and A. Saffiotti, editors. *Robotics and Autonomous Systems, special issue on Perceptual Anchoring.* Elsevier Science, 2003. This issue.

[9] P. Doherty. The WITAS integrated software system architecture. *Linköping Electronic Articles in Computer and Information Science*, 4(17), 1999. http://www.ep.liu.se/ea/cis/1999/017.

[10] R. Duda and P. Hart. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.

[11] G. Frege. Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, pages 25–50, 1892.

[12] J. Fritsch, M. Kleinehagenbrock, S. Lang, F. Loemker, G. A. Fink, and G. Sagerer. Multi-modal anchoring for human-robot-interaction. *In [8]*, 2003.

[13] S. Harnad. The symbol grounding problem. *Physica D*, 42:335–346, 1990.

[14] A. Khoo and I. Horswill. Grounding inference in distributed multi-robot environments. *In [8]*, 2003.

[15] D. C. Marr. *Vision*. W.H. Freeman and Co, San Francisco, CA, 1982.

[16] M. Pavel. *Fundamentals of Pattern Recognition*. Marcel-Dekker, New York, 1993.

[17] B. Russell. On denoting. *Mind*, XIV:479–493, 1905.

[18] A. Saffiotti. Pick-up what? In C. Bäckström and E. Sandewall, editors, *Current trends in AI Planning*, pages 266–277. IOS Press, Amsterdam, NL, 1994.

[19] A. Saffiotti, K. Konolige, and E. H. Ruspini. A multivalued-logic approach to integrating planning and control. *Artificial Intelligence*, 76(1-2):481–526, 1995.

[20] J. Saramago. *The Stone Raft*. Harcourt Brace, New York, 1995.

[21] S. C. Shapiro and H. O. Ismail. Anchoring in a grounded layered architecture with integrated reasoning. *In [8]*, 2003.

[22] L. Steels and J. C. Baillie. The grounding of dynamic world models by autonomous robots. *In [8]*, 2003.

[23] P. Vogt. Bootstrapping semiotic symbols through adaptive language use. *In [8]*, 2003.