# Perceptual Anchoring:
## A key concept for plan execution in embedded systems

Silvia Coradeschi and Alessandro Saffiotti
{silvia.coradeschi,alessandro.saffiotti}@aass.oru.se

Center for Applied Autonomous Sensor Systems
Dept. of Technology, Örebro University
S-70182 Örebro, Sweden
http://www.aass.oru.se

**Abstract.** Anchoring is the process of creating and maintaining the correspondence between symbols and percepts that refer to the same physical objects. This process must necessarily be present in any physically embedded system that includes a symbolic component, for instance, in an autonomous robot that uses a planner to generate strategic decisions. However, no systematic study of anchoring as a problem *per se* has been reported in the literature on intelligent systems. In this paper, we advocate for the need for a domain-independent framework to deal with the anchoring problem, and we report some initial steps in this direction. We illustrate our arguments and framework by showing experiments performed on a real mobile robot.

## 1 Perceptual anchoring

*It's ten o'clock and I need more coffee. I tell Milou, my personal robot, to go and fetch my cup of coffee, which is on the kitchen table. Milou rolls to the kitchen, approaches the table, and uses its camera to find an object that looks like a cup of coffee. Two cups are standing on the table, one filled with chocolate and one with coffee. From the camera image, both cups match the given description "cup of coffee," so Milou decides to acquire more information. Milou is equipped with an electronic nose that can be used to identify coffee by smell. It turns toward the first cup, approaches it, and smells it: this does not smell like coffee. It then turns to recover sight of the second cup, approaches it, and smells it: this time the odor matches the intended signature. The camera image gives enough information to accurately estimate the position of the cup, so Milou can comfortably grasp it and bring it to my room.*

This hypothetical scenario illustrates a common mechanism of our everyday life: the use of words to refer to an object in the physical world, and to communicate this reference to another agent. In this case, "my cup of coffee which is on the kitchen table". Our ability to deploy intelligent robots that can provide services to non-technical human users critically depends on our ability to develop

In: M. Beetz, J. Herzberg, M. Ghallab and M. Pollack (eds)
Advances in Plan-Based Control of Robotic Agents (Springer, DE, 2002) pp. 89-105

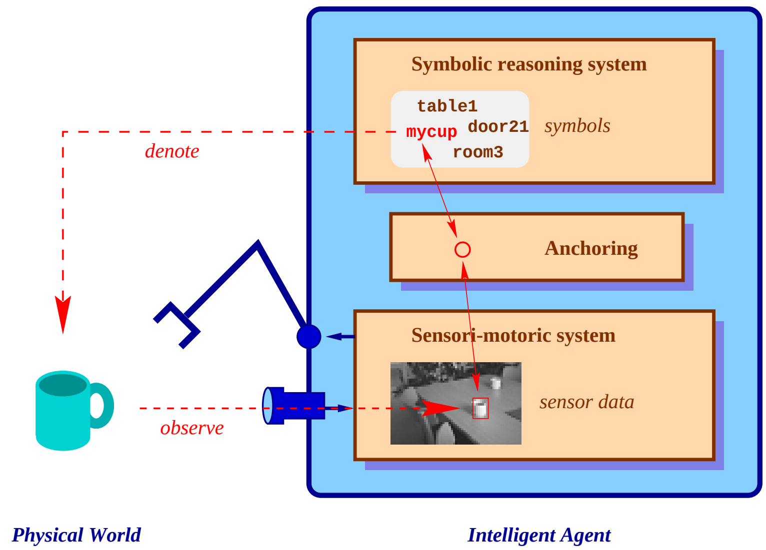



**Fig. 1.** Graphical illustration of the anchoring problem.

mechanisms like this for human-robot interaction. An important prerequisite for this is that the robot must be able to establish the right correspondence between the *symbols* used to refer to a given object, and the *perceptual data* acquired by observing that object. We call **anchoring** the process of creating and maintaining the correspondence between symbols and sensor data that refer to the same physical objects — see Figure 1.

A closer look to anchoring reveals that this is a complex process, which requires the use of several functionalities. In our initial example, Milou is given a linguistic (symbolic) description of the object named "my cup", and it must *find* the intended object through perception. To do so, it must match the data coming from its sensors against this description; it must also recognize ambiguities and decide how to resolve them, e.g., by acquiring more information. Once the object has been identified, the robot must *track* it while moving. Finally, the object may temporarily disappear from the sensor's view, e.g., because it has moved, or because the gaze is turned to another direction. The robot must maintain a virtual image of the object in memory, and be able to *reacquire* the object when it comes back into view. There is currently no general theory that tells us how to define and combine these functionalities to perform anchoring. In this paper, we advocate the need of such a theory, and outline our initial steps toward its development.

## 2 Why plan-based robots need perceptual anchoring

Since its conception in 1956, the field of AI has been pursuing the objective of building intelligent agents that can move and operate in the physical world. For many years, however, the belief that the interesting issues were at the level of the abstract reasoning processes, together with the disconcerting difficulties of perception, kept most AI researchers isolated from embedded systems. Typical AI systems, like expert systems, did not try to directly sense or act upon the physical world: humans did the job of translating observations of the world into the symbols used by these systems, and translating those symbols back to actions in the world. It is only recently that intelligent systems started to be more and more often connected with sensors and actuators to produce *embedded intelligent systems* able to perform useful operations in real world environments (e.g., [21, 3, 11]).

An embedded intelligent system must incorporate motor and perceptual processes to interface with the physical world, and abstract cognitive processes to reason about the world and the available options. In many cases, the abstract processes are symbol-based: when they rely on classical AI techniques, but also when we want our robots to use linguistic communication to interact with humans. In these cases, a crucial aspect of the integration between the cognitive and sensori-motoric level is the connection between the *symbols* used by the symbol system to denote a physical object in the world, and the *sensor data* in the sensori-motoric system that corresponds to the same object (see Fig. 1). The problem of how to create this connection, and how to maintain it in time, is exactly the anchoring problem.

Autonomous robots that incorporate a symbolic planner to plan their operation are examples of intelligent embedded systems. In these robots, anchoring is needed in order to associate the terms used in the plan to the relevant sensor data. Consider for instance a plan that contains the action "PickUp(cup-22)," where "cup-22" is the symbol used by the planner to denote the specific object (say, Alex' cup) to be used to achieve the given goal (say, bring Alex some coffee). In order to execute this action, the robot has to: (i) identify the sensor data in the perceptual stream that pertains to that object; and (ii) use these data in a sensori-motor loop to execute the grasping on the correct object. That is, it has to anchor the symbol "cup-22" to the perceptual data corresponding to the intended object.

Although anchoring must necessarily take place in any robotic system that comprises a symbolic planning and reasoning component, an analysis of the existing literature reveals that the anchoring problem has received little attention in the fields of AI and autonomous robotics as a problem *per se* (see Section 5.1). Instead, anchoring is typically solved on a system-by-system basis on a restricted domain, and the solution is hidden in the code. This is unfortunate, since a deep study of the anchoring problem would allow us to develop a set of common principles and techniques that can be applied to any such system. In a more general perspective, a study of anchoring would increase our understanding of the delicate issue of integration between symbolic planning and reasoning on the one hand, and physical perception and action on the other hand.

To the best of our knowledge, the first domain independent definition of the anchoring problem was given in [24], while the first attempt at defining a computational theory of anchoring was reported in [7]. In what follows, we outline the basic elements of this theory, and show its relevance to the design of plan-based autonomous robots by discussing a few examples.

## 3 A formal model of anchoring

We give here an outline of the formal model of perceptual anchoring proposed in [7] and [9]. The reader is addressed to those references for more details on the model.

### 3.1 The ingredients of anchoring

We consider an intelligent embedded system that includes the following elements.

- A *symbol system* $\Sigma$, which contains individual symbols (variables and constants), predicate symbols, and an inference mechanism. Our interest is directed to the individual and predicate symbols. Examples of individual symbols are `mycup`, `suitcase1`, and `car2`; examples of predicates are `large`, `small`, and `red`.
- A *perceptual system* $\Pi$, which includes percepts and attributes. We take a percept to be a structured collection of measurements that are assumed to originate from the same physical object; an attribute is a measurable property of percepts. Examples of percepts are image regions identified as representing objects; common attributes computed on these percepts are *color*, *width*, and *area*.
- A *predicate grounding relation* $g$, which embodies the correspondence between unary predicates and values of measurable attributes. For instance, $g$ may encode the correspondence between the predicate `red` and the corresponding Hue values measured in the image.[1] We do not make any assumption about the origin of $g$: for instance, $g$ can be hand-coded by the designer of the system, or it can be learned by the system.

We are not concerned with the internal details of $\Sigma$, $\Pi$, anf $g$ here. These can be whatever, as long as they include the elements listed above. What interests us is how to connect these ingredients in order to perform anchoring.

Lets consider a simple example to illustrate the ingredients of anchoring. $\Sigma$ may be a planner that includes the individual symbol 'A' and the predicate symbols 'large' and 'small.' $\Xi$ may be a vision system able to recognize suitcases:

[1] The use of crisp definitions for colors is obviously problematic, and more complex forms of $g$ may be needed in practice. In our implementation, for example, we use fuzzy logic [6] and [5]. We assume a crisp $g$ here for sake of simplicity.

from the image shown in Fig. 2, $\Xi$ may extract two percepts $\pi_1$ and $\pi_2$. Attributes computed by $\Xi$ may include 'color' and 'width.' The predicate grounding relation $g$ may include the triple $\langle \text{small}, \text{width}, 10 \rangle$: this says that the measure 10 for an object's observed width is consistent with the predication of its being small.[2]

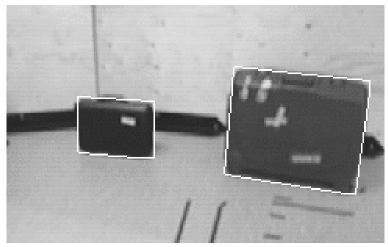


**Fig. 2.** Two percepts extracted from a camera image.

At every point in time, $\Sigma$ contains a set of individual symbols and $\Pi$ contains a set of percepts. Moreover, both systems contain information about the properties associated to these symbols and percepts. This information is represented as follows.

- The *symbolic description* $\sigma$ of a symbol is a set of predicates[3] that are predicated in $\Sigma$ of the symbol; an example of a symbolic description is {`small`, `red`},
- The *perceptual signature* $\gamma$ of a percept is a function that gives the values of all the attributes of the percept, as measured by $\Pi$. $\Gamma$ is the set of all signatures. An example of perceptual signature is: $\gamma(\text{width}) = 230, \gamma(\text{area}) = 380$.

Let us consider our previous example. At time $t$, the symbol system may associate property 'small' to symbol 'A' by having `small` belonging to the symbolic description of 'A'. The perceptual system may extract the width of the two percepts $\pi_1$ and $\pi_2$ in the image, and associate them with their respective perceptual signatures $\gamma_1$ and $\gamma_2$ such that $\gamma_1(\text{width}) = 10$ and $\gamma_2(\text{width}) = 20$.

The task of anchoring is to use the above ingredients to create and maintain the right correspondence between symbols in $\Sigma$ and percepts in $\Pi$. The pivot to this correspondence is the matching between a symbolic description and a perceptual signature: intuitively, we connect a percept and a symbol if their properties are compatible according to $g$. This correspondence is reified in an

[2] For the sake of simplicity we consider here a very simple $g$. The $g$ relation can be quite complex in real domains.

[3] We currently consider in our framework just unary predicates, meant to denote properties of individual objects. The extension to n-ary predicates, meant to denote relations among individuals, is part of our future work.

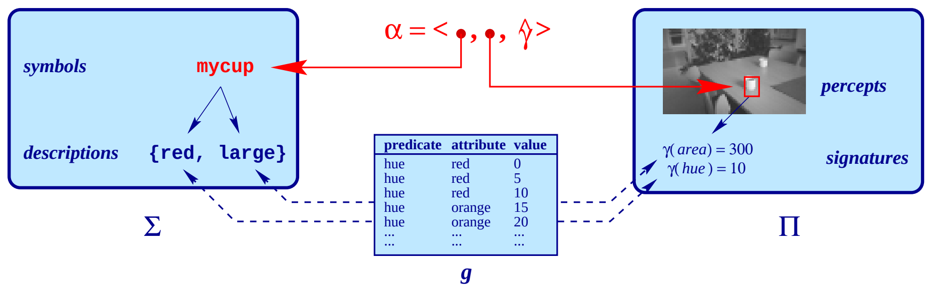



**Fig. 3.** The elements of our model of anchoring. The task of anchoring is to create and maintain the anchor $\alpha$ for a given object.

internal data structure, called an *anchor*, that uniquely represents a specific physical object. The anchor contains three elements: the *symbol* that denotes that object inside $\Sigma$; the *percept* from $\Pi$ that has been matched to that symbol; and a *perceptual signature* $\hat{\gamma}$ that gives the current estimate of the observable properties of the object.

**Definition 1** *An* anchor $\alpha$ *is any partial function from time to triples in* $\mathcal{X} \times \Pi \times \Gamma$.

The observable properties of an object can be used to *act* on the object (e.g., the observed position is needed in order to approach the object), or to *reidentify* the object at a later time. The elements described above are illustrated in Figure 3.

The task of the anchoring process is to create and maintain an anchor for each specific object of interest using the elements $\langle \Sigma, \Pi, g, \sigma, \gamma \rangle$ as basic ingredients.

### 3.2 The dynamical aspect

As we will shortly see, the initial creation of an anchor resembles a structural patter recognition process. Once an anchor has been created, however, this must be continuously updated to account for changes in the symbolic properties, the acquisition of new percepts in the perception stream, or the change of properties with time. This is especially important in a dynamic environment where objects move around. In general, updating is based on a combination of prediction and new observations, as illustrated in Figure 4. Prediction is used in order to make sure that the new percepts used in re-anchoring a symbol are compatible with the previous observations. In other words, we want to make sure that we are still tracking the same object. Comparison with the symbolic descriptor is used to make sure that the updated anchor still satisfies the predicated properties. In other words, we want to be sure that the object still has the properties that make it "the right one" for the goals of the symbolic system.

The main outcome of the update is the computation of new signature, which is stored in the anchor. At every time $t$, this signature provides an estimate of the

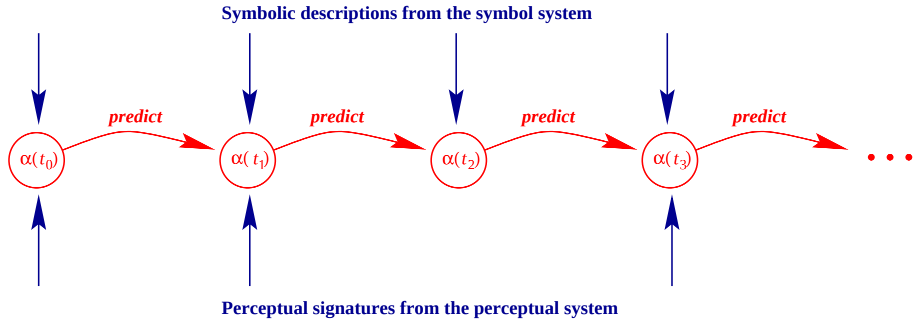



**Fig. 4.** Anchor dynamics. After its initial creation, the anchor is constantly updated by a combination of prediction and new observations.

observable properties of the object; these may be used, for instance, by a control system to guide action. When no matching percept is found, the signature stored in the anchor is only based on prediction.

The anchor dynamics resembles the usual predict-measure-update cycle of recursive estimators, e.g., of a Kalman filter. There is, however, an important difference: in updating the anchor, we also consider the abstract information (symbolic descriptor) provided by the symbol system. The experiment shown in Section 4.2 below illustrates a case where this information is crucial to a correct anchoring.

### 3.3 The functionalities of anchoring

The anchoring process described above can be defined by three abstract functionalities:

**Find** This functionality corresponds to the initial creation of an anchor for an object given its symbolic description. This functionality selects the adequate percept from the perceptual stream provided by the perceptual system $\Pi$ according to a domain-specific matching function, which uses the $g$ predicate grounding relation. If several matching percepts are found, then one of them is selected using a domain-specific selection function. (In [9] we consider the possibility of creating several anchors corresponding to several anchoring hypotheses.) This functionality is summarized in Figure 5.

**Track** This functionality corresponds to dynamical update of the anchor to take into account the passage of time and the arrival of new percepts in the perceptual stream. This functionality relies on domain-dependent functions for matching and selecting, as above, plus functions for predicting and updating the anchor's signature. This functionality is summarized in Figure 6.

**Reacquire** It is useful to distinguish the case where the object is kept under constant observation, which is solved by the Track functionality, from the

```
procedure Find (x)
   percept ← Select a percept such that its perceptual signature
               matches the symbolic description for x;
   if percept = null
      then fail
      else create an anchor storing the symbol x,
            the percept, and its perceptual signature
   return anchor
```

**Fig. 5.** Algorithm for a general FIND functionality.

case where the object is re-observed after some time. The Reacquire functionality takes care of this case. Although the general algorithm for Reacquire resembles the one for Track, the domain-dependent functions may be different. For instance, the Predict function may involve more complex reasoning about how the observed properties may have changed, and which ones of them should still be considered in the match.

Reacquire is somehow a combination of the Find and the Track functionalities. For example, consider a robot that has executed the action "PutDown(cup-22)," has gone to attend to another task, and needs later on to perform the action "PickUp(cup-22)." The robot needs to re-establish the anchor for "cup-22." This should not be achieved simply by a Find functionality, since the robot has some perceptual information about the cup (e.g., its shape and color as it was perceived, or its position) which may be more detailed than the symbolic descriptor for it. This cannot be achieved simply by the Track functionality either, since the prediction may be more complex (e.g., the lighting conditions may be different, thus making the previous color measurement difficult to use). Part of the prediction can be done inside the symbolic system $\Sigma$, e.g., by hypothetical reasoning. The result of the symbolic-level prediction would then be an updated symbolic description, which would then be fed into the Reacquire functionality.

In our experiments, we have found that FIND, TRACK, and REACQUIRE form a complete set of functionalities that is sufficient to solve the anchoring problem in all the cases that we have considered until now.

## 4 Using the model

Does our formal model contain all the basic elements which are needed in general to perform anchoring? In order to start answering this question, we have tried to use this model to solve the anchoring problem in several different autonomous robots, equipped with different symbolic planners and different perceptual components, and we have tested them in different domains.

```
procedure Track (x)
   anchor    ← current anchor for x
   signature ← Predict signature at current time from anchor
   percept   ← Select a percept such that its perceptual signature matches
               the symbolic description of x and the predicted signature
   if percept = null
      then Update anchor with signature
      else Update anchor by combining the predicted signature
           and the perceptual signature of percept
   return anchor
```

**Fig. 6.** Algorithm for a general TRACK functionality.

### 4.1 A robot navigation experiment

One such experiment was performed on a Nomad 200 mobile robot. The robot was controlled by a system comprising a symbolic module, consisting in a world model and a conditional planner, [18] and [19], a perceptual module using vision, and the Thinking Cap navigation system.[4] The architecture of the system is shown in Figure 7. The Symbol and perceptual systems are outlined. The task was to approach a small black suitcase with a white label. (The full task would involve fetching the suitcase, but our robot currently does not have a manipulator.) The initial set-up is shown in Figure 8 (left).

To perform this task, the planner is given the goal to be near a small black suitcase with a white mark:

```
(exists (?x)
   (and (suitcase ?x) (small ?x) (black ?x) (white-label ?x) (near ?x)))
```

where `?x` is a variable. The world model contains information about three suitcases, identified by the symbols `A`, `B`, and `C`: `A` is a large green suitcase, while `B` and `C` are small black ones. However, the world model does not contain any information about labels. Therefore, the planner generates a conditional plan:

```
((gonear C) (observe C)
 (if ((white-mark C . true)) (:success))
 (if ((white-mark C . false))
     ((gonear B) (observe B)
      (if ((white-mark B . true)) (:success))
      (if ((white-mark B . false)) (:fail)))))
```

What this plan says is the following: First go near suitcase `C` and look for a white label; if this is found, then we are done; otherwise, go near suitcase `B` and look for a white label; if this is found, then we are done; otherwise, we have failed. The reason why the robot needs to go near a suitcase before performing an `Observe` action is that the planner knows that labels can only be perceived from close by.

[4] The Thinking Cap is an autonomous robot architecture based on fuzzy logic [23]. This is a successor of the architecture originally developed for the robot Flakey [25].

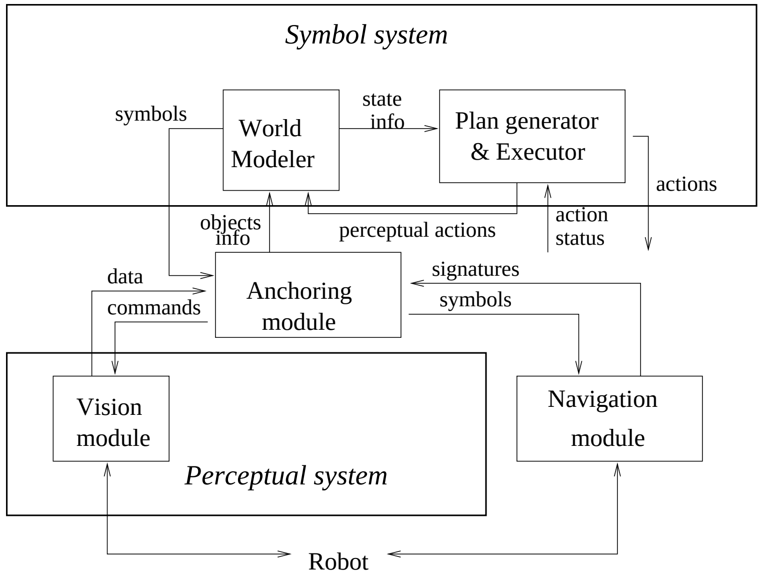



**Fig. 7.** System architecture in the robot navigation experiment.

After the plan has been generated, it must then be executed: here is where the anchoring problem arises. To execute the first action `(gonear C)`, the symbol `C` must be given a meaning from the point of view of the sensori-motoric system, that is, it must be anchored to specific sensor data using the Find functionality. These data are then used by the navigation system to direct the robot toward the correct physical object as perceived by the vision system.

Two types of information are available to the Find functionality: (1) the symbolic description of `C` provided by the world model in the symbol system: {`suitcase, small, black`}; and (2) the objects identified by the vision system (percepts), together with their perceptual signatures. In our case, the vision system identifies three percepts as suitcases and measures their size, color, and position, together with the presence and color of a label. To create an anchor for `C`, the Find selects a percept whose signature best matches the symbolic description of `C` based on the predicate grounding relation $g$.

Once the best matching percept is found, the anchor is created, and is is filled with the observed properties of the object, as measured on the percept. These properties are used to perform action: for instance, the position of the suitcase is used by the navigation system to perform the "gonear" action. During navigation, the suitcase is constantly tracked using the Track functionality, and the properties of the anchor (e.g., its position relative to the robot) are constantly updated.

In our experiment, the robot navigated to suitcase `C` but did not find any white label on it, so it had to execute the `(gonear B)` action. To do so, it turned

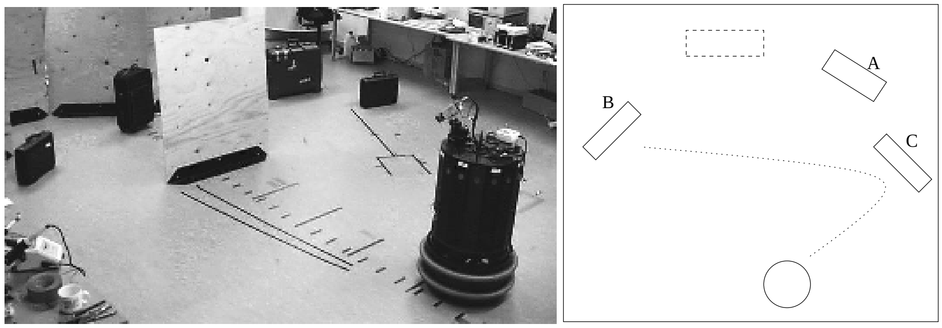



**Fig. 8.** Anchoring in action. To execute the action (`gonear C`) generated by the planner, the robot must anchor the symbol `C` to the correct suitcase in the environment.

toward the expected position of `B`, as stored in the world model, anchored `B` to the perceived suitcase, and used the observed position to precisely navigate in front of it. From there, the robot could actually observe a white label on the suitcase, so the task completed successfully, as shown in Figure 8 (right).

### 4.2 A UAV experiment

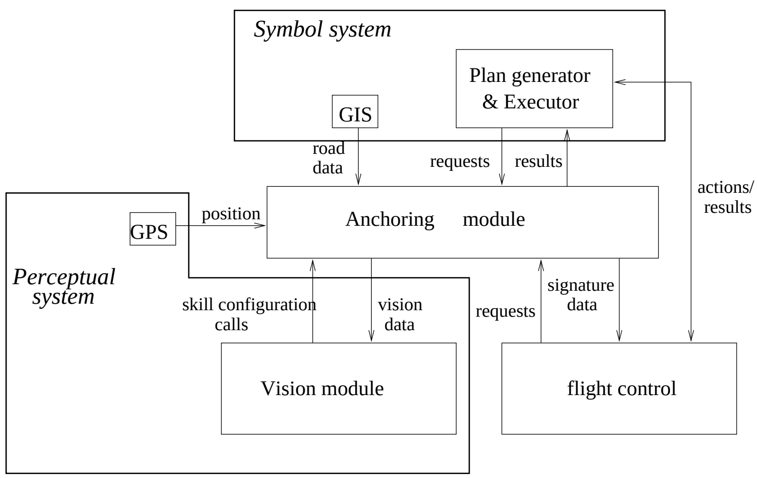
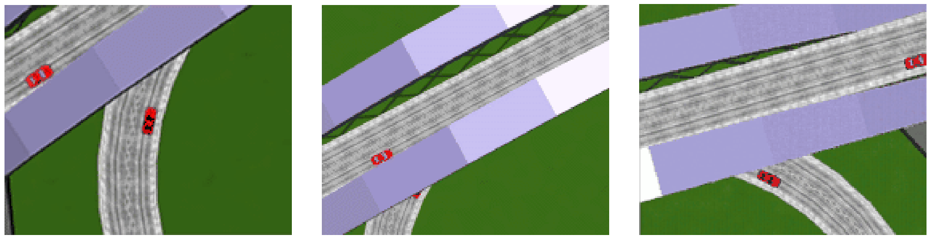



**Fig. 9.** Simplified view of the architecture of the WITAS system as it was when the experiment was performed (December 1999).

The following experiment stresses the dynamical aspect of the anchoring problem. This experiment, performed in the framework of the WITAS project



(a) (b) (c)

**Fig. 10.** A difficult case of anchor maintenance. The followed car disappears under a bridge and a similar car appears over the bridge.

[11], involves an autonomous helicopter performing traffic surveillance tasks in a simulated environment. The helicopter is controlled by a system that integrates a planner, a reactive plan executor, a vision system, and a flight control system. An anchoring modules connects the plan executor to the vision and the control systems. Figure 9 shows a simplified view of the architecture of the WITAS system as it was when the experiment was performed (December 1999). The helicopter is following a car that had been previously anchored (FIND). The flight control system takes the current position of the car from the signature in the anchor. The TRACK functionality is used to regularly update this anchor using the percepts provided by the vision system. In this experiment, we mainly relied on a Kalman filter for this functionality.

Where the experiment becomes more interesting is in the case shown in Figure 10. The helicopter is following the car in the middle of the image (a), which has already been anchored. At (b), our car disappears under the bridge. Simultaneously, an identical car appears on the bridge at the position in the image where the first car would have been, had it not been occluded by the bridge. Now, the TRACK routine predicts the perceptual signature of the first car's anchor. In particular, its expected position is extrapolated from its previous position and speed, and from higher level information derived from the symbol system concerning the road on which the car was traveling and the topology of the road network. The percept provided by the vision system is the image region containing the second car over the bridge. The attributes of this percept are compared to the predicted perceptual signature. Using high-level information, the percept is discarded: the intended object must be on the road passing under the bridge and not on the bridge — and we know that cars can only change roads at a crossing. Thus, the anchor of the first car is updated by prediction only, until an appropriate percept is found.

The flight controller can still use the predicted position in the anchor to maneuver the helicopter. However, because the continuous tracking has failed, the anchoring system now goes into the REACQUIRE modality. This modality uses a more complex prediction model, which also includes information about the road network and about possible car behaviors. This makes the predicted position of the car to be at the other end of the bridge. This prediction, stored in the anchor, is used to maneuver the helicopter and the camera accordingly. When eventually the first car reappears from under the bridge (c), a percept is generated which is compatible with the predicted signature, and the anchor is updated using this percept. We emphasize that the key to perform a correct anchoring in this example is the combined use of previously perceived attributes and symbolic domain knowledge in both prediction and matching.

## 5 Discussion

### 5.1 Work related to anchoring

Although anchoring as defined in this paper has not been the subject of previous rigorous investigation, issues related to anchoring have been discussed in the fields of autonomous robotics, machine vision, linguistics, and philosophy.

The autonomous robotics literature contains a few examples in which the need and the role of anchoring, under different names, has been explicitly identified (e.g., [15, 25]). Jung and Zelinsky [17] use a similar concept to achieve grounded communication between robots. None of these works, however, pursue a systematic study of the anchoring problem. Bajcsy and Košecká [1] offer a general discussion of the links between symbols and signals in mobile robotic systems. Yet, they do not deal with the problem of how to create and maintain these links, which is the main issue in anchoring.

The machine vision community has done much work on the problems of object recognition and tracking. While anchoring relies on these underlying perceptual abilities, it is mainly concerned with the integration of these with a symbol system. Some work in vision has explicitly considered the integration with symbols. Satoh *et al.* [27] present a system which associates faces and names in news videos looking at co-occurrences between the speech and the video streams. Horswill's Ludwig system [16] answers natural language queries by associating linguistic symbols to markers and to marker operations in the image space. Interestingly, Ludwig may refer to physical objects using indexical terms, like "the block on the red block." Markers are also used by Wasson *et al.* [29] to provide a robot with a perceptual memory similar to our anchors above. All the work, however, describe specific implementations, and they do not attempt a study of the general anchoring concept.

The problem of connecting linguistic descriptions of objects to their physical referents has been largely studied in the philosophical and linguistic tradition, most notably in the work by Frege and by Russell [12, 22]. In fact, we have borrowed the term *anchor* from situation semantics [2], where this term denotes an assignment of variables to individuals, relations, and locations. These traditions provide a rich source of inspiration for the conceptualization of the anchoring problem, but they typically disregard the formal and computational aspects necessary to turn these ideas into techniques.

Anchoring is related to two large research areas: pattern recognition and symbol grounding. Pattern recognition is the problem of how to recognize a pattern given sensory measurements. Symbol grounding [14] is the problem of how to give an interpretation to a formal symbol system that is based on something that, contrary to classical formal semantics, is not just another symbol system. Anchoring can be seen an important, concrete aspect that lays in the intersection of these research areas. It shares with symbol grounding the assumption that a symbolic system is present, which is not necessary the case in Pattern recognition; and it shares with pattern recognition the assumption that sensory measurements are used, which is not necessary the case in Symbol grounding. Moreover, anchoring focus on a specific problem: connecting symbol-level representations of individual objects to the perceptual image of these objects. While symbol grounding and pattern recognition are very wide problems for which there is little hope to find a "general" solution, anchoring is a more restricted problem for which we can hope to find a practical and general solution.

A peculiar aspect of anchoring is its reliance on internal representations that are uniquely associated to each object of interest, and which include all the sensor-level properties needed to operate on these objects. In our proposal, these representations are provided by the anchors, but other systems that perform anchoring also incorporate similar representations — see, for instance, the markers used in [16] or the PML-structures in [28]. Anchors provide perceptual handles to the actual objects denoted by symbols, so that we can perform physical actions on them. Anchors also provide a means to share a reference to a physical object between different sub-systems. In our examples, we have seen the use of anchors to share this reference between a symbolic planner, a vision system, and a motor control system.

### 5.2 Where to go next

The above treatment of anchoring must be seen as a starting point. Anchoring hides a number problems that need to be carefully investigated. Some of these problems are technical in nature, and their solution is needed in order to apply any theory of anchoring to complex domain. For instance, perceptual information is typically affected by *uncertainty*, and this uncertainty should be taken into account in the anchoring process. Moreover, the predicates used in symbolic descriptions can be inherently *vague*. This is especially true for many predicates commonly used in linguistic human-robot communication, like "red" or "large." In our actual implementations, we have accounted for both these factors using fuzzy logic [6] and [5], but other solutions may be possible. Furthermore, the meaning of many predicates in terms of physical quantities is highly context dependent, as in the case of the predicate "red" when referred to wine. An interesting possibility to address some of these problems would be to frame the predicate grounding problem in the context of Gärdenförs' conceptual spaces [13]and [4]. Finally, in cases of perceptual ambiguity we may need to maintain *multiple hypotheses* for the anchor. Multiple hypotheses are also needed in the presence of partial matching, that is, when some of the attributes corresponding to the required predicated cannot be observed [9].

Other problems are more conceptual, and they are related to some subtle issues in the definition of anchoring. One such issue is the distinction between *definite* descriptions, like "the cup of coffee on the table," and *indefinite* descriptions, like "a cup of coffee." These descriptions need different treatments, e.g., if the robot sees two cups on the table. A second important issue is the origin of the *g predicate grounding relation*. In our example, this relation was built-in by the system designer, but this may not be adequate in other applications. For instance, in the case of the artificial nose mentioned in our opening scenario, there is no well-established numerical model of the sensor. Therefore, the relation between predicates denoting properties that relate to smell and the actual measurements acquired from this sensor has to be learned rather than given a priori. A preliminary experiment in this sense is reported in [20], where an artificial neural network has been trained to recognize vanilla-, lavender-, and yogurt-like aroma using an artificial nose. Finally, the information contained in the anchor can be used to reason on the use of *perceptual resources*. In both experiments reported above we have used the expected properties of the object, stored in the anchor's signature, to direct the video camera and to parameterize the vision routines. In another application [26], we have used this information to select the focus of attention when tracking multiple objects.

## 6 Conclusions

The problem of perceptual anchoring can be extraordinary complex, and many of its subtleties have been the object of much thought throughout the history of philosophy. Nonetheless, if we want to build a physically embedded agent that incorporates a symbolic component, we have to solve (a specific instance of) it. This is true in particular for Plan-based robotic systems. In this paper, we have advocated the study of a theory of anchoring that is general enough to allow solutions to be ported across different systems, but still specific enough to be manageable.

The quest for such a theory is still in its initial phase, but it is already elicitating much interest [8] and [10]. It will probably have an inter-disciplinary flavor, combining insights from the study of symbol grounding, estimation theory, belief revision, pattern recognition, and still others. Inter-disciplinary will also be needed because a general theory of anchoring must be solidly grounded in experiments performed on many different systems operating in many different domains. We believe that having such a theory will greatly advance our ability to build intelligent embedded systems.

*Acknowledgements* This work was funded by the Swedish KK Foundation. We thank Lars Karlsson and Zbigniew Wasik for providing substantial help in running the robot experiments.

## References


1. R. Bajcsy and Košecká. The problem of signal and symbol integration: a study of cooperative mobile autonomous agent behaviors. In *Proceedings of KI-95: Advances in Artificial Intelligence, 19th Annual German Conference on Artificial Intelligence*, volume 981 of *LNCS*, pages 49–64, Berlin, Germany, 1994. Springer.
2. J. Barwise and J. Perry. *Situations and Attitudes.* The MIT Press, 1983.
3. W. Burgard, A. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2):3–55, 1999.
4. A. Chella, M. Frixione, and S. Gaglio. Anchoring symbols on conceptual spaces: the case of dynamic scenarios. *To appear in [10]*, 2003.
5. S. Coradeschi, D. Driankov, L. Karlsson, and Saffiotti A. Fuzzy anchoring. In *In Proc. of the 10th IEEE International Conference on Fuzzy Systems*, Melbourne, December 2001.
6. S. Coradeschi and A. Saffiotti. Anchoring symbols to vision data by fuzzy logic. In *Qualitative and Quantitative Approaches to Reasoning with Uncertainty*, LNAI, pages 104–115. Springer, Berlin, Germany, 1999.
7. S. Coradeschi and A. Saffiotti. Anchoring symbols to sensor data: preliminary report. In *Proc. of the 17th National Conference on AI (AAAI-2000)*, pages 129–135, 2000. Online at http://www.aass.oru.se/~asaffio/.
8. S. Coradeschi and A. Saffiotti, editors. *Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems. Papers from the 2001 AAAI Fall Symposium*, Technical Report FS-01-01. AAAI, 2001.
9. S. Coradeschi and A. Saffiotti. Perceptual anchoring of symbols for action. In *Proc. of the 17th IJCAI Conf.*, pages 407–412, Seattle, WA, 2001. Online at http://www.aass.oru.se/~asaffio/.
10. S. Coradeschi and A. Saffiotti, editors. *Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems. Special issue of the Robotics and Autonomous Systems journal.*, 2003.
11. P. Doherty, G. Granlund, K. Kuchcinski, E. Sandewall, K. Nordberg, E. Skarman, and J. Wiklund. The WITAS unmanned aerial vehicle project. In *Proc. of the European Conf. on AI (ECAI)*, Berlin, 2000.
12. F.L.G. Frege. Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, pages 25–50, 1892.
13. P. Gärdenförs. *Conceptual Spaces: The Geometry of Thought.* MIT Press, 2000.
14. S. Harnard. The symbol grounding problem. *Physica D*, 42:335–346, 1990.
15. H. Hexmoor, J. Lammens, and S. C. Shapiro. Embodiment in GLAIR: A grounded layered architecture with integrated reasoning for autonomous agents. In D. Dankel, editor, *Proceedings of the Florida AI Research Symposium*, pages 325–329, 1993.
16. I. Horswill. Visual architecture and cognitive architecture. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2):277–292, 1997.
17. D. Jung and A. Zelinsky. Grounded symbolic communication between heterogeneous cooperating robots. *Autonomous Robots*, 8(3):269–292, 2000.
18. L. Karlsson. Conditional progressive planning: a preliminary report. In *Proc. of the 17th IJCAI Conf.*, Seattle, WA, 2001.
19. Karlsson L. and T. Schiavinotto. Progressive planning for mobile robots — a progress report. In *This volume*. Springer, 2002.

20. A. Loutfi, S. Coradeschi, T. Duckett, and P. Wide. Odor source identification by grounding linguistic descriptions in an artificial nose. In *Proc. of the SPIE conference on Sensor Fusion*, Orlando, 2001.
21. N. Muscettola, P. Nayak, B. Pell, and B. Williams. Remote agent: To boldly go where no ai system has gone before. *Artificial Intelligence*, 103(1-2):5–48, 1988.
22. B. Russell. On denoting. In *Mind XIV*, pages 479–493. 1905.
23. A. Saffiotti. The Thinking Cap home. www.aass.oru.se/~asaffio/Software/TC/.
24. A. Saffiotti. Pick-up what? In C. Bäckström and E. Sandewall, editors, *Current trends in AI Planning*, pages 266–277. IOS Press, Amsterdam, Netherlands, 1994.
25. 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.
26. A. Saffiotti and K. LeBlanc. Active perceptual anchoring of robot behavior in a dynamic environment. In *IEEE Int. Conf. on Robotics and Automation*, pages 3796–3802, 2000.
27. S. Satoh, Y. Nakamura, and T. Kanade. Name it: naming and detecting faces in video by integration of image and natural language processing. In *Proc. of the 15th IJCAI Conf.*, pages 1488–1493, 1997.
28. S.C. Shapiro and O.H. Ismail. Symbol-anchoring in cassie. In S. Coradeschi and A. Saffiotti, editors, *Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems: Papers from the 2001 AAAI Fall Symposium*, pages 2–8. AAAI, 2001.
29. G. Wasson, D. Kortenkamp, and E. Huber. Integrating active perception with an autonomous robot architecture. *Robotics and Automation Journal*, 1999.