Some Similarities Between Anchoring and Pattern Recognition Concepts

Isabelle Bloch
Ecole Nationale Sup. des Télécommunications
Dept. TSI - CNRS URA 820
46 rue Barrault, F-75013 Paris, France
isabelle.bloch@enst.fr

Alessandro Saffiotti
Center for Applied Autonomous Sensor Systems
Dept. of Technology, Örebro University
S-70182 Örebro, Sweden
alessandro.saffiotti@aass.oru.se

Abstract

The notion of anchoring has been introduced as a special form of symbol grounding which is needed in practical robotic systems that comprise a symbolic reasoning component. However, several concepts used in anchoring are also familiar to the field of pattern recognition. The goal of this note is to set the stage for a comparison between anchoring and pattern recognition, by pointing at some of the common points between the two notions.

Introduction

Anchoring has been introduced in the context of hybrid architectures for mobile robotics as a key notion to connect the higher and lower layers. As stated in (Coradeschi & Saffiotti 2000),

> *anchoring is the process of creating and maintaining the correspondence between symbols and sensor data that refer to the same physical object.*

When looking more precisely as the main ingredients that compose this complex notion, some similarities can be noted with the ones appearing in the completely different field of pattern recognition. Indeed, although originally different in nature, anchoring and pattern recognition involve rather similar steps and concepts. This paper is an attempt to highlight these similarities, as a starting point for a wider discussion about the similarities and differences between the two notions, both at the conceptual and the operational levels, that can help to better characterize the nature of anchoring problem.

Anchoring: general principles

The anchoring problem arises in any intelligent system embedded in the physical world that incorporates a symbolic representation and reasoning process. This process will use symbols to denote objects in the world that are relevant to the performance of the system tasks: for instance, it may use the symbol 'mycup' to denote a specific cup that should be grasped. The fact that the system is embedded in the physical world means that it also includes sensori-motoric processes that can observe and manipulate the actual objects in the world: for instance, the specific cup to grasp. In order to consistently perform the intended actions, this system must make sure that the symbolic process and the sensori-motoric process "talk about" the same physical objects. In our example, it must make sure that the symbol 'mycup' is correctly linked to the sensor data that originate from the intended cup. Said differently, the system must *anchor* this symbol to the right sensor data.

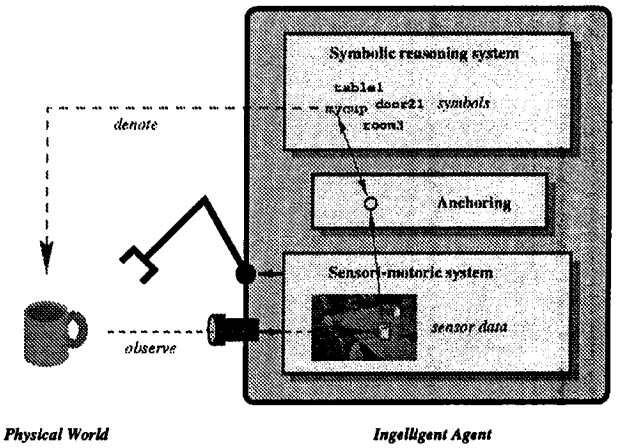

Figure 1: The anchoring problem.

Figure 1 illustrates the problem of anchoring. The intelligent system, or agent, comprises a *symbol system* and a *perceptual system*. The correspondence between the symbol used to denote the cup to grasp and the sensor (image) data originating from that cup is represented, or reified, in a data structure called *anchor*. Importantly, the anchor contains a *signature* that gives the current best estimate of the observable properties of the object, which may be needed both for acting on the object (e.g., its position) and for re-identifying it later on. The anchoring problem is the problem of creating and maintaining anchors.

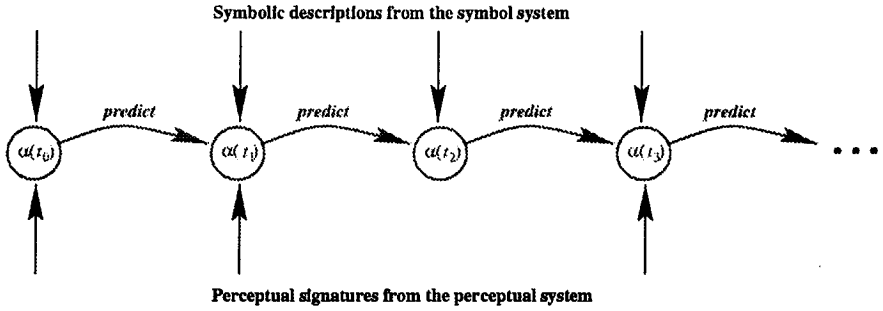

Figure 2: Updating of an anchor.

In (Coradeschi & Saffiotti 2000), the authors propose a computational model of anchoring that comprises three main ingredients: a symbol system, a perceptual system, and a *predicate grounding relation*. The latter encodes the correspondence between observable features, like the average RGB value, and symbolic predicates, like 'red'. No assumption is made on the origin of this relation: for instance, it can be hand-coded by the system designer, or it can be learned from a number of samples. The task of anchoring is to use this relation to connect symbols that denote objects, and individual percepts. For instance, suppose that 'red' is predicated of 'mycup', and that the average RGB values of a given region in an image match the predicate 'red' according to the predicate grounding relation. Then that region could be anchored to the symbol 'mycup'.[1]

In practice, the task of anchoring has been decomposed into three main functionalities: FIND, that establishes an anchor for a symbol the first time; TRACK, that maintains this anchor with time while the object is kept under continuous observation; and REACQUIRE, that re-establishes an anchor for an object that has not been observed for some time. The FIND functionality operates as in the example above.

The TRACK and REACQUIRE functionalities continuously update the anchor when new percepts are observed, or simply to account for the passage time. (This is especially important in dynamic environments, where objects may move around.) Updating is based on a combination of prediction and new observations, as illustrated in Figure 2. The main outcome of the update is the computation of new signature, which may be used, for instance, to guide action, or for future matches.

There are more subtleties involved in the notion of anchoring, including the possibility of partial matching, of uncertain matching, and of combining anchoring with active perception. We address the reader to the above cited papers for more on these issues. For the goals of this note, however, the basic elements given here suffice.

[1]In a more realistic example, several other properties will be true of 'mycup' and should thus be checked in the percept, including the fact of being cup-shaped.

Pattern recognition: general principles

Pattern recognition can be expressed in a very general way as a representation and decision problem for interpreting data provided by sensors (Duda & Hart 1973; Dubuisson 1990; Pavel 1993). Interpretation can be understood as categorizing a perceived phenomenon and attributing it to a family of similar phenomena or to an individual known phenomenon (generic versus specific object recognition). Given a pattern or a form, the problem amounts to find to which class (or prototype, or typical form) it has to be attached, or, the other way around, find in a set of data (in an image for instance) the object that corresponds to a given class. A system performing this task should have learning and adaptivity capabilities. A key point is the choice of the representation.

Usually two types of approaches are distinguished, although this classification is far from being crisp. The first class consists of statistical pattern recognition. A pattern is considered as a point in a representation space defined by a set of features or parameters. Prototypes are points or regions of this space which are representative of the classes.

The second class consists of structural pattern recognition. This approach is also called descriptive, and the description should also reflect the structure of the scene or of the patterns and classes, in a symbolic form.

The most important concepts in pattern recognition are:

- Forms or patterns: these are the actual entities to be recognized.
- Classes and prototypes: known or learned models of the objects to be recognized (can be generic or specific: for instance in brain imaging a generic class can be all grey matter structures, while a specific one can be the right caudate nucleus. Both can be considered simultaneously or successively, in order to focus progressively on more specific objects, see e.g. (Géraud, Bloch, & Maître 1999)).
- Knowledge, information, observations: usual distinctions between these terms occur also in pattern recognition. Observations are related to facts, at a given

time, and deliver what is called accessible information, which can be of numerical or symbolic nature. Global or prior knowledge about classes and prototypes relies on the past, and is often incomplete. *Features* is the most often used term for designating information extracted from the data on which the recognition will be based (typically it concerns characteristics of the objects that are relevant for the recognition purpose).

- Representation space: its choice is crucial, it should be informative enough to allow for a good precision during the identification step, and condensed enough to avoid unnecessary redundancy; this raises the classical problem of antagonism between precision and exactness on the one hand, and mathematical and algorithmical possibilities on the other hand.
- Invariance under transformation, equivalence with respect to transformations or metrics: these concepts are important when considering that the observed objects can be deformed with respect to the model (this can be simply a translation or a rotation, but also deformations induced by the acquisition system, by the view point, by artifacts or noise, and so on).
- Learning: a very important feature of a pattern recognition system. It aims at defining the classes or the prototypes in the representation space.
- Adaptivity: a requirement for dealing with incomplete knowledge, with systems evolving in time, etc.; we should carefully note that this term has a very broad meaning and may concern the model or its parameters, as well as the observations, or even the correspondence between models and observations.
- Interpretation and decision: the core of the recognition. It establishes the correspondence between observed objects and prototypes or classes. If characteristics of data and of classes are mapped in a common representation space, the problem amounts to define similarity functions, equivalence relations, boundaries between classes, etc. If the representation spaces are different for models and for data, the problem is more difficult and requires to define correspondence functions between both types of representations before decision rules can be designed.
- Reject: this notion allows not to decide (or to postpone the decision until more information is available) either if the object is ambiguous, midway between several classes (reject in ambiguity) or to far from all prototypes (reject in distance), the second class leading typically to creation of new classes or to no decision at all. Attention should be paid to the fact that rejected objects do not form a class, but possibly several ones (or just noise) and therefore reject can not be treated in the same way as any other class.

Let us mention two domains where adaptivity is required. The first one concerns active vision and active object recognition (Aloimonos, Weiss, & Bandyopadhyay 1987; Bajcsy 1988), where the purpose is to actively determine the identity of an object (or search for a particular object in a scene) by acquiring successive information in order to resolve ambiguities. This often also involves a fusion step for merging the pieces of information when they become available. Examples can be found in (Hutchinson & Kak 1992; Kovacic, Leonardis, & Pernus 1998).

The second example concerns diagnosis of complex systems (Dubuisson 1990), which aims at assessing the functioning mode of a system as well as its evolution in time. Adaptivity is required to account for possible new functioning modes or to update characteristics of a known mode. Recognizing functioning modes includes state estimation. Diagnosis of complex systems can be seen from the point of view of pattern recognition and the reference (Dubuisson 1990) entirely deals with this point of view. Therefore, diagnosis, state estimation, and assessing the functioning mode of a complex dynamic system can be considered as examples of pattern recognition problems.

Parallels between anchoring and pattern recognition

From the previous description of both fields we can exhibit some similarities. We can see anchoring as a pattern recognition problem, where the models are specific (rather than generic), and where the recognition process is model-driven rather than data-driven, i.e. we do not start from the observed object and try to attach it to a class, but we look at an object that matches the description of one specific model. The two ways of reasoning are not symmetrical, since when starting from the model, the object can be sensed in order to gather features that are relevant for recognizing this particular class. For instance if the model is described in terms of shapes, it might be unnecessary to acquire information about color.

The symbol system can be assimilated to the classes and prototypes, while the perceptual systems can be assimilated to the objects to be recognized. In both domains, classes and objects are described in terms of features, characteristics, attributes. Knowledge is attached to the symbol system, while observations are attached to the perceptual system. Anchoring focuses on symbolic descriptions of individual models, which corresponds to problems in pattern recognition where models and observations are given under different representations. The predicate grounding relation corresponds to the notion of similarity used in pattern recognition (in this case between a symbolic predicate and an attribute value), or mapping between different representations. The notion of representation space is also a key point in anchoring.

The notion of invariance under transformations, or of equivalence with respect to some transformations is not explicitly mentioned in the anchoring concepts. However, it can be argued that this is implicitly included in the symbolic descriptions, as well as in the notion of an-

chor signature. For the signature, in many cases these will be implicitly assumed to be independent of the view point. And for the description, in most cases the fact that the model description is symbolic automatically implies invariance under some types of transformations (like geometrical ones), as it appears in the examples given in (Coradeschi & Saffiotti 2000).

Learning can play several roles in anchoring, as discussed in (Coradeschi & Saffiotti 2001), although this point is still at a very preliminary stage of development.

Interpretation and decision correspond to the task FIND in anchoring, while reject can be assimilated to the conclusion FAIL (see algorithm in (Coradeschi & Saffiotti 2000)). This notion is not much developed in the work on anchoring, and it could be interesting to see if the distinction between reject in ambiguity and reject in distance could bring something to this domain.

An essential aspect of anchoring is its dynamic side, that is, the fact that we must maintain the symbol-percepts correspondence as time evolves. This aspect is provided by the TRACK and REACQUIRE functionalities. It is still unclear in our opinion if this aspect can find a direct correspondent in pattern recognition. On the one hand, these functionalities are related to the notion of adaptivity, since they modify the internal signature in the anchor which can be interpreted as a model of the object (pattern) that we are trying to track (recognize) — and this signature will be used for matching new percepts. On the other hand, these functionalities do more than simply updating the internal signature in the anchor: they perform tasks like data association (decide which percepts to use to update each anchor), state estimation (computing the best estimate of the observable properties), and data filtering using high-level information — see the examples given in (Coradeschi & Saffiotti 2000). Moreover, the the anchor signature is not simply concerned with the matching process, but contains for instance information needed by the controller for operating on the object.

Discussion

Anchoring, as defined in (Coradeschi & Saffiotti 2000), must necessarily take place in any intelligent embedded system that comprises a symbolic component. However, only recently people have attempted to state anchoring as a problem *per se*, and the definition of anchoring is still somehow elusive. We hope that the above notes may contribute to the discussion at this Symposium, leading to a more precise characterization of the anchoring problem, of its relations to other problems, and of its peculiarities.

In particular, establishing the differences and similarities between anchoring and pattern recognition is important in order to assess which results and experience could be profitably transferred, and which complementarities could be exploited. An especially interesting aspect is the possibility to transfer ideas and techniques from anchoring to pattern recognition about the use of high-level heuristics and knowledge, including human knowledge.

Although several aspects of anchoring have corresponding concepts in pattern recognition, the dynamic aspects of it seem to be closer to other fields like (recursive) state estimation and data association. We can claim that anchoring needs to incorporate techniques both from pattern recognition and from state estimation, but it puts a greater emphasis on the close integration between these and on the use of symbolic models.

References

Aloimonos, J.; Weiss, I.; and Bandyopadhyay, A. 1987. Active Vision. In *First International Conference on Computer Vision*, 35–54.

Bajcsy, R. 1988. Active Perception. *Proceedings of the IEEE* 76(8):996–1005.

Coradeschi, S., and Saffiotti, A. 2000. Anchoring Symbols to Sensor Data: Preliminary Report. In *Proc. of the 17th AAAI Conf.*, 129–135. Menlo Park, CA: AAAI Press. Online at http://www.aass.oru.se/~asaffio/.

Coradeschi, S., and Saffiotti, A. 2001. On the Role of Learning in Anchoring. In *AAAI Spring Symposium on Learning Grounded Representations.*

Dubuisson, B. 1990. *Diagnostic et reconnaissance des formes.* Paris: Hermès.

Duda, R., and Hart, P. 1973. *Pattern Classification and Scene Analysis.* New-York: Wiley.

Géraud, T.; Bloch, I.; and Maître, H. 1999. Atlas-guided Recognition of Cerebral Structures in MRI using Fusion of Fuzzy Structural Information. In *CIMAF'99 Symposium on Artificial Intelligence*, 99–106.

Hutchinson, S. A., and Kak, A. C. 1992. Multisensor Strategies using Dempster-Shafer Belief Accumulation. In Abidi, M. A., and Gonzalez, R. C., eds., *Data Fusion in Robotics and Machine Intelligence.* Academic Press. 165–309.

Kovacic, S.; Leonardis, A.; and Pernus, F. 1998. Planning Sequences of Views for 3D Object Recognition. *Pattern Recognition* 31(10):1407–1417.

Pavel, M. 1993. *Fundamentals of Pattern Recognition.* New-York: Marcel Dekker.