Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Robotics and Autonomous Systems 43 (2003) 83

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Foreword

The anchoring problem is an important aspect of the connection between symbolic and sensory based processes in autonomous robotic systems. Anchoring is in fact the problem of how to create, and maintain in time, the connection between the symbol- and the signal-level representations of the same physical object. An example is the problem of connecting, inside a mobile robot, the symbol used by a planner to refer to a particular person to follow, say John, to the sensor data that correspond to that person in the robot's vision system. This connection must be dynamic since the same symbol must be associated to new entities in the perceptual stream in order to track the object over time or re-acquire it at a later moment.

Although anchoring must necessarily occur in any physically embedded system that comprises a symbolic reasoning component, most current solutions to the anchoring problem are developed on a system by system basis, and the solution is often hidden in the code. This is unfortunate, since having a general theory of anchoring would greatly advance our ability to build intelligent embedded systems and transfer techniques and results across different systems.

On these grounds, we have organized in November 2001 a symposium on anchoring within the AAAI Fall Symposium Series (http://www.aass.oru.se/Agora/FSS01). The ambition of that symposium was to create an interdisciplinary community that will eventually develop a general theory of anchoring. The symposium showed that there is a growing interest around the anchoring problem in the robotics and artificial intelligence communities. It also showed that there starts to be a critical mass of work related to the anchoring problem, but that this work tends to be scattered across different scientific communities and different topics. The aim of this special issue is to collect in one place relevant pieces of work that can be instrumental in building such a general theory of anchoring.

In putting together this special issue, we received substantial help by several people. Andreas Birk, Isabelle Bloch, Andrea Bonarini, Antonio Chella, Dimiter Driankov, Tom Duckett, Marcello Frixione, Joachim Hertzberg, Ian Horswill, David Jung, Lars Karlsson, Kurt Konolige, Benjamin Kuipers, Yves Lespérance, Lisa Meeden, Illah Nourbakhsh, Erich Prem, Murray Shanahan, Stuart Shapiro, Josefina Sierra Santibanez, Monique Thonnat, Thierry Vidal, Paul Vogt, and Holly Yanco helped us in carefully reviewing the many submissions received. We are grateful to Frans Groen, the Editor of this journal, for offering us this opportunity. Finally, we wish to thank the participants in the above mentioned symposium. This special issue would not exist if it were not for the enthusiasm and the exciting discussions seen at the symposium.

Silvia Coradeschi, Alessandro Saffiotti
AASS Mobile Robotics Lab
Department of Technology, Örebro University
S-70182 Örebro, Sweden
E-mail addresses: silvia.coradeschi@aass.oru.se
(S. Coradeschi)
asaffio@aass.oru.se (A. Saffiotti)

0921-8890/03/$ – see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0921-8890(03)00020-4

ELSEVIER

Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Robotics and Autonomous Systems 43 (2003) 85–96

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

An introduction to the anchoring problem

Silvia Coradeschi*, Alessandro Saffiotti

Department of Technology, Center for Applied Autonomous Sensor Systems, Örebro University,
Fakultetsgatan 1, S-70182 Örebro, Sweden

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 highlight some of its subtle points.

Keywords: Perceptual anchoring; Cognitive robotics; Embedded AI; Symbol grounding; Reference; Tracking

1. Introduction

> *For things to exist there are two essential conditions, that a man should see them and be able to give them a name* [19, p. 53].

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 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 that embody two different types of processes: one that reasons about abstract representations of objects, and the other 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, i.e., 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 [4,17].

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

[*] Corresponding author.
E-mail addresses: silvia.coradeschi@aass.oru.se (S. Coradeschi), alessandro.saffiotti@aass.oru.se (A. Saffiotti).

0921-8890/03/$ – see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0921-8890(03)00021-6

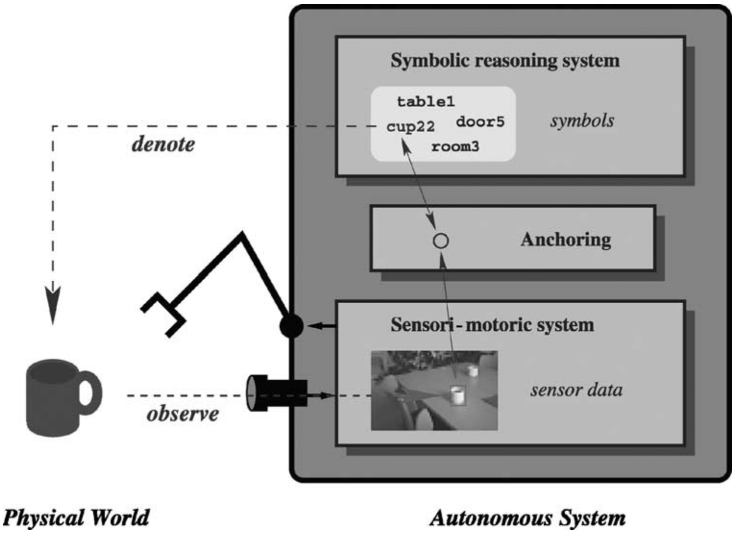


Fig. 1. Graphical illustration of the anchoring problem.

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, i.e., 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 discuss possible solutions to the anchoring problem in its different facets and 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 Section 1, 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 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 [21], 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 Section 1, 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 support prediction of the future. It can be used to track the object and 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 [13] use markers, Shapiro and Ismail [20] use PML-descriptions, and Fritsch et al. [11] use a hierarchy of anchors.

An important aspect of anchors is that they can be shared across different subsystems of the agent in order to provide them with a *common handle* to refer to a specific physical object. In the example given in Section 1, 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 [13], engage in communicative actions [20], achieve a shared language [21,22], and 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 [14]. 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. [11] 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 the 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* [3,21]. 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 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 problems—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 a 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 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. [11]. 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 to not only anchor symbols internally but also exchange information among them and 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 [13,21,22] or between robots and humans [2].

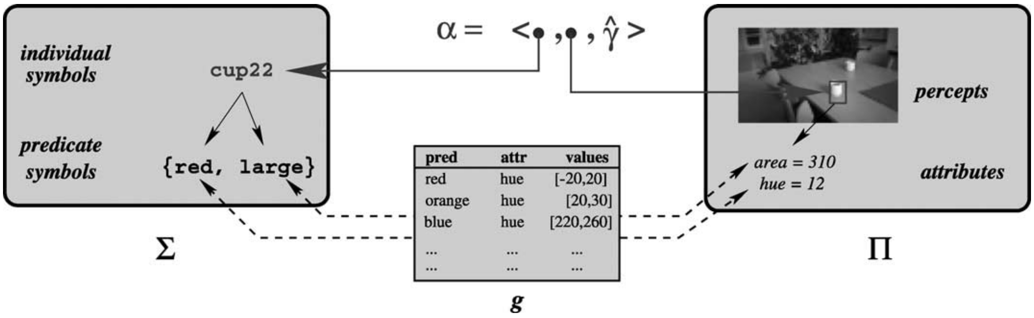
Finally, a fundamental challenge of the anchoring problem is to investigate the formal properties of the anchoring process. Intuitively one may feel that some correspondences between the symbols and the sensor data are correct while some are not. How to express this formally, and prove the correctness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

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 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 examples. A detailed description of this framework and 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



pred	attr	values
red	hue	[-20,20]
orange	hue	[20,30]
blue	hue	[220,260]
...	...	...
...	...	...

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

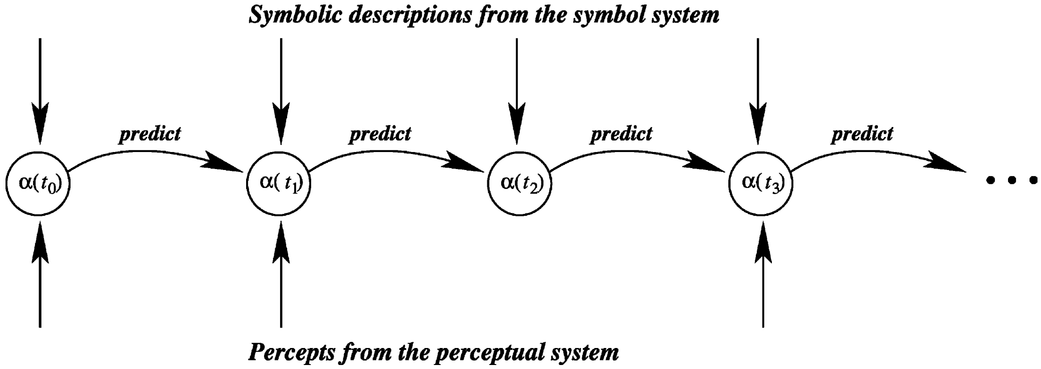

Symbolic descriptions from the symbol system
predict
predict
predict
predict
α(t0)
α(t1)
α(t2)
α(t3)
• • •
Percepts from the perceptual system

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

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 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 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 for both acting on the object and 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 subsystems 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, i.e., 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, i.e., 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

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

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 to 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 [18], 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 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 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

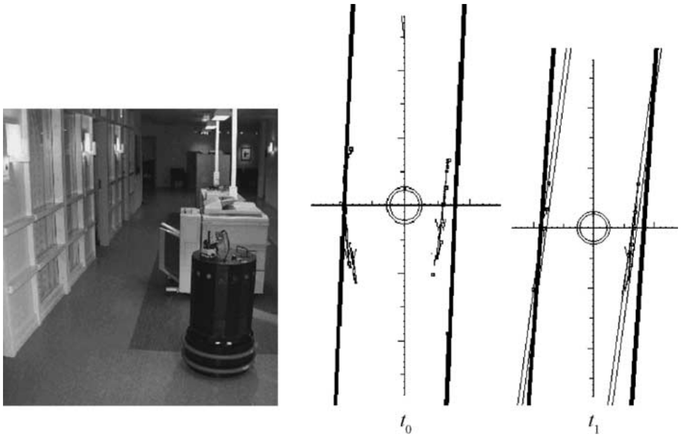


Fig. 4. Anchoring a corridor: t_0 (before anchoring); t_1 (after anchoring).

autonomous surveillance tasks in a simulated environment developed within the WITAS project [8]. 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, control the camera, and 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 the other one traveling on the bridge. The UAV is 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 predict the next visible position of the car. This position is stored in the signature of the anchor, and used to direct the UAV and the camera towards 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.

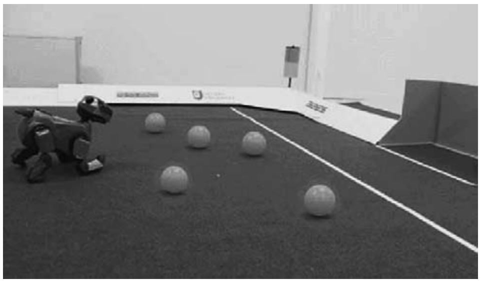


Fig. 6. Anchoring “a red ball” to perform a ball collection task.

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 multiple identical objects [7]. The task is one of the three “technical challenges” of the RoboCup 2002 competition in the Sony four-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 : Ball$(x) \wedge$ 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

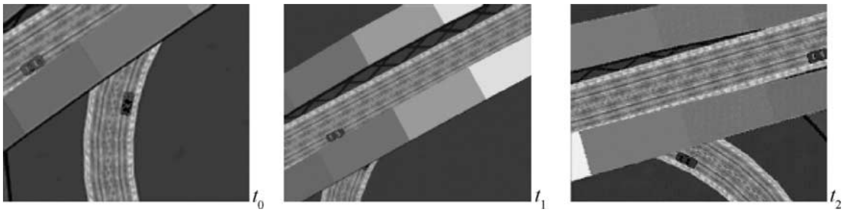


Fig. 5. Anchoring a moving object. The followed car disappears under a bridge and a similar car appears at its place over the bridge.

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 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 [16] and Frege [10]. 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 implement 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 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 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 [9,15]. Taking pattern recognition in its most general sense, anchoring can be considered a sub-problem 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 subsystems 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 [12]. 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 an 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

[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.

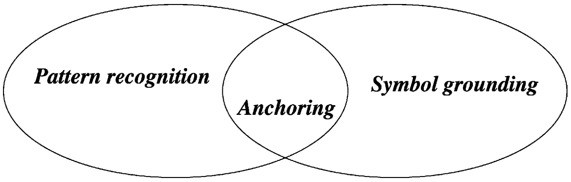


Fig. 7. Relations among Anchoring, Symbol grounding and Pattern recognition.

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 [21] in this issue, 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 Section 1, although some of them deal with anchoring intended in a somewhat wider sense.

Shapiro and Ismail [20] 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 [13] 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. [11] 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 [22] 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 [21] 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 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, J. Perry, Situations and Attitudes, MIT Press, Cambridge, MA, 1983.

[2] N. Bredeche, Y. Chevaleyre, J.D. Zucker, A. Drogoul, G. Sabah, A meta-learning approach to ground symbols from visual percepts, Robotics and Autonomous Systems 43 (2003) 149–162 (this issue).

[3] A. Chella, M. Frixione, S. Gaglio, Anchoring symbols on conceptual spaces: the case of dynamic scenarios, Robotics and Autonomous Systems 43 (2003) 175–188 (this issue).

[4] S. Coradeschi, A. Saffiotti, Anchoring symbols to sensor data: preliminary report, in: Proceedings of the 17th National Conference on Artificial Intelligence (AAAI), Austin, TX, 2000, pp. 129–135. http://www.aass.oru.se/Research/Robots/.

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

[6] S. Coradeschi, A. Saffiotti, Perceptual anchoring of symbols for action, in: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI), Seattle, WA, 2001, pp. 407–412. http://www.aass.oru.se/Research/Robots/.

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

[8] 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.

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

[10] G. Frege, Über Sinn und Bedeutung, Zeitschrift für Philosophie und philosophische Kritik, 1892, pp. 25–50.

[11] J. Fritsch, M. Kleinehagenbrock, S. Lang, T. Plötz, G.A. Fink, G. Sagerer, Multi-modal anchoring for human–robot interaction, Robotics and Autonomous Systems 43 (2003) 133–147 (this issue).

[12] S. Harnad, The symbol grounding problem, Physica D 42 (1990) 335–346.

[13] A. Khoo, I. Horswill, Grounding inference in distributed multi-robot environments, Robotics and Autonomous Systems 43 (2003) 121–132 (this issue).

[14] D.C. Marr, Vision, Freeman, San Francisco, CA, 1982.

[15] M. Pavel, Fundamentals of Pattern Recognition, Marcel Dekker, New York, 1993.

[16] B. Russell, On denoting, Mind XIV (1905) 479–493.

[17] A. Saffiotti, Pick-up what? in: C. Bäckström, E. Sandewall (Eds.), Current Trends in AI Planning, IOS Press, Amsterdam, 1994, pp. 266–277.

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

[19] J. Saramago, The Stone Raft, Harcourt, Brace and Jovanovich, New York, 1995.

[20] S.C. Shapiro, H.O. Ismail, Anchoring in a grounded layered architecture with integrated reasoning, Robotics and Autonomous Systems 43 (2003) 97–108 (this issue).

[21] L. Steels, J.C. Baillie, Shared grounding of event descriptions by autonomous robots, Robotics and Autonomous Systems 43 (2003) 163–173 (this issue).

[22] P. Vogt, Anchoring of semiotic symbols, Robotics and Autonomous Systems 43 (2003) 109–120 (this issue).

Silvia Coradeschi is an Assistant Professor at the Center for Applied Autonomous Sensor Systems at Örebro University, Sweden. She has received a Masters degree in Philosophy at the University of Florence, Italy, a Masters degree in Computer Science at the University of Pisa, Italy, and a Ph.D. in Computer Science at Linköping University, Sweden. She is a Member of the Board of Trustees of the RoboCup Federation and was General Chair of the Third Robot World Cup Soccer Games and Conferences (RoboCup-99). She is also a Member of the Board of the European Coordinating Committee for Artificial Intelligence (ECCAI). Her main research interest is in establishing the connection (anchoring) between the symbols used to perform abstract reasoning and the physical entities which these symbols refer to. She also works in multi-agent systems and cooperative robotics.

Alessandro Saffiotti (Ph.D.) is Professor of Computer Science at the University of Örebro, Sweden, where he heads the AASS Mobile Robotics Lab since 1998. His research interests encompass autonomous robotics, soft computing, and non-standard logics for common-sense reasoning. He has published more than 70 papers in international journals and conferences, and co-edited the book *Fuzzy Logic Techniques for Autonomous Vehicle Navigation* (Springer, 2001). He is the leader of Team Sweden, the national Swedish team competing in RoboCup. He is a Member of IEEE, AAAI, and IAS.

ELSEVIER

Robotics and Autonomous Systems 43 (2003) 97–108

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Anchoring in a grounded layered architecture with integrated reasoning

Stuart C. Shapiro[a,*], Haythem O. Ismail[b]

[a] Department of Computer Science and Engineering, and Center for Cognitive Science, University at Buffalo, The State University of New York, 201 Bell Hall, Buffalo, NY 14260-2000, USA
[b] Department of Engineering Mathematics and Physics, Cairo University, Giza, Egypt

Abstract

The GLAIR grounded layered architecture with integrated reasoning for cognitive robots and intelligent autonomous agents has been used in a series of projects in which Cassie, the SNePS cognitive agent, has been incorporated into hardware- or software-simulated cognitive robots. In this paper, we present an informal, but coherent, overview of the GLAIR approach to anchoring the abstract symbolic terms that denote an agent's mental entities in the lower-level structures used by the embodied agent to operate in the real (or simulated) world. We discuss anchoring in the domains of: perceivable entities and properties, actions, time, and language.

Keywords: Anchoring; Symbol grounding; Autonomous agents; Cognitive robotics

1. Introduction

GLAIR (grounded layered architecture with integrated reasoning) is a three-level architecture for cognitive robots and intelligent autonomous agents [15,16]. GLAIR has been used in the design and implementation of Cassie, a cognitive robot [20–23,39,41, 44,46–48,50], which has been implemented as a hardware robot and in various software-simulated versions. The capabilities of the embodied Cassie have included: input and output in fragments of English, reasoning, performance of primitive and composite acts, motion, and vision.

Previous papers have described various aspects of GLAIR and Cassie. In this paper, we present, for the first time, a coherent, unified, overview of the GLAIR approach to anchoring the abstract symbolic terms that denote an agent's mental entities in the lower-level structures used by the embodied agent to operate in the real (or simulated) world.

In Section 2 we give an overview of the three levels of the GLAIR architecture. In Section 3 we discuss a hardware implementation of Cassie. In Section 4, we discuss anchoring in the domains of: perceivable entities and properties, actions, time, and language. In Section 5, we discuss some related work, and in Section 6, we summarize the paper. This paper has deliberately been kept as an informal, but coherent, overview of our approach. For more details, and more formal presentations, of particular aspects of our approach, see the papers cited herein.

2. GLAIR

GLAIR (grounded layered architecture with integrated reasoning) consists of three levels: the

[*] Corresponding author. Fax: +1-716-645-3464.
E-mail address: shapiro@cse.buffalo.edu (S.C. Shapiro).

0921-8890/03/$ – see front matter © 2003 Elsevier Science B.V. All rights reserved.
PII: S0921-8890(02)00352-4

knowledge level, the perceptuo-motor level, and the sensori-actuator level.

The knowledge level (KL) is the level at which conscious reasoning takes place. The KL is implemented by the SNePS knowledge representation and reasoning system [46,48,50], and its subsystem SNeRE (the SNePS rational engine) (see [28–31] and [53, Chapter 4]), which is used for scheduling and initiating the execution of intentional acts.

We refer to the KL as the "conscious" level, since that is the locus of symbols accessible to reasoning and to natural language interaction. It is the level containing the "abstract-level representations of objects" [5,6]. Similarly, the KL-level acts are "intentional" in the sense that they are scheduled as a result of natural language understanding and reasoning.

Atomic symbols in the KL are terms of the SNePS logic [42]. Symbol structures in the KL are functional terms in the same logic [40,42]. All terms denote mental entities [31,46]. For example, if Cassie is asked to "*Find a green thing*", she conceives of an entity whose only properties are being green and being a thing, by creating a KL term denoting that entity, and KL terms denoting propositions that the entity is green and that the entity is a thing, even though no such object, without further properties, exists in the world. When, in response to this request, Cassie find a particular green robot she recognizes (re-cognizes), by already having a KL term for it, she adds a KL term for the proposition that the two entities have the same extension. (Compare Frege's example that "The Morning Star is the Evening Star" [10].) This approach is in general accord with what Jackendoff calls "conceptualist semantics" [25,26]. We will consistently use "entity" for such a mental entity—the denotation of a KL term, and "object" for an object in the real (or simulated) world.

SNePS (and hence the KL) is implemented in Common Lisp.

The perceptuo-motor level (PML) is the level containing the "physical-level representations of objects" [5,6] consisting of object characteristics such as size, weight, texture, color, and shape. At this level objects are not characterized by KL terms such as categories (box, robot, person, etc.) or properties (green, tall, etc.). The PML also contains routines for well-practiced behaviors, including those that are primitive acts at the KL, and other subconscious activities that ground Cassie's consciousness of its body and surroundings.

The PML has been implemented in three sub-levels:

(1) The highest sub-level (which we will refer to as PMLa) has been implemented in Common Lisp, and contains the definitions of the functions that implement the activity represented by KL primitive acts.
(2) The middle sub-level (henceforth PMLw) contains a set of Common Lisp symbols and functions defined in the `World` package which use Common Lisp's foreign function facility to link to the lowest sub-level.
(3) The lowest sub-level (henceforth PMLc) has been a C implementation of "behavioral networks" [17,18].

The sensori-actuator level (SAL) is the level controlling the operation of sensors and actuators (being either hardware or simulated). The SAL has been implemented in C and other languages, depending on the implementation of the hardware or software-simulated robot.

The Common Lisp programs, PMLc, and the SAL run on different processes, and, in some circumstances, on different machines.

The topic of this paper is our approach to anchoring the KL terms that denote Cassie's (or any GLAIR-based agent's) mental entities in the PML structures used by embodied Cassie to operate in the real world. Briefly, our theoretical stance is that a KL term (symbol) serves as a *pivot*, supporting and coordinating various modalities. Anchoring is achieved by associating (we use the term "aligning") a KL term with one or more PML structures—more than one, if different PML structures are used by different modalities. Some PML structures are accessible to sensors, some to effectors. Others are accessible to natural language interaction. KL terms, but not PML structures, are accessible to reasoning. Cassie's ability to understand a natural language description, and then visually locate an object in the world satisfying that description depends on going from PML structures supporting natural language perception to KL symbol structures, possibly clarified and enhanced by reasoning, to PML structures supporting visual perception. Her ability to describe in natural language an object she is seeing in the world depends on the following that same path in the other direction.

3. The FEVAHR

Cassie in the role of a FEVAHR (foveal extravehicular activity helper-retriever) [2,14,41] was implemented, as a joint project of researchers at the University at Buffalo and researchers at Amherst Systems, Inc., on a Nomad 200 mobile robot, including sonar, bumpers, and wheels, enhanced with an hierarchical foveal vision system [1] consisting of a pair of cameras with associated hardware and software [7]. Henceforth, we will refer to Cassie in the role of a FEVAHR as Cassie$_F$ (in [14], Cassie$_F$ is referred to as Freddy).

Cassie$_F$ operates in a 17 × 17 ft. room containing: Cassie$_F$; Stu, a human supervisor; Bill, another human; a green robot; three indistinguishable red robots. In the actual room in which the Nomad robot operated, "Stu" was a yellow cube, "Bill" was a blue cube, the green robot was a green ball, and the red robots were red balls. Cassie$_F$ is always talking to either Stu or Bill. That person addresses Cassie$_F$ when he talks, and Cassie$_F$ always addresses that person when she talks. Cassie$_F$ can be told to talk to the other person, to find, look at, go to, or follow any of the people or other robots in the room, to wander, or to stop. Cassie$_F$ can also engage in conversations on a limited number of other topics in a fragment of English, similar to some of the conversations in [39]. While Cassie$_F$ is moving, she avoids obstacles.

Cassie$_F$'s SAL was designed and implemented by the researchers at Amherst Systems, Inc. Its hierarchical foveal vision system [1,2,7] was implemented and trained to recognize the several colors and shapes of the objects in the room.

Cassie$_F$'s KL and PML were designed and implemented by the researchers at the University at Buffalo, including the senior author of this paper. During development of the KL, and subsequently, we used several simulations of the robot and of the world it operates in:

The Nomad simulator uses the commercial simulator that was included with the Nomad robot, enhanced by a simulation of Cassie$_F$'s world and its vision system.

The VRML simulation simulates Cassie$_F$ and her world by VRML (virtual reality modeling language [3]) objects visible through a world-wide web browser.

The Garnet simulation simulates Cassie$_F$ and her world by Garnet [11] objects in a Garnet window.

The ASCII simulation, used to create examples for Section 4, implements the PMLw, PMLc, and SAL as sets of Common Lisp functions which print indications of what Cassie$_F$ would do.

No code at the KL or PMLa levels need be changed when switching among the hardware robot and these four different simulations. All that is required is a different PMLw file of functions that just print messages, or make calls to the appropriate PMLc sub-level.

4. Anchoring in GLAIR

4.1. Perceivable entities

There are KL terms for every mental entity Cassie has conceived of, including individual entities, categories of entities, colors, shapes, and other properties of entities.

There are PML structures (at the PMLw and PMLc sub-levels) for features of the perceivable world that Cassie's perceptual apparatus can detect and distinguish. For example, in the hardware and Nomad simulator versions of Cassie$_F$, each distinguishable color and each distinguishable shape is represented by a single integer, while in the VRML simulation, each is represented by a string, and in the Garnet and ASCII simulations, each is represented by a Lisp symbol. Each particular perceived object is represented at this level by an n-tuple of such structures, $\langle v_1, \ldots, v_n \rangle$, where each component, v_i, is a possible value of some perceptual feature domain, D_i. What domains are used and what values exist in each domain depend on the perceptual apparatus of the robot. We will call the n-tuples of feature values "PML-descriptions".

Our approach to grounding KL terms for perceivable entities, categories, and properties is to align a KL term with a PML-description, possibly with unfilled (null) components. For example, Cassie$_F$ used two-component PML-descriptions in which the domains were color and shape. In the hardware and Nomad simulator versions, the KL term denoting Cassie$_F$'s idea of blue was aligned with a PML-description whose color component was the PML structure the vision system used when it detected blue in the visual field, but whose shape component was null. The KL term denoting people was aligned with a PML-description whose shape component was the PML structure the vision system used when it detected a cube in the visual field, but whose color component was null. We have implemented alignment in various ways, including association lists, hash tables, and property lists.

Call a PML-description with some null components an "incomplete PML-description", and one with no null components a "complete PML-description". KL terms denoting perceivable properties and KL terms denoting recognizable categories of entities are aligned with incomplete PML-descriptions. Examples include the terms for blue and for people mentioned above, and may also include terms for the properties tall, fat, and bearded, and the categories man and woman. The words for these terms may be combined into verbal descriptions, such as "a tall, fat, bearded man", whose incomplete PML-descriptions may be used to perceptually recognize the object corresponding to the entity so described.

In this paper, we will use "description" (unqualified by "PML") only to mean a verbal description that can be used for perceptual recognition, such as "a tall, fat, bearded man", and not to mean a verbal description that cannot be used for perceptual recognition, such as "a college-educated businessman who lives in Amherst, NY". Cassie might have a KL term for an entity about which she knows no descriptive terms. For example, all she might believe about Fred is that he is a college-educated businessman who lives in Amherst, NY. Thus, she would be incapable of describing Fred (the way we are using "describe"). Nevertheless, it might be the case that Cassie's term denoting Fred is aligned with a complete PML-description. In this case, Cassie would be able to recognize Fred, though not describe him verbally. We call such a PML-description aligned with an entity-denoting term, the entity's PML-description.

A complete PML-description may be assembled for an entity by unifying the incomplete PML-descriptions of its known (conceived of) properties and categories. For example, if Cassie knows nothing about Harry, and we tell her that Harry is a tall, fat, bearded man, she would be able to assemble a PML-description of Harry and recognize him on the street (assuming that Cassie's terms for tall, fat, bearded, and man are aligned with incomplete PML-descriptions). In some cases, this might result in a set of several complete PML-descriptions. For example, the PML-descriptions of some, but not a particular, red chair might include PML-descriptions with different shape components. Once a PML-description is assembled for an entity, it can be cached by aligning the term denoting the entity directly with it. Afterwards, Cassie could recognize the entity without thinking about its description.

To find (come to be looking at) an entity, Cassie finds a PML-description of the entity that is as complete as possible, and directs her perceptual apparatus (via the SAL) to do what is necessary to cause an object satisfying it to be in her visual field. For example, in the Nomad version of Cassie$_F$, the PML-description of Bill is the 2-tuple $\langle 13, 21\rangle$, which is passed to the appropriate SAL routines, which move the cameras until a blue cube is in their field-of-view (see the section on actions, for a description of how actions are grounded).

If Cassie is looking at some object, she can recognize it if its PML-description is the PML-description of some entity she has already conceived of. If there is no such entity, Cassie can create a new KL term to denote this new entity, align it with the PML-description, and believe of it that it has those properties and is a member of those categories whose incomplete PML-descriptions unify with the PML-description of the new entity.

If there are multiple entities whose PML-descriptions match the object's PML-description, disambiguation is needed, or Cassie might simply not know which one of the entities she is looking at.

We are currently investigating the issue of when Cassie might decide that the object she is looking at is new, even though it looks exactly like another she has already conceived of (see [36]).

We have not worked on the problem of recognizing an entity by context. For example, a store clerk might be recognized as any person standing behind a cash register.[1] We speculate that this problem requires

[1] This example was suggested by one of the anonymous reviewers of Shapiro and Ismail [43].

Table 1
Objects and descriptions of Cassie$_F$'s world

Object	Color	Shape
`World:Bill`	`World:blue`	`World:square`
`World:Stu`	`World:yellow`	`World:square`
`World:Cassie`	`World:cyan`	`World:circle`
`World:Greenie`	`World:green`	`World:circle`
`World:Redrob-1`	`World:red`	`World:circle`
`World:Redrob-2`	`World:red`	`World:circle`
`World:Redrob-3`	`World:red`	`World:circle`

Table 2
Some of Cassie$_F$'s KL terms and their PML-descriptions

KL term	⟨Color, Shape⟩
`b1`	⟨`World:cyan`, `World:circle`⟩
`b5`	⟨`World:yellow`, `World:square`⟩
`b6`	⟨`World:blue`, `World:square`⟩
`m21`	⟨`World:green`, `nil`⟩
`m25`	⟨`World:red`, `nil`⟩
`m19`	⟨`nil`, `World:square`⟩
`m22`	⟨`nil`, `World:circle`⟩

a combination of KL knowledge and KL–PML alignment. Knowing that a person standing behind a cash register is a clerk is KL knowledge. Recognizing a person, a cash register, and the "behind" relation requires KL–PML alignment.

Consider an example interaction with the ASCII version of Cassie$_F$. In this simulation, created so that interactions can be shown in print, the entire PML and the simulated world are implemented in Common Lisp. The PML-descriptions have two domains, called "color" and "shape". There are seven objects in the simulated world. The Common Lisp symbols that represent these objects and their PML-descriptions are shown in Table 1.[2] Recall that Lisp symbols of the PMLw are in the `World` package, so Lisp prints them preceded by "`World:`".

The KL terms that are aligned with PML-descriptions are shown in Table 2. Notice that `b1`, `b5`, and `b6` are aligned with complete PML-descriptions, while `m21`, `m25`, `m19`, and `m22` are aligned with incomplete PML-descriptions. `b1`, `b5`, and `b6` denote individuals. `m21` and `m25` denote the properties green and red, respectively. `m19` and `m22` denote the categories of people and robots, respectively.

Table 3
Some of Cassie$_F$'s beliefs

b1's name is Cassie
b5's name is Stu
b6's name is Bill
Cassie is a FEVAHR
FEVAHRs are robots
Bill and Stu are people
Robbie is a green robot
b8, b9, and b10 are red robots
People and robots are things

Cassie$_F$'s relevant beliefs about the entities denoted by these terms may be glossed as shown in Table 3. The only descriptive terms Cassie$_F$ has for Bill and Stu are that they are people, and the only descriptive term she has for herself is that she is a robot. Nevertheless, Bill, Stu, and Cassie are aligned with complete PML-descriptions, so she can recognize them. On the other hand, neither Robbie, `b8`, `b9`, nor `b10` are aligned with PML-descriptions, although PML-descriptions can be assembled for them from their properties and categories.

Following is an interaction with Cassie$_F$ about these entities. Sentences preceded by "`:`" are human inputs. Sentences preceded by "`PML:`" and "`SAL:`" are reports of behaviors and simulated actions and perceptions by the ASCII version of Cassie$_F$ at the respective levels, and are not output by the other four versions. Notice that the PML deals with PML-descriptions, and only the SAL deals with (simulated) objects in the world. Sentences beginning with "`I`" are generated by Cassie$_F$. At the beginning of the interaction, Cassie$_F$ is looking at, listening to, and talking to Stu. (See next page).

4.2. Deictic registers

An important aspect of being embodied is being situated in the world and having direct access to components of that situatedness. This is modeled in GLAIR via a set of PML registers (variables), each of which can hold one or more KL terms or PML structures. Some of these registers derive from the theory of the Deictic Center [8], and include: `I`, the register that holds the KL term denoting the agent itself; `YOU`, the register that holds the KL term denoting the individual the agent is talking with; and `NOW`, the register that holds the KL term denoting the current time.

[2] The examples in this paper were created using SNePS 2.6 [50] running under Franz, Inc.'s Allegro CL 6.2 [9].

```
 : Find a robot.
PML: The FEVAHR is looking at (World:yellow World:square)
PML: The FEVAHR is looking for something that's (nil World:circle)
SAL: The FEVAHR found World:RedRob-1
PML: The FEVAHR found (World:red World:circle)
 I found a red robot.
PML: The FEVAHR is looking at (World:red World:circle)
 I am looking at a red robot.

 : Find a person.
PML: The FEVAHR is looking at (World:red World:circle)
PML: The FEVAHR is looking for something that's (nil World:square)
SAL: The FEVAHR found World:Stu
PML: The FEVAHR found (World:yellow World:square)
 I found you, Stu.
PML: The FEVAHR is looking at (World:yellow World:square)
 I am looking at you.

 : Find a green thing.
PML: The FEVAHR is looking at (World:yellow World:square)
PML: The FEVAHR is looking for something that's (World:green nil)
SAL: The FEVAHR found World:Greenie
PML: The FEVAHR found (World:green World:circle)
 I found Robbie.
PML: The FEVAHR is looking at (World:green World:circle)
 I am looking at Robbie.

 : Find Bill.
PML: The FEVAHR is looking at (World:green World:circle)
PML: The FEVAHR is looking for something that's (World:blue World:square)
SAL: The FEVAHR found World:Bill
PML: The FEVAHR found (World:blue World:square)
 I found Bill.
PML: The FEVAHR is looking at (World:blue World:square)
 I am looking at Bill.
```

It was by use of these registers that, in the example interaction shown in Section 4.1, Cassie used “`I`” to refer to the individual denoted by `b1` (herself), “you” to refer to the individual denoted by `b5` (Stu), and the appropriate tense in all the sentences she generated. The use of `NOW` is discussed further in Section 4.5, and language is discussed further in Section 4.6.

Embodiment is further modeled in GLAIR via a set of modality registers.

4.3. Modality registers

How does an agent know what it is doing? A standard technique in the Artificial Intelligence literature amounts to the following steps:

(1) I started doing *a* at some previous time or in some previous situation.
(2) I have not done anything since then to stop me from doing *a*.
(3) Therefore, I am still doing *a*.

However, we human's do not have to follow these steps to know what we are doing, because we have direct access to our bodies.

GLAIR agents know what they are doing via direct access to a set of PML registers termed "modality registers". For example, if one of Cassie's modalities were speech, and she were currently talking to Stu, her `SPEECH` register would contain the KL term denoting the state of Cassie's talking to Stu (and the term denoting Stu would be in the `YOU` register). In many cases, a single modality of an agent can be occupied by only one activity at a time. In that case the register for that modality would be constrained to contain only one term at a time.

One of the modality registers we have used is for keeping track of what Cassie is looking at. When she recognizes an object in her visual field, the KL term denoting the state of looking at the recognized entity is placed in the register, and is removed when the object is no longer in the visual field. If one assumed that Cassie could be looking at several objects at once, this register would be allowed to contain several terms. If asked to look at or find something that is already in her visual field, Cassie recognizes that fact, and doesn't need to do anything. The following interaction with Cassie$_{\mathrm{F}}$ continues from the previous one:

```
 : Look at Robbie.
PML: The FEVAHR is looking at (World:blue World:square)
PML: The FEVAHR is looking for something that's (World:green World:circle)
SAL: The FEVAHR found World:Greenie
PML: The FEVAHR found (World:green World:circle)
 I found Robbie.
PML: The FEVAHR is looking at (World:green World:circle)
 I am looking at Robbie.

 : Find a robot.
PML: The FEVAHR is looking at (World:green World:circle)
 I am looking at Robbie.
```

Comparing Cassie's response to the second request with her response to the previous requests, one can see that she realized that she was already looking at a robot, and so did not need to do anything to find one.

4.4. Actions

Some KL terms denote primitive actions that the GLAIR agent can perform. We call a structure consisting of an action and the entity or entities it is performed on, an "act". For example, the act of going to Bill consists of the action of going and the object Bill. Acts are denoted by KL functional terms.

Each KL action term that denotes a primitive action is aligned with a procedure in the PML. The procedure takes as arguments the KL terms for the arguments of the act to be performed. For example, when Cassie is asked to perform the act of going to Bill, the PML going-procedure is called on the KL Bill-term. It then finds the PML-description of Bill, and (via the SAL) causes the robot hardware to go to an object in the world that satisfies that description (or causes the robot simulation to simulate that behavior). The PML going-procedure also inserts the KL term denoting the state of Cassie's going to Bill into the relevant modality register(s), which when `NOW` moves (see Section 4.5), causes an appropriate proposition to be inserted into Cassie's belief space.

Acts whose actions are primitive are considered to be primitive acts. Composite acts are composed of primitive "control actions" and their arguments, which, themselves are primitive or composite acts. Control actions include sequence, selection, iteration, and non-deterministic choice [21,27–30,50]. There are also propositions for act preconditions, goals, effects, and for plans (what some call recipes) for carrying out non-primitive acts.

In the interactions shown above, sentences preceded by "SAL:" were printed by the simulated action function, which was called by the PML procedure aligned with the KL term for finding something. When Cassie was asked to look at Robbie, she did so by finding Robbie, because there is a KL belief that the plan for carrying out the non-primitive act of looking at something is to find that thing.

4.5. *Time*

As mentioned above, the NOW register always contains the KL term denoting the current time [20,23,24,41]. Actually, since "now" is vague (it could mean this minute, this day, this year, this century, etc.), NOW is considered to include the entire semi-lattice of times that include the smallest current now-interval Cassie has conceived of, as well as all other times containing that interval.

NOW moves whenever Cassie becomes aware of a new state. Some of the circumstances that cause her to become aware of a new state are: she acts, she observes a state holding, she is informed of a state that holds. NOW moves by Cassie's conceiving of a new smallest current now-interval (a new KL term is introduced with that denotation), and NOW is changed to contain that time. The other times in the old NOW are defeasibly extended into the new one by adding propositions asserting that the new NOW is a subinterval of them.

Whenever Cassie acts, the modality registers change (see above), and NOW moves. The times of the state(s) newly added to the modality registers are included in the new NOW semi-lattice, and the times of the state(s) deleted from the modality registers are placed into the past by adding propositions that assert that they precede the new NOW.

The following interaction, following the ones shown above, shows an action of Cassie's first being in the present, and then being in the past:

```
 : Who have you talked to?
 I am talking to you.

 : Talk to Bill.
PML: The FEVAHR is starting to talk
 to b6
 I am talking to you, Bill.
 : Who have you talked to?
 I talked to Stu
and I am talking to you.
```

The term denoting the state of Cassie's talking to Stu did not change between the first of these interactions and the third. What did change were: the state of Cassie's talking to Stu was replaced in the SPEECH register by the state of Cassie's talking to Bill; a propositional term was added to the KL that the time of talking to Stu was before the time of talking to Bill; and the NOW register was changed to include the time of talking to Bill and the times that include it.

To give GLAIR agents a "feel" for the amount of time that has passed, the PML has a COUNT register acting as an internal pacemaker [20,24]. The value of COUNT is a non-negative integer, incremented at regular intervals. Whenever NOW moves, the following happens:

(1) The old now-interval t_0 is aligned with the current value of COUNT, grounding it in a PML-measure of its duration.
(2) The value of COUNT is quantized into a value δ which is the nearest half-order of magnitude [19] to COUNT, providing an equivalence class of PML-measures that are not noticeably different.
(3) A KL term d, aligned with δ, is found or created, providing a mental entity denoting each class of durations.
(4) A belief is introduced into the KL that the duration of t_0 is d, so that the agent can have beliefs that two different states occurred for about the same length of time.
(5) COUNT is reset to 0, to prepare for measuring the new now-interval.

4.6. *Language*

Cassie interacts with humans in a fragment of English. Although it is possible to represent the linguistic knowledge of GLAIR agents in the KL, use reasoning to analyze input utterances [32–34,45], and use the acting system to generate utterances [12,13], we do not currently do this. Instead, the parsing and generation grammars, as well as the lexicon, are at the PML (see, e.g. [35,38,49]). There are KL terms for lexemes, and these are aligned with lexemes in the PML lexicon. We most frequently use a KL unary functional term to denote the concept expressed by a given lexeme, but this does not allow for polysemy, so we have occasionally used binary propositions that assert that some concept may be expressed by some lexeme. There may also be KL terms for inflected words, strings of words, and sentences. This allows one to discuss sentences and other language constructs with GLAIR agents.

This facility was used for Cassie to understand the human inputs shown in the example interactions in this paper, and for her to generate her responses (the sentences beginning with “`I`”). We can also use the low level `surface` function to see the NL expression Cassie would use to express the denotation of various SNePS terms (the prompt for this Lispish interaction level is “*”):

```
* (surface b1)
 me
* (surface b5)
 Stu
* (surface b6)
 you
* (surface m21)
 green
* (surface m115)
 I found a red robot.
* (surface m332)
 I am looking at Robbie.
```

(Remember, Cassie is currently looking at Robbie and talking to Bill.)

5. Related work

Coradeschi and Saffiotti [4,6] present a model of anchoring in an agent with a symbol system, which includes object symbols and unary predicate symbols, and a perceptual system, which includes attributes and percepts. Their grounding relation relates predicate symbols, attributes, and attribute values. Their anchor is a partial function from time to quadruples of: object symbols; percepts; partial functions from attributes to attribute values; and sets of predicate symbols. Their anchor is “reified in an internal data structure” [7, p. 408]. Their symbol system corresponds to our KL, and their perceptual system to a combination of our PML and SAL. While their anchor is a data structure that cuts across their symbol and perceptual systems, our KL and PML communicate by passing PML-descriptions from one to the other, sometimes by socket connections between different computers. Their discussion of “perceptual anchoring of symbols for action” [6] concerns the anchoring of object symbols of objects the actions are performed on. We also discussed the anchoring of action symbols to the PML procedures that carry them out.

Santos and Shanahan [37] discuss anchoring as the “process of assigning abstract symbols to real sensor data” and develop a theory whose “universe of discourse includes sorts for time points, depth, size, peaks, physical bodies and viewpoints. *Time points*, *depth* and *size* are variables that range over positive real numbers (R^+), *peaks* are variables for depth peaks, *physical bodies* are variables for objects of the world, *viewpoints* are points in R^3” [pp. 39–40, italics in the original]. We consider data such as these to belong at the PML, as not being the sort of entities people reason and talk about, and therefore, not the sort of entities cognitive robots should have at the KL. We view anchoring as the aligning of physical-level representations such as these to the KL terms used for reasoning.

Jackendoff [26] explicates a theory in which “the character of a consciously experienced entity is functionally determined by a cognitive structure that contains the following feature types: an indexical feature to which descriptive features can be attached; one or more modalities in which descriptive features are present; the actual descriptive features in the available modalities” [27, p. 313]. His indexical features correspond with our KL term, and his descriptive features correspond with our PML-descriptions. His suggestion that “we think of the descriptive features as being linked to a common indexical feature” [27, pp. 311–312] parallels our suggestion in Section 2 of KL terms as pivots.

6. Summary

We have given an informal, but coherent, unified, overview of our approach to connecting the abstract-level representations to the physical-level representations in GLAIR, an architecture for cognitive robots and intelligent autonomous agents. The abstract-level representations are terms of SNePS logic contained in the knowledge level (KL) of the GLAIR architecture, while the physical-level representations are n-tuples of perceptual features, procedures, and other symbol structures contained at the perceptuo-motor level (PML) of the architecture.

KL terms denoting perceivable entities, perceivable properties, and recognizable categories are aligned with PML-descriptions. Primitive actions are aligned with PML procedures. Deictic and modality registers hold KL terms for individuals and states that the agent is currently aware of, including states of its own body. They are updated by the PML procedures. The `NOW` register is used to give the agent a personal sense of time, including keeping track of current and past states. KL terms denoting times and temporal durations are aligned with PML numeric measures of durations created by the PML pacemaker. Lexemes are represented by KL terms that are aligned with PML lexicon entries used by the parsing and generation grammars, which, like PML procedures, mediate between the agent and the outside world, in this case, humans with which she communicates.

Acknowledgements

Development of GLAIR and the FEVAHR were supported in part by NASA under contracts NAS 9-19004 and NAS 9-19335. SNePS, GLAIR, and Cassie are products of SNeRG, the SNePS Research Group, Department of Computer Science and Engineering, University at Buffalo. Many group members, past and present, have contributed to these efforts. This paper is a revised version of Shapiro and Ismail [43]. We appreciate the valuable comments on previous drafts made by William J. Rapaport and other members of SNeRG, and by Venugopal Govindaraju, Peter D. Scott, and the anonymous reviewers. We also appreciate the efforts and discussions of Silvia Coradeschi, Alessandro Saffiotti, and other organizers and attendees of the 2001 AAAI Fall Symposium on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems.

References

[1] C. Bandera, P. Scott, Foveal machine vision systems, in: Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, November 1989, pp. 596–599.

[2] C. Bandera, S. Shapiro, H. Hexmoor, Foveal machine vision for robots using agent based gaze control, Final Technical Report No. 613-9160001, Amherst Systems, Inc., Buffalo, NY, September 1994.

[3] R. Carey, G. Bell, The Annotated VRML 2.0 Reference Manual, Addison-Wesley Developers Press, Reading, MA, 1997.

[4] S. Coradeschi, A. Saffiotti, Anchoring symbols to sensor data: Preliminary report, in: Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-2000) and 12th Innovative Applications of Artificial Intelligence Conference (IAAI-2000), AAAI Press/The MIT Press, Menlo Park, CA, 2000, pp. 129–135.

[5] S. Coradeschi, A. Saffiotti, Forward, in: S. Coradeschi, A. Saffiotti (Eds.), Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems: Papers from the 2001 AAAI Fall Symposium, Technical Report No. FS-01-01, AAAI Press, Menlo Park, CA, 2001, p. viii.

[6] S. Coradeschi, A. Saffiotti, Perceptual anchoring of symbols for action, in: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01), Morgan Kaufmann, San Francisco, CA, 2001, pp. 407–412.

[7] F. Du, A. Izatt, C. Bandera, An MIMD computing platform for a hierarchical foveal machine vision system, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'96), IEEE Computer Society Press, Silver Spring, MD, June 1996, pp. 720–725.

[8] J.F. Duchan, G.A. Bruder, L.E. Hewitt (Eds.), Deixis in Narrative: A Cognitive Science Perspective, Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1995.

[9] Franz, Inc., Oakland, CA, Allegro CL 6.2 Documentation, 2002. http://www.franz.com/support/documentation/6.2/doc/introduction.htm.

[10] G. Frege, On sense and reference, in: P. Geach, M. Black (Eds.), Translations from the Philosophical Writings of Gottlob Frege, Blackwell Scientific Publications, Oxford, 1970, pp. 56–78 (original work published in 1892).

[11] Garnet Group, Garnet Reference Manual, Version 2.2, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1993.

[12] S. Haller, Planning text about plans interactively, International Journal of Expert Systems 9 (1) (1996) 85–112.

[13] S. Haller, An introduction to interactive discourse processing from the perspective of plan recognition and text planning, Artificial Intelligence Review 13 (4) (1999) 259–333.

[14] H. Hexmoor, C. Bandera, Architectural issues for integration of sensing and acting modalities, in: Proceedings of the IEEE International Symposium on Intelligent Control: ISIC/CIRA/ISIS Joint Conference, IEEE Press, New York, 1998, pp. 319–324.

[15] H. Hexmoor, J. Lammens, S.C. Shapiro, Embodiment in GLAIR: a grounded layered architecture with integrated reasoning for autonomous agents, in: D.D. Dankel II, J. Stewman (Eds.), Proceedings of the Sixth Florida AI Research Symposium (FLAIRS'93), The Florida AI Research Society, April 1993, pp. 325–329.

[16] H. Hexmoor, S.C. Shapiro, Integrating skill and knowledge in expert agents, in: P.J. Feltovich, K.M. Ford, R.R. Hoffman (Eds.), Expertise in Context, AAAI Press/MIT Press, Cambridge, MA, 1997, pp. 383–404.

[17] H.H. Hexmoor, Representing and learning routine activities, Ph.D. Dissertation, Technical Report No. 98-04, Department of Computer Science, State University of New York at Buffalo, Buffalo, NY, December 1995.

[18] H.H. Hexmoor, Learning routines, in: M.J. Wooldrige, J.P. Muller, M. Tambe (Eds.), Intelligent Agents. II. Agent Theories, Architectures and Languages, Lecture Notes in Artificial Intelligence, vol. 1037, Springer, Berlin, 1996, pp. 97–110.

[19] J.R. Hobbs, Half orders of magnitude, in: L. Obrst, I. Mani (Eds.), Papers from the Workshop on Semantic Approximation, Granularity, and Vagueness, Proceedings of the Workshop of the Seventh International Conference on Principles of Knowledge Representation and Reasoning, Breckenridge, CO, 2000, pp. 28–38.

[20] H.O. Ismail, Reasoning and acting in time, Ph.D. Dissertation, Technical Report No. 2001-11, University at Buffalo, The State University of New York, Buffalo, NY, August 2001.

[21] H.O. Ismail, S.C. Shapiro, Cascaded acts: Conscious sequential acting for embodied agents, Technical Report No. 99-10, Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY, November 1999.

[22] H.O. Ismail, S.C. Shapiro, Conscious error recovery and interrupt handling, in: H.R. Arabnia (Ed.), Proceedings of the International Conference on Artificial Intelligence (IC-AI'2000), CSREA Press, Las Vegas, NV, 2000, pp. 633–639.

[23] H.O. Ismail, S.C. Shapiro, Two problems with reasoning and acting in time, in: A.G. Cohn, F. Giunchiglia, B. Selman (Eds.), Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR'2000), Morgan Kaufmann, San Francisco, 2000, pp. 355–365.

[24] H.O. Ismail, S.C. Shapiro, The cognitive clock: A formal investigation of the epistemology of time, Technical Report No. 2001–08, Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY, August 2001.

[25] R. Jackendoff, Semantic Structures, MIT Press, Cambridge, MA, 1990.

[26] R. Jackendoff, Foundations of Language: Brain, Meaning, Grammar, Evolution, Oxford University Press, Oxford, UK, 2002.

[27] D. Kumar, From beliefs and goals to intentions and actions: an amalgamated model of inference and acting, Ph.D. Dissertation, Technical Report No. 94-04, State University of New York at Buffalo, Buffalo, NY, 1994.

[28] D. Kumar, The SNePS BDI architecture, Decision Support Systems 16 (1) (1996) 3–19.

[29] D. Kumar, S.C. Shapiro, Acting in service of inference (and vice versa), in: D.D. Dankel II (Ed.), Proceedings of the Seventh Florida AI Research Symposium (FLAIRS'94), The Florida AI Research Society, May 1994, pp. 207–211.

[30] D. Kumar, S.C. Shapiro, The OK BDI architecture, International Journal on Artificial Intelligence Tools 3 (3) (1994) 349–366.

[31] A.S. Maida, S.C. Shapiro, Intensional concepts in propositional semantic networks, Cognitive Science 6 (4) (1982) 291–330. Reprinted in: R.J. Brachman, H.J. Levesque (Eds.), Readings in Knowledge Representation, Morgan Kaufmann, San Mateo, CA, 1985, pp. 170–189.

[32] J.G. Neal, S.C. Shapiro, Parsing as a form of inference in a multiprocessing environment, in: Proceedings of the Conference on Intelligent Systems and Machines, Oakland University, Rochester, MI, 1985, pp. 19–24.

[33] J.G. Neal, S.C. Shapiro, Knowledge-based parsing, in: L. Bolc (Ed.), Natural Language Parsing Systems, Springer, Berlin, 1987, pp. 49–92.

[34] J.G. Neal, S.C. Shapiro, Knowledge representation for reasoning about language, in: J.C. Boudreaux, B.W. Hamill, R. Jernigan (Eds.), The Role of Language in Problem Solving, vol. 2, Elsevier, Amsterdam, 1987, pp. 27–46.

[35] W.J. Rapaport, S.C. Shapiro, J.M. Wiebe, Quasi-indexicals and knowledge reports, Cognitive Science 21 (1) (1997) 63–107. Reprinted in: F. Orilia, W.J. Rapaport (Eds.), Thought, Language, and Ontology: Essays in Memory of Hector–Neri Castañeda, Kluwer Academic Publishers, Dordrecht, 1998, pp. 235–294.

[36] J.F. Santore, S.C. Shapiro, Identifying perceptually indistinguishable objects: Is that the same one you saw before? in: C. Baral, S. McIlraith (Eds.), Cognitive Robotics (CogRob2002), Papers from the AAAI Workshop, Technical Report No. WS-02-05, AAAI Press, Menlo Park, CA, 2002, pp. 96–102.

[37] P.E. Santos, M.P. Shanahan, From stereoscopic vision to symbolic representation, in: S. Coradeschi, A. Saffiotti (Eds.), Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems: Papers from the 2001 AAAI Fall Symposium, Technical Report No. FS-01-01, AAAI Press, Menlo Park, CA, 2001, pp. 37–43.

[38] S.C. Shapiro, Generalized augmented transition network grammars for generation from semantic networks, The American Journal of Computational Linguistics 8 (1) (1982) 12–25.

[39] S.C. Shapiro, The CASSIE projects: An approach to natural language competence, in: J.P. Martins, E.M. Morgado (Eds.), Proceedings of the Fourth Portugese Conference on Artificial Intelligence (EPIA'89), Lecture Notes in Artificial Intelligence, vol. 390, Springer, Berlin, 1989, pp. 362–380.

[40] S.C. Shapiro, Belief spaces as sets of propositions, Journal of Experimental and Theoretical Artificial Intelligence (JETAI) 5 (2–3) (1993) 225–235.

[41] S.C. Shapiro, Embodied Cassie, in: Cognitive Robotics: Papers from the 1998 AAAI Fall Symposium, Technical Report No. FS-98-02, AAAI Press, Menlo Park, CA, October 1998, pp. 136–143.

[42] S.C. Shapiro, SNePS: A logic for natural language understanding and commonsense reasoning, in: Ł. Iwańska, S.C. Shapiro (Eds.), Natural Language Processing and Knowledge Representation: Language for Knowledge and

Knowledge for Language, AAAI Press/The MIT Press, Menlo Park, CA, 2000, pp. 175–195.

[43] S.C. Shapiro, H.O. Ismail, Symbol-anchoring in Cassie, in: S. Coradeschi, A. Saffiotti (Eds.), Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems: Papers from the 2001 AAAI Fall Symposium, Technical Report No. FS-01-01, AAAI Press, Menlo Park, CA, 2001, pp. 2–8.

[44] S.C. Shapiro, H.O. Ismail, J.F. Santore, Our dinner with Cassie, in: Working Notes for the AAAI 2000 Spring Symposium on Natural Dialogues with Practical Robotic Devices, AAAI Press, Menlo Park, CA, 2000, pp. 57–61.

[45] S.C. Shapiro, J.G. Neal, A knowledge engineering approach to natural language understanding, in: Proceedings of the 20th Annual Meeting of the Association for Computational Linguistics, ACL, Menlo Park, CA, 1982, pp. 136–144.

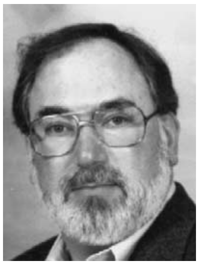
[46] S.C. Shapiro, W.J. Rapaport, SNePS considered as a fully intensional propositional semantic network, in: N. Cercone, G. McCalla (Eds.), The Knowledge Frontier, Springer, New York, 1987, pp. 263–315.

[47] S.C. Shapiro, W.J. Rapaport, Models and minds: Knowledge representation for natural-language competence, in: R. Cummins, J. Pollock (Eds.), Philosophy and AI: Essays at the Interface, MIT Press, Cambridge, MA, 1991, pp. 215–259.

[48] S.C. Shapiro, W.J. Rapaport, The SNePS family, Computers and Mathematics with Applications 23 (2–5) (1992) 243–275. Reprinted in: F. Lehmann (Ed.), Semantic Networks in Artificial Intelligence, Pergamon Press, Oxford, 1992, pp. 243–275.

[49] S.C. Shapiro, W.J. Rapaport, An introduction to a computational reader of narratives, in: J.F. Duchan, G.A. Bruder, L.E. Hewitt (Eds.), Deixis in Narrative: A Cognitive Science Perspective, Lawrence Erlbaum Associates, Inc., Hillsdale, NJ, 1995, pp. 79–105.

[50] S.C. Shapiro, The SNePS Implementation Group, SNePS 2.6 User's Manual, Department of Computer Science and Engineering, University at Buffalo, The State University of New York, Buffalo, NY, 2002.



Stuart C. Shapiro received his S.B. degree in mathematics from the Massachusetts Institute of Technology in 1966, and the M.S. and Ph.D. degrees in computer sciences from the University of Wisconsin–Madison in 1968 and 1971. He is currently Professor of Computer Science and Engineering at the University at Buffalo, The State University of New York. where he was Department Chair from 1984 to 1990, and from 1996 to 1999.

Prof. Shapiro is editor-in-chief of *The Encyclopedia of Artificial Intelligence* (Wiley, First Edition, 1987; Second Edition, 1992), co-editor of *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language* (AAAI Press/The MIT Press, 2000), author of *Techniques of Artificial Intelligence* (Van Nostrand, 1979), *LISP: An Interactive Approach* (Computer Science Press, 1986), *Common Lisp: An Interactive Approach* (Computer Science Press, 1992), and author or co-author of over 190 technical articles and reports. He has served on the editorial board of the *American Journal of Computational Linguistics*, and as Guest Editor of special issues of *Minds and Machines*, and of the *International Journal of Expert Systems*. Prof. Shapiro is a member of the ACM, the ACL, the Cognitive Science Society, and Sigma Xi, a senior member of the IEEE, and a fellow of the American Association for Artificial Intelligence. He has served as Chair of ACM/SIGART and as President of Knowledge Representation and Reasoning, Inc.

Haythem O. Ismail received his B.Sc. degree in communications and electronics in 1992, Diploma in mathematics in 1993, and M.Sc. in computer science (Natural Language Processing) in 1996, all from Cairo University, and his Ph.D. in computer science and engineering from the University at Buffalo, The State University of New York, in 2001. He is currently Assistant Professor of Engineering Mathematics and Physics at Cairo University.

ELSEVIER

Robotics and Autonomous Systems 43 (2003) 109–120

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Anchoring of semiotic symbols

Paul Vogt

Institute for Knowledge and Agent Technology, Universiteit Maastricht, P.O. Box 616, 6200 MD Maastricht, The Netherlands

Abstract

This paper presents arguments for approaching the anchoring problem using *semiotic symbols*. Semiotic symbols are defined by a triadic relation between forms, meanings and referents, thus having an implicit relation to the real world. Anchors are formed between these three elements rather than between ‘traditional’ symbols and sensory images. This allows an optimization between the form (i.e. the ‘traditional’ symbol) and the referent. A robotic experiment based on adaptive language games illustrates how the anchoring of semiotic symbols can be achieved in a bottom-up fashion. The paper concludes that applying semiotic symbols is a potentially valuable approach toward anchoring.

Keywords: Anchoring problem; Symbol grounding problem; Physical grounding; Adaptive language games; Semiotics

1. Introduction

The symbol grounding problem that deals with the question how symbols can be used meaningfully [8] is one of the hardest problems in AI and robotics. As many robotic applications use symbols for reasoning, problem solving and communication, solutions for this problem are extremely important for robotics research and development. But symbol grounding is also an important problem in studying foundations of cognition such as the evolution of language, as human language is primarily symbolic [7].

Recently, a formalized solution for the technical aspect of the symbol grounding problem has been proposed under the name of *anchoring* [5]. Anchoring concentrates on constructing and maintaining a relation between a symbol and a sensory image that is acquired from observing a physical object. Symbol grounding is, in addition to anchoring, also concerned with ‘anchoring’ abstractions and, more fundamentally, with philosophical issues relating to the meaning of symbols.

Many attempts to tackle the anchoring problem start with the design of predefined symbol systems that have predefined anchors to relate symbols with visual percepts [5,13]. Recently, an increasing number of attempts have been made to approach the anchoring problem from the bottom-up in which robots develop their symbolic representations during their evolution—be it phylogenetic and/or ontogenetic. These attempts often relate to the development of symbolic communication [2,12,16,22,24].

The common approach to tackle the anchoring problem focuses on the development—hand-coded or learned—of anchors between symbols and sensory images [5]. This is a difficult problem since the robots have to deal with the object constancy problem: when viewing an object from different locations, the sensory images relating to this object differ enormously because the size of the projection may differ or because the object may be obscured. Humans are well capable of dealing with object constancy, but it is unclear how this works. One approach to tackle

E-mail address: p.vogt@cs.unimaas.nl (P. Vogt).

0921-8890/03/$ – see front matter © 2003 Elsevier Science B.V. All rights reserved.
PII: S0921-8890(02)00353-6

the problem of object constancy would be to develop anchors between symbols and the real world object, rather than between symbols and sensory images.

This paper proposes that the anchoring problem can be solved in terms of *semiotic symbols*, which have implicit anchors in the real world [22]. An experiment based on Steels' [14] language game model illustrates how anchors in these semiotic symbols may be constructed from the bottom-up through the use of language. In addition, it is discussed how the presented language game model may explain the cognitive phenomenon of family resemblance [23].

The paper is organized as follows: Section 2 presents the notion of semiotic symbols and discusses some of the requirements for anchoring these. The experimental setup is presented in Section 3. Section 4 presents the experimental results. Discussions of the issues raised in the paper are presented in Section 5. Conclusions are given in Section 6.

2. The anchors of semiotic symbols

In this section, I will define the notion of semiotic symbols as opposed to the definition of symbols that is commonly used in AI. As I will argue below, semiotic symbols have implicit anchors between some internal structures and reality. Finally, I will discuss under what conditions semiotic symbols may emerge.

The definition of semiotic symbols is adopted from Peirce [11], who defined a semiotic symbol in terms of a sign, which in semiotics is a relation between a *referent*, *meaning* and *form*.[1] These three elements can be described as follows:

Form. A form (or *word*) is the shape of the sign, which is not necessarily material.

Meaning. The meaning is the sense that is made of the sign.

Referent. A referent is the object that stands for the sign, which may include abstractions, actions or other signs.

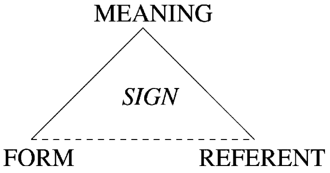


Fig. 1. The semiotic triangle illustrates the relations between referent, form and meaning that constitute a sign. Each line is an anchor, but the dotted line indicates that the relation between a form and a referent need not be a physical anchor, which must be established between referent and meaning, and between meaning and form.

The relation between the referent, form and meaning is often illustrated with the semiotic triangle [10] as shown in Fig. 1. According to Peirce, a sign becomes a (*semiotic*) *symbol* when its form, in relation to its meaning is arbitrary or conventionalized so that the relationship has to be learned; otherwise the sign is either an *icon* or an *index*.

A semiotic symbol becomes meaningful when it is constructed and used functionally by an agent, which is conform Wittgenstein [23]. As such the meaning arises from the interaction of an agent that uses a form with the referent. Elsewhere, I have argued that the symbol grounding problem as presented by Harnad is no longer relevant when we adopt semiotic symbols, because these are *per definition* grounded as their meanings have intrinsic relations with their referents [22]. This, however, does not solve the symbol grounding problem, but translates it into another—more technical—problem, which I have coined the *physical symbol grounding problem*.[2]

The physical symbol grounding problem is related to the anchoring problem in that it aims at constructing and maintaining anchors between symbols—i.e., the *forms* in semiotic symbols—and reality. Coradeschi and Saffiotti's [5] description of anchoring, however, focuses on anchors between forms and sensory data. As the sensory data is acquired from a robot's interaction with its environment, the forms relate to the real world. The anchors, however, are not necessarily

[1] Peirce called this a symbol rather than a semiotic symbol. I call it a semiotic symbol to distinguish it from the—in AI and some other disciplines of cognitive science—commonly used definition of a symbol, which is similar to the form of the semiotic symbol. In addition, Peirce used the terms *representamen*, *interpretant* and *object* where I use the terms form, meaning and referent.

[2] This problem is coined the physical symbol grounding problem to indicate that semiotic symbols provide a way to approach symbol grounding with the physical grounding hypothesis [4] as the semiotic symbols themselves form a coupling between the environment and an agent's behavior and thus are physically grounded.

constructed to maintain a relation with the real world entity, but rather with the sensory image of this entity. The physical symbol grounding problem, on the other hand, does focus on constructing and maintaining a relation with the real world by constructing anchors between forms and real world entities, mediated by anchors between forms and meanings, and between meanings and referents. In addition, where anchoring relates forms to sensory images (and thus to the sensing of physical objects), the physical symbol grounding problem is not restricted to constructing semiotic symbols about physical objects, but also include abstractions, movements and even other semiotic symbols.

The development of semiotic symbols depends on how an agent interacts with its environment. When the semiotic symbols are used in language, the way the meaning is constructed depends on how it is used [23]. However, the meaning of semiotic symbols also must have a part that can be memorized, which can be represented in terms of prototypical categories. When mediating on the meaning of a semiotic symbol, agents must confer to a similar meaning. Hence they must try to find a common way to name the meaning. It is not unlikely that this requires for the agents to construct similar representations of the meanings they use. In addition, the construction of semiotic symbols should be adaptive, because it may be impossible to design 'static' anchors that apply to the dynamic interactions of a robot with its environment [9]. An adaptive approach to construct semiotic symbols allows robots to create new anchors when none exist or when existing ones are insufficient. As a result, I assume that a semiotic symbol can have multiple meanings (or prototypes) to stand for a referent in relation to a form. These different meanings of a semiotic symbol will then be used to interpret a referent on different occasions. To achieve such a development of semiotic symbols in communication, I assume that the meanings co-develop with linguistic forms [3] by means of cultural interactions between agents and their environment [18].

The anchors between meanings and referents arise from the physical interactions between an agent and its environment. The meanings are anchored to linguistic forms through the production and interpretation of expressions. These physical anchors between referents, meanings and forms provide an implicit non-physical anchor between the forms and referents through their use in language (Fig. 1). The way these anchors are formed is influenced by the agents' interactions with their environment and individual adaptations as a self-organizing process [14].

For robots that develop semiotic symbols from the bottom-up, the above requires that robots are capable of interacting with their environment, including each other. Furthermore, they have to construct and memorize categorizations that provide anchors between the referents and the categories such that these can be used appropriately in language. To use these in language they also have to construct anchors between the categories and linguistic forms adaptively. How this can be modeled is explained in the next section.

3. Adaptive language games

To illustrate how a set of anchored symbols can be developed from the bottom-up, an experiment is presented in which two mobile LEGO robots bootstrapped a symbolic communication system. To achieve this, the robots engaged in a series of *adaptive language games* [14,17] in which they tried to communicate the form that stands for an object and adapt their internal structures in order to improve their performance on later occasions. Various types of language games have been implemented such as *observational games*, *guessing games* and *selfish games*, which differ from each other in the type of learning mechanism the robots use and in what non-verbal input they use to determine the reference of an utterance [19,20]. For the experiment of this paper, the robots played a series of *guessing games*. Below follows a technical description of the experimental setup.

3.1. The environment

In the experiment two mobile LEGO robots were used that were equipped with light sensors, bumpers, active infrared, two motors, a radio module and a sensorimotor board, see Fig. 2. The light sensors were used to detect the objects in the robots' environment. The other sensors and the motors were used to process the physical behaviors of the robots.

The robots were situated in a small environment (2.5 m × 2.5 m) in which four light sources were placed

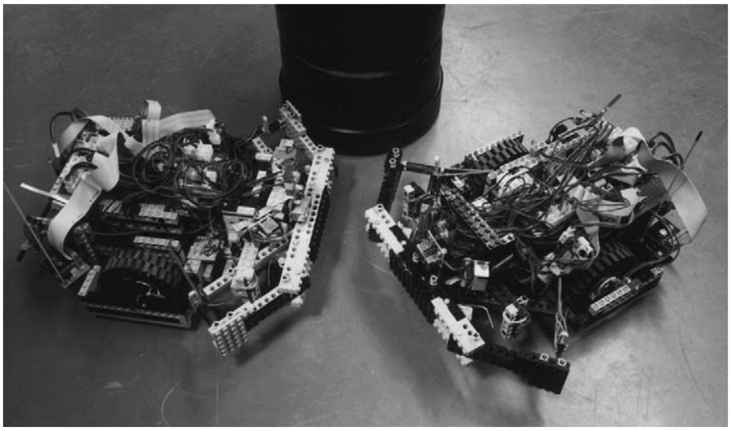


Fig. 2. The LEGO robots and a light source as used in the experiment.

at different heights. The light sources acted as the objects that the robots tried to name. The four light sensors of the robots were mounted at the same height as the different light sources. Each sensor outputs its readings on a *sensory channel*. A sensory channel is said to *correspond* with a particular light source if the sensor has the same height as this light source. The goal of the experiment was that the robots developed a lexicon with which they could successfully name the different light sources.

3.2. Sensing, segmentation and feature extraction

Through the interactions of the robots with their environment, they obtained raw sensory data. In order to reduce the redundant information from this high dimensional data, the robots transferred this data into low dimensional *feature vectors*. The process of acquiring feature vectors was done by *sensing*, *segmentation* and *feature extraction*. Each subsequent step reduced the amount of sensory data as if it were a sieve.

3.2.1. Sensing

A guessing game started when both robots were standing close to each other with their backs ‘facing’ each other.[3] During the sensing phase, the robots rotated one by one 720° to obtain a spatial view of their environment. A spatial view contained the raw sensory data from the middle 360°, which can be written in the form of a matrix [4]

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,q} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,q} \end{bmatrix}, \tag{1}$$

where each row represents the sensory data of the n sensory channels (four in the experiment) and the detection of q measurements are given in the columns.[5] The sensory data was sent to a stand alone PC where all further processing took place off-line.

Fig. 3 shows the sensing of the two robots during a guessing game. The left figure shows that robot A clearly detected the four light sources; there appears a ‘winning’ peak for every light sensor s_i that corresponds to one light source. The right figure shows that robot *B* did not sense all four light sources clearly and hence acquired a different view than robot *A*. This happened because both robots were not located at the same position.

[3] In the original implementation, the robots aligned themselves autonomously [17], but to speed up the experiments, the robots were placed by hand for this experiment.

[4] The robots rotated twice instead of once to ensure they rotated at a constant speed when the actual sensing started. This is done because the onset and offset of the movement induced a warped view, which in turn induced much noise for the segmentation.

[5] Note that although the robots have more than four sensors only the four light sensors are used to construct anchors.

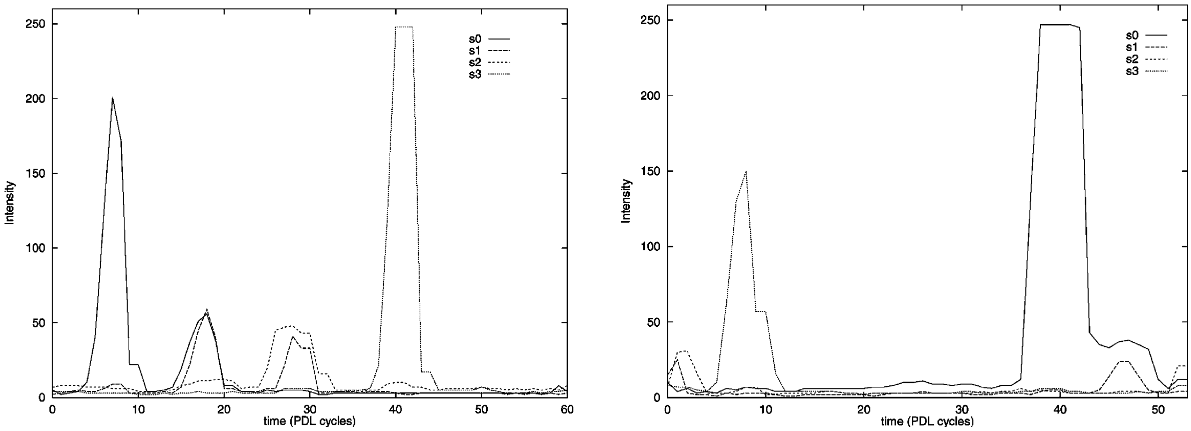


Fig. 3. The sensing of robot A (left) and robot B (right) during a language game. The plots show the spatial view of the robots' environment. It was acquired during 360° of their rotation. The y-axis shows the intensity of the sensors, while the x-axis determines the time (or angle) of the sensing in PDL units. A PDL unit takes about 1/40 s, hence the total time of these sensing events took about 1.5 s for robot A and 1.3 s for robot B.

3.2.2. Segmentation

The segmentation phase extracted connecting regions where the sensory data exceeded a threshold that represented the upper noise level of that sensor. These regions were supposed to be induced by the sensing of a light source. To accomplish this segmentation, the raw sensory input X was thresholded for noise resulting in $X' = \text{matrix}(x'_{i,j})$ according to

$$x'_{i,j} = H(x_{i,j} - \Theta_i), \tag{2}$$

where

$$H(x) = \begin{cases} x & \text{if } x \geq 0, \\ 0 & \text{if } x < 0 \end{cases} \tag{3}$$

and Θ_i represents the upper noise level of light sensor i, which was acquired empirically for each sensor.

Given the preprocessed sensory data X', a segment S_k can be defined as the *largest* matrix

$$S_k = \begin{bmatrix} x'_{1,r} & \dots & x'_{1,m} \\ \vdots & \ddots & \vdots \\ x'_{n,r} & \dots & x'_{n,m} \end{bmatrix}, \tag{4}$$

where in each column j there is at least one element for which $x'_{i,j} > 0$ for $i = 1, \dots, n$ and $j = r, \dots, m$; and where $1 \leq r < m < q$. Note that the inequality $r < m$ implies that the segments have to contain at least two measurements to filter out further noise. When a segment was detected at the start of the view and another was detected at the end, both segments were concatenated.

Ideally, the segmentation resulted in a set that contained a segment for each light source. This set constituted what is called the *context* of the guessing game, i.e. $\text{Cxt} = \{S_1, \dots, S_N\}$, where N is the number of segments that were sensed. Each robot participating in the guessing game acquired its own context which could differ from another.

3.2.3. Feature extraction

The feature extraction resulted in a feature vector $\mathbf{f} = (f_1 \dots, f_n)$, where $f_i = \varphi(S_k)$ was a function that normalized the maximum intensity of a sensory channel i to the overall maximum intensity within a segment S_k. That is, the maximum value in row i of the matrix S_k was normalized to the maximum value of the entire matrix. Mathematically, the function $\varphi(S_k)$ is given by

$$\varphi(S_k) = \frac{\max_{j \in [r,m]}(x'_{i,j})}{\max_{S_k}(x'_{p,q})}. \tag{5}$$

This way the function extracted the invariant property that the feature of the sensory channel with the overall highest intensity inside a segment had a value of 1, whereas all other features had a value ≤ 1. Or, in other words, the feature with value 1 corresponded to the light source the feature vector referred to. The space that spans all possible feature vectors $\mathbf{f}$ is called the n-dimensional feature space $\mathcal{F} = [0,1]^n$, or *feature space* for short.

3.3. Discrimination game

Each robot played a *discrimination game* [15] to form a memorized representation of the meaning—or meaning for short—for each (potential) *topic*. A topic is a segment from the acquired context as described by its feature vector. The speaker selected its topic randomly from the context and this topic became the subject of communication. As the hearer of a guessing game tried to guess what the speaker's utterance referred to, it had to consider all segments in its context as a *potential topic*. A discrimination game was successful when it resulted in one or more categories that distinguished the topic from all other segments in the context. When the robot failed to find such a category, the discrimination game failed and the robot expanded its ontology in which the categories were stored. The discrimination game is a sequence of three processes: *categorization*, *discrimination* and *adaptation*.

3.3.1. Categorization

A category $c = \langle \mathbf{c}, \nu, \rho, \kappa \rangle$ was defined as a region in the feature space $\mathcal{F}$ and it was represented by scores ν, ρ and κ and a prototype $\mathbf{c} = (y_1, \ldots, y_n)$, where y_i were the coordinates of the prototype in each of the n dimensions of $\mathcal{F}$. The category was the region in $\mathcal{F}$ in which the points had the nearest distance to $\mathbf{c}$. Each feature vector in the context was categorized using the *1-nearest neighbor algorithm* [6]. So, feature vector $\mathbf{f}$ was categorized with that category c for which the prototype $\mathbf{c}$ had the smallest Euclidean distance $\|\mathbf{f} - \mathbf{c}\|$.

In order to allow generalization and specialization of the categories, different versions of the feature space $\mathcal{F}_\lambda$ were available to a robot. In each space a different resolution was obtained by allowing each dimension of $\mathcal{F}_\lambda$ to be exploited up to 3^λ times, where $\lambda = 0, \ldots, \lambda_{\max}$. How this was done will be explained in Section 3.3.3.

The use of different feature spaces allowed the robots to categorize a segment in different ways. The categorization of segment S_k resulted in a set of categories $C_k = \{c_0, \ldots, c_m\}$, where $m \leq \lambda_{\max}$.

3.3.2. Discrimination

Suppose that a robot wants to find distinctive categories for (potential) topic S_t, then a distinctive category set, DC, can be defined as follows:

$$\mathrm{DC} = \{c_i \in C_t | \forall (S_k \in \mathrm{Cxt} \backslash \{S_t\}) : c_i \notin C_k\}. \tag{6}$$

Or in words, the distinctive category set DC consists of all categories c_i of the topic S_t that are not a category of any other segment S_k in the context Cxt.

3.3.3. Adaptation

If $\mathrm{DC} = \emptyset$, the discrimination game fails and the robot should adapt its ontology by constructing new categories. Suppose that the robot tried to categorize feature vector $\mathbf{f} = (f_1, \ldots, f_n)$, then new categories were created as follows:

(1) Select an arbitrary feature $f_i > 0$.
(2) Select a feature space $\mathcal{F}_\lambda$ that has not been exploited 3^λ times in dimension i for λ as low as possible. If no such space can be found, the adaptation is stopped.
(3) Create new prototypes $\mathbf{c_j} = (y_1, \ldots, y_n)$, where $y_i = f_i$ and the other y_r are made by combining the features from all existing prototypes in $\mathcal{F}_\lambda$.
(4) Add the new prototypical categories $c_j = \langle \mathbf{c}_j, \nu_j, \rho_j, \kappa_j \rangle$ to the feature space $\mathcal{F}_\lambda$, with $\nu = \rho = 0.01$ and $\kappa = 1 - (\lambda/\lambda_{\max})$.

The three scores ν, ρ and κ together constitute the meaning score $\mu = (1/3)(\nu + \rho + \kappa)$, which was used in the naming phase of the guessing games. Although the influence of this score was small, it helped to select a form-meaning association in case of an impasse. Where κ was kept constant, ν and ρ were increased when the category was distinctive (ν) and when it was used successfully in the naming phase (ρ); they were lowered otherwise. Exact details of these updates can be found in [20].

If the distinctive category set $\mathrm{DC} \neq \emptyset$, the discrimination game was a success and the DC was forwarded to the naming phase of the guessing game. If a category c was used successfully in the guessing game, the prototype $\mathbf{c}$ of this category was moved toward the feature vector $\mathbf{f}$ of the topic

$$\mathbf{c} := \mathbf{c} + \epsilon \cdot (\mathbf{f} - \mathbf{c}), \tag{7}$$

where $\epsilon = 0.1$ is a constant step size with which the prototype moved towards $\mathbf{f}$. This way the prototypes became more representative samples of the feature vectors it categorized.

The discrimination game as implemented here differs from the implementation of Steels [15] mainly in the representation and construction of categories. Steels used binary trees to split up the sensory (or feature) channels rather than using prototypes. The reason for using prototypes is that the world as sensed by a robot is not binary and splitting up categories in binary trees seems therefore inappropriate. In addition, Steels allowed categories to be formed in only one dimension or in any combination of the different feature dimensions; while in this implementation the categories were always n-dimensional.

It is important to realize that all processing up to this point was carried out by each robot individually. This way, the ontologies, contexts and distinctive category sets differed from robot to robot.

3.4. *Production*

After both robots obtained distinctive categories of the (potential) topic(s), the speaker tried to communicate its topic based on its lexicon. The lexicon L was defined as a set of form-meaning associations: $L = \{\mathrm{FM}_i\}$, where $\mathrm{FM}_i = \langle F_i, M_i, \sigma_i \rangle$ was a lexical entry. Word-form F_i was made from an arbitrary combination of consonants and vowels taken from the alphabet, meaning M_i was represented by some category, and association score $\sigma_i \in \langle 0, 1 \rangle$ was a real number that indicated the effectiveness of the lexical entry based on past interactions. Each form could be associated with multiple meanings, and each meaning could have associations with more than one form.

The speaker of the guessing game ordered the distinctive category set DC based on the meaning score μ. It selected the distinctive category with the highest meaning score and searched its lexicon for form-meaning associations of which the meaning matched this distinctive category. If it failed to find such an element, the speaker first considered the next best distinctive category from the ordered DC. If all distinctive categories were explored and still no entry was found, the speaker could invent a new form as will be explained in Section 3.7.

If there were one or more lexical entries that fulfilled the above condition, the speaker selected the entry that had the highest association score σ. The form that was thus produced was uttered to the hearer. In the on-board implementation this was done using radio communication, off-line the utterance was a shared variable.

3.5. *Interpretation*

On receipt of the utterance, the hearer searched its lexicon for entries for which the form matched the utterance *and* the meaning matched one of the distinctive categories of the potential topics. If it failed to find one, the lexicon had to be expanded, as explained in Section 3.7.

If the hearer found one or more entries, it selected the entry that had the highest score $\Sigma = \sigma + \alpha \cdot \mu$, where $\alpha = 0.1$ is a constant weight. The potential topic that was categorized by this meaning was selected by the hearer as *the* topic of the guessing game. That is, this segment was what the hearer guessed to be the subject of communication.

3.6. *Corrective feedback*

The effect of the guessing games was evaluated by the corrective feedback. If the speaker had no lexical entry that matched a distinctive category, or if the hearer could not interpret the speaker's utterance because it did not have a proper lexical entry in the context of the game, then the guessing game was a failure. The guessing game was successful when both robots communicated about the same referent. So if the hearer interpreted the utterance and thus guessed the speaker's topic, the robots had to evaluate whether they communicated about the same referent.

In previous work there have been various attempts to implement the corrective feedback physically as a pointing behavior [17]. All these attempts, however, failed. In order not to focus too long on this problem and to prove the principle, it was assumed for the time being that the robots could do this and the verification was simulated. Naturally, this problem needs to be solved in the future.

The corrective feedback was simulated by comparing the feature vectors of the two robots relating to their topics. If the features with value 1 matched for both topics, this means that the topics corresponded to the same referent and the guessing game was considered successful. If the hearer selected an inconsistent topic during the interpretation, then there was a *mismatch in referent* and the guessing game failed.

3.7. Lexicon adaptation

Depending on the outcome of the game, the lexicon of the two robots was adapted. There were four possible outcomes/adaptations:

(1) *The speaker had no lexical entry.* In this case the speaker created a new form and associated this with the distinctive category it tried to name. This was done with a certain probability, which was kept constant during the experiment at $P_s = 0.1$.

(2) *The hearer had no lexical entry.* The hearer adopted the form uttered by the speaker and associated this with the distinctive categories of a randomly selected segment from its context.

(3) *There was a mismatch in referent.* Both robots adapted the association score σ of the used lexical entry by $\sigma := \eta \cdot \sigma$, where $\eta = 0.9$ is a constant learning parameter. In addition, the hearer adopted the utterance and associated it with the distinctive categories of a different randomly selected segment.

(4) *The game was a success.* Both robots reinforced the association score of the used entry by $\sigma := \eta \cdot \sigma + 1 - \eta$. In addition, they lowered competing entries (i.e. entries for which either the form or the meaning was the same as in the used entry) by $\sigma := \eta \cdot \sigma$. The latter update is called lateral inhibition.

The coupling of the naming phase with the discrimination game and the sensing part makes that the emerging lexicon is grounded in the real world. The robots successfully solve the physical symbol grounding problem in some situation when the guessing game is successful, because only in those case a semiotic triangle (Fig. 1) is constructed completely in a functional—and thus meaningful—sense.

4. Experimental results

An experiment was done for which the sensory data of the sensing phase during 1000 guessing games was recorded. From this data set it was calculated that the a priori chance for successful communication was 23.5% when the robots randomly chose a topic. Because the robots did not always detect all the light sources that were present, their context was not always coherent. This incoherence caused an upper limit to the success rate that could be reached, called the *potential understandability*, which was 79.5% on the average.

The 1000 recorded situations were processed off-line on a PC in 10 runs of 10,000 guessing games. Fig. 4 (left) shows the average communicative success (CS) and discriminative success (DS) of the 10 runs. The CS measures the number of successful guessing

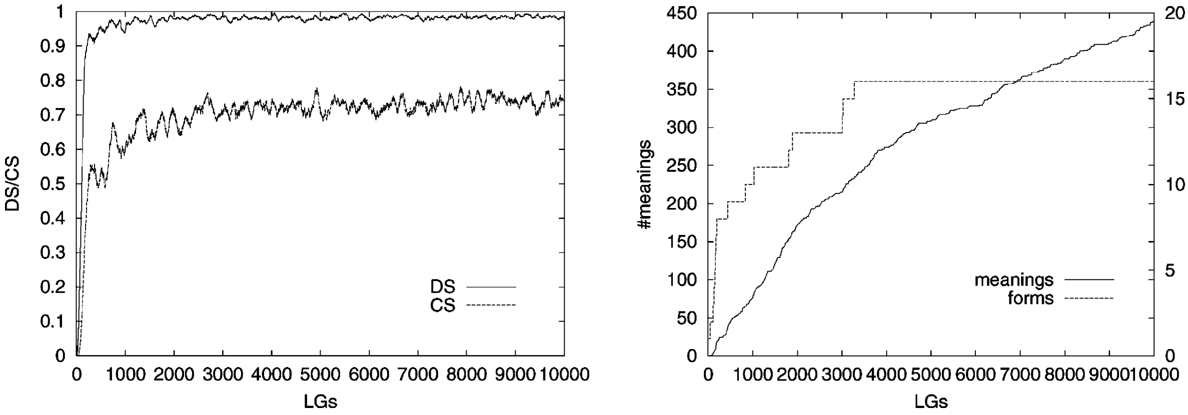


Fig. 4. (Left) The CS and DS of the experiment. (Right) The evolution of the number of meanings and forms that were used successfully by the robots in one run of the experiment.

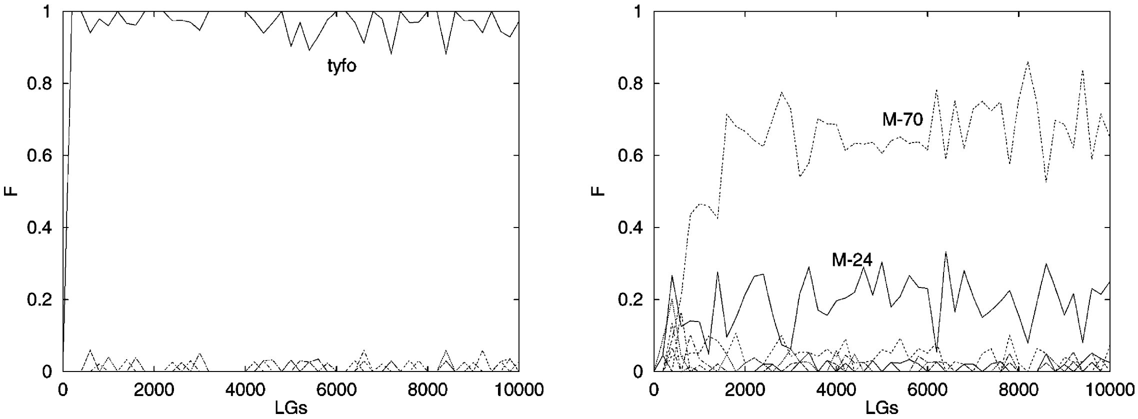


Fig. 5. The referent-form competition diagram (left) shows the competition between forms to name referent light source L1. The referent-meaning diagram (right) shows the competition between meanings to interpret light source L1. In both diagrams the y-axis shows the occurrence frequencies of successfully used forms or meanings over the past 200 games relative to the occurrence of the referent. The x-axis shows the number of games played.

games, averaged over the past 100 games. The DS measures the number of successful discrimination games, also averaged over the past 100 guessing games. As the figure shows, the DS reaches a value near 1 very fast. Hence, the robots were well capable of finding distinctive categories for the sensed light sources. The CS was somewhat lower. It increased towards a value slightly below 0.8 near the end. Since this is close to the potential understandability, the robots were capable to construct a shared lexicon within its limits.

Fig. 4 (right) shows the number of different meanings and forms that were used at least once successfully in one run of the experiments. As the figure shows, the number of meanings used were much higher than the number of used forms. The robots used up to 450 meanings in relation to the four referents, while they only used 16 forms to name them. So, there are approximately 28× more meanings used than forms. Although the robots used about 450 meanings to distinctively categorize the four light sources, further analysis revealed they only used about 20–25 meanings frequently. In addition, only six or seven forms were used regularly. So, the robots named each referent consistently with one or two forms.

The competition diagram of Fig. 5 (left) shows how the occurrence frequencies of the used forms to name one of the referents evolved during one run of the experiment. As this figure makes clear, the most frequently used form "tyfo" clearly won the competition to name light source L1. At the bottom of the diagram, other forms reveal a weak competition. Similar competitions can be observed for the other referents [20,22]. Fig. 5 (right) shows that the competition between meanings to categorize a referent is stronger, which would be expected given Fig. 4 (right). More experimental results can be found in [20,22].

5. Discussion

In this section, I will discuss why the notion of semiotic symbols is useful in relation to the anchoring problem. The discussion will be based on the observation that semiotic symbols can be constructed by optimizing the anchor between their forms and the objects they stand for; thus solving the object constancy problem. Furthermore, I will explain how the use of semiotic symbols can model the phenomenon of family resemblance.

In this paper the 'alternative' definition of symbols as semiotic symbols is adopted to provide the possibility to construct anchors between symbols (or forms as I call them) and the real world. But is there any advantage of using semiotic symbols over the traditional symbols in relation to the anchoring problem? In the original anchoring problem [5], anchors are sought between symbols and perceptual features, while the symbols' relations to the real world objects are somewhat brought to the background. The experiment of this paper revealed that it is the relation between the form and the real world object that is being optimized in terms of a one-to-one relationship. The relation between the form and the sensory data (or even the categories) does not reveal this optimization. I do not argue that the relationship between form and sensory data is unimportant, but I do want to argue that the relation between form and referent is the one we should care for.

Before explaining why the relation between form and referent is crucial, I will elaborate on the importance of the relation between sensory images and forms. The processes between sensing and feature extraction are extremely important because these transform the raw sensory data into more manageable feature vectors that additionally bear some invariant information concerning the referents. In addition, the intermediate representations of categories are important to allow the optimization between form and referent, because the discrimination games function—like the sensing, segmentation and feature extraction—as a sieve. This sieve enables the robots to bind the numerous variation of the sensing to more informative granules that are less numerous. These granules are, although still numerous, more manageable than the raw sensory data; thus allowing to close the coupling between referents, meanings and forms more easily.

The optimization between referent and form, however, is the most dominant process for the construction of consistent anchors between these two elements. To understand how this optimization works, it is important to realize that robots try to construct a lexicon that they can apply in different contexts. The lexicon is constructed through the interplay of adaptations under selective pressures and pragmatic language use. In the experiment, anchors were formed between referents and meanings, between meanings and forms; and between forms and referents. The results show that many anchors were used between referents and meanings, and between meanings and forms. However, when forms were used, they were well anchored to the referents they name. Failures in the discrimination game caused the emergence of so many meanings, because every time a discrimination game fails, a new category was added to the ontology. Many of them were associated with a form when they became distinctive in a later discrimination game. As associations were selected during a guessing game when their meanings fitted in the context—even if the scores were not high—a lot of these meanings were used successfully in the game.

The same context dependency causes the emergent tendency that the robots do not use so many forms, despite the variability of the acquired contexts during different games and between the robots. This can be understood by realizing that when one robot categorizes a referent differently in different guessing games, this does not necessarily mean that the other robot finds different distinctive categories. When the robot that uses the same distinctive category on different occasions, it will most likely use the same form to express this meaning too. This allows the other robot to use the form in association with the two different meanings successfully, as the game is context dependent. When such situations occur frequently, this, in turn, allows the robots to use more meanings than forms. These emerging dynamics of the lexicon can be classified as semiotic dynamics and illustrates how conceptual development is, at least to some extent, dependent on language acquisition and language use and vice versa. This is conform the—in a weaker version—revived Sapir-Whorf thesis [3]. A similar argument in favor of this weaker version of the Sapir-Whorf thesis was made in another study using language games [1]. In that study it was shown that agents developed a shared categorization of the color space when they used language, but a distinctive categorization when they developed categories without engaging in guessing games.

The optimization between referent and form solves, at least to some extent, the notion of object constancy: How can an object be recognized as being the same when different views of such an object can result in dramatically different sensory stimuli, for instance, because it is partly obscured? Fig. 6 (left) illustrates how the semiotic dynamics can explain the solution to the object constancy problem. In the experiment, the robots detected the light sources from different positions, resulting in different sensings—illustrated as the continuum of sensings P in Fig. 6 (left)—which may yield different meanings M1 and M2. Nevertheless, the system identifies the objects consistently, because the one-to-many relations between form and meaning converge at the level of form and referent.

The results of the experiment in this paper show that minimal autonomous robots can develop a shared set of semiotic symbols from the bottom-up by optimizing their anchors between forms and referents. However, one of the driving forces for this optimization—the corrective feedback—was simulated. This is a major shortcoming as the method used—inspecting each other's internal states—is unrealistic and may undermine the principle. Nevertheless, the assumption was adopted to test the principles of the underlying bootstrapping mechanisms and not to get stuck on solving this problem. A solution may come from applications were robots evaluate the corrective feedback using task-oriented behaviors, as was recently investigated in simulations [21]. In these simulations, the feedback came from the effect of the task that the agents had to perform using the evolved language.

The semiotic dynamics of the guessing games help to solve the object constancy problem, but it may also help to explain another interesting phenomenon observed in cognitive science, namely *family resemblance* [23]. Family resemblance is the observation that seemingly different things are called the same without being ambiguous, like the meaning of *games*. Where soccer and chess are typical games, a game like swinging is not typical. Swinging lies near the border of the 'conceptual space' of games, e.g. referent R1 in Fig. 6 (right). It has no direct resemblance with games like soccer and chess, e.g. R2 and R3—but it has some resemblance with other games that in turn do have resemblance with soccer and chess. Such categorization process can be explained with the one-to-many relations between form and meaning. The word "games" is associated with different meanings for soccer, chess and swinging. The successful use of these meanings in different situated language games allows the system to emerge a family of resemblance. Optimization here should be made on the relation between a form and different referents. This optimization can be realized through the use of language.

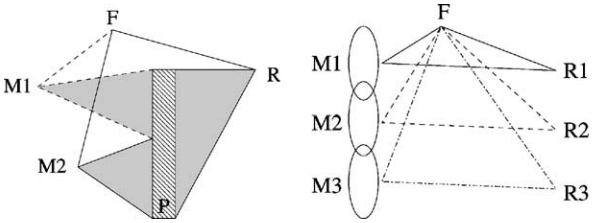


Fig. 6. Illustration of two semiotic relations between referent R, meaning M and form F. The left figure shows the continuum of possible views of P of referent R as displayed as a rectangle. Some part of the rectangle may be interpreted by M1 and another by M2. When both meanings relate to the same form, this mechanism solves the problem of object constancy. The right figure shows how the model may explain family resemblance. The ovals should be interpreted as Venn diagrams of the meanings M1 and M2.

Concluding, the above discussions provide many arguments in favor of using semiotic symbols over the traditional symbols with respect to anchoring. The most important argument is that in the construction of semiotic symbols, anchors between forms and reality are implicitly being optimized, rather than optimizing anchors between symbols and sensory images.

6. Conclusions

This paper illustrates how a small group of autonomous robots can develop a set of shared semiotic symbols in a bottom-up fashion by engaging in adaptive language games. The semiotic symbols the robots construct are defined by physical anchors between referents and meanings, and between meanings and forms, which yield a non-physical anchor between form and referent. The use of semiotic symbols allows a profitable optimization to find, track and (re)acquire anchors between forms and referents, rather than between forms and sensory images as proposed in the original description of the anchoring problem [5].

The experiments show how a consistent construction of semiotic symbols is positively influenced by their use in language. Through the use of language, the forms are shared externally to the robots. In addition, the robots share the reference of their communication through the received feedback. These external factors, together with the internal adaptations influence the way the robots organize their conceptual spaces. Thus their conceptual development is influenced to a large extent by their language use, hence providing an argument in favor of a weak interpretation of the Sapir-Whorf thesis as discussed in [3].

To further broad our understanding on the emergence of semiotic symbol systems in language use, additional research is required on the emergence of compositionality as this is one of the key aspects of human language use. Future research should concentrate on how compositional structures can be grounded in the sensorimotor flow through grammatical language use. In addition, more research is required to design robotic applications that are capable of verifying the effectiveness of their language use in order to provide corrective feedback autonomously. Although further research is required to improve and scale the model, adaptive language games provide a potentially valuable technology for a bottom-up approach towards anchoring semiotic symbols.

Acknowledgements

The research has been carried out at the AI Lab of the Vrije Universiteit Brussel. I would like to thank Luc Steels very much for enriching me with his expert advice and inspiration. I am also indebted to all other colleagues of the AI Lab for their support in one way or another. Ida Sprinkhuizen-Kuyper, Rens Kortmann and three anonymous reviewers are thanked for their critical comments that helped to improve this paper.

References

[1] T. Belpaeme, Factors influencing the origins of colour categories, Ph.D. Thesis, Artificial Intelligence Lab, Vrije Universiteit Brussel, 2002.

[2] A. Billard, Imitation: a means to enhance learning of a synthetic proto-language in an autonomous robot, in: K. Dautenhahn, C.L. Nehaniv (Eds.), Imitation in Animals and Artifacts, MIT Press, Cambridge, MA, 2001.

[3] M. Bowerman, S.C. Levinson (Eds.), Language Acquisition and Conceptual Development, Cambridge University Press, Cambridge, 2001.

[4] R.A. Brooks, Elephants don't play chess, Robotics and Autonomous Systems 6 (1990) 3–15.

[5] S. Coradeschi, A. Saffiotti, Anchoring symbols to sensor data: preliminary report, in: Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-2000), Austin, 2000, pp. 129–135.

[6] T.M. Cover, P.E. Hart, Nearest neighbour pattern classification, Institute of Electrical and Electronics Engineers Transactions on Information Theory 13 (1967) 21–27.

[7] T. Deacon, The Symbolic Species, W. Norton and Co., New York, 1997.

[8] S. Harnad, The symbol grounding problem, Physica D 42 (1990) 335–346.

[9] D. Jung, A. Zelinsky, Grounded symbolic communication between heterogeneous cooperating robots, Autonomous Robots 8 (2000) 269–292.

[10] C.K. Ogden, I.A. Richards, The Meaning of Meaning: A Study of the Influence of Language upon Thought and of the Science of Symbolism, Routledge & Kegan Paul Ltd., London, 1923.

[11] C.S. Peirce, Collected Papers, vol. I-VIII, Harvard University Press, Cambridge, MA, 1931 (volumes were published from 1931 to 1958).

[12] D.K. Roy, A.P. Pentland, Learning words from sights and sounds: a computational model, Cognitive Science 26 (2002) 113–146.

[13] J.M. Siskind, Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic, Journal of Artificial Intelligence Research 15 (2001) 31–90.

[14] L. Steels, Emergent adaptive lexicons, in: P. Maes (Ed.), From Animals to Animats 4, Proceedings of the Fourth International Conference on Simulating Adaptive Behavior, 1996, MIT Press, Cambridge, MA.

[15] L. Steels, Perceptually grounded meaning creation, in: M. Tokoro (Ed.), Proceedings of the International Conference on Multi-Agent Systems, AAAI Press, Menlo Park, CA, 1996.

[16] L. Steels, F. Kaplan, AIBO's first words. The social learning of language and meaning, Evolution of Communication 4 (1) (2000) 3–32.

[17] L. Steels, P. Vogt, Grounding adaptive language games in robotic agents, in: C. Husbands, I. Harvey (Eds.), Proceedings of the Fourth European Conference on Artificial Life, MIT Press, Cambridge, MA, and London, 1997.

[18] M. Tomasello, The cultural origins of human cognition, Harvard University Press, Cambridge, MA, 1999.

[19] P. Vogt, Bootstrapping grounded symbols by minimal autonomous robots, Evolution of Communication 4 (1) (2000) 89–118.

[20] P. Vogt, Lexicon grounding on mobile robots, Ph.D. Thesis, Vrije Universiteit Brussel, 2000.

[21] P. Vogt, Anchoring symbols to sensorimotor control, in: H. Blockdeel, M. Denecker (Eds.), Proceedings of the 14th Belgian/Netherlands Artificial Intelligence Conference, BNAIC'02, 2002, pp. 331–338.

[22] P. Vogt, The physical symbol grounding problem, Cognitive Systems Research 3 (3) (2002) 429–457.

[23] L. Wittgenstein, Philosophical Investigations, Basil Blackwell, Oxford, UK, 1958.

[24] H. Yanco, L. Stein, An adaptive communication protocol for cooperating mobile robots, in: J.-A. Meyer, H.L. Roitblat, S. Wilson (Eds.), From Animals to Animats 2, Proceedings of the Second International Conference on Simulation of Adaptive Behavior, MIT Press, Cambridge, MA, 1993, pp. 478–485.

Paul Vogt received his M.Sc. degree in cognitive science and engineering at the Rijksuniversiteit Groningen, The Netherlands, in 1997. He obtained his Ph.D. from the Artificial Intelligence Laboratory of the Vrije Universiteit Brussel, Belgium. Currently, he is a postdoc researcher with the Institute for Knowledge and Agent Technology at the Universiteit Maastricht, The Netherlands. His main research interests are in symbol grounding, language evolution and robotics.

ELSEVIER

Robotics and Autonomous Systems 43 (2003) 121–132

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Grounding inference in distributed multi-robot environments

Aaron Khoo*, Ian Horswill

Computer Science Department, Northwestern University, 1890 Maple Avenue, Evanston, IL 60201, USA

Abstract

Traditional symbolic reasoning systems are typically built on a transaction model of computation, which complicates the process of synchronizing their world models with changes in a dynamic environment. This problem is exacerbated in the multi-robot case, where there are now *n* world models keep in synch. In this paper, we describe an inference grounding and coordination mechanism for robot teams based on tagged behavior-based systems. This approach supports a large subset of classical AI techniques while providing a novel representation that allows team members to share information efficiently. We illustrate our approach on two problems involving systematic spatial search.

Keywords: Autonomous mobile robots; Multi-robot teams; Behavior-based control

1. Introduction

Autonomous robots that reside in complex, dynamic environments must anchor the abstract representations they use to actual physical objects. The world around the robot continually changes, and its sensory systems must track those changes. In turn, the robot's control systems must be ready to alter plans and actions to suit its changing model of the world. Traditional symbolic reasoning systems are typically built on a transaction-oriented model of computation. Knowledge about the world, or the "world model", is stored in a database of assertions in some logical language, indexed perhaps by predicate name [20]. Populating this database from a highly dynamic environment is a difficult and non-trivial problem [13]. In this paper, we argue that the issue of world model synchronization is even more problematic in a cooperative multi-robot team, and we propose an alternative coordination approach based on periodic knowledge-base broadcast. Finally, we describe an implementation of this approach that was used on two tasks involving systematic spatial search.

2. World model synchronization

When changes in the environment occur often, the world model must also be updated frequently, or the reasoning system will operate on stale data. Additionally, assertions in the world model database can be dependent on other assertions. For example, the assertion that an area is safe could depend on the assertion that the robot does not currently observe any predators in the area. If the latter assertion is withdrawn, then the former must be too. Hence, each update from the perceptual systems can trigger a cascade of further transactions, resulting in additional load on the system. In principle, modifying such a system to track changes in the environment would require recording dependencies between stored assertions and their justifications such that when the perceptual system added

* Corresponding author. Fax: +1-847-491-5258.
E-mail addresses: khoo@cs.northwestern.edu (A. Khoo), ian@cs.northwestern.edu (I. Horswill).

0921-8890/03/$ – see front matter © 2003 Elsevier Science B.V. All rights reserved.
PII: S0921-8890(02)00354-8

or retracted an assertion, the reasoning system could enumerate and update the set of existing assertions affected by the change. This is a sufficiently complicated process that we know of no implemented physical robots that do it.

Keeping the knowledge-base synchronized with the external environment becomes even more difficult in cooperative activity. Rather than one robot with a single knowledge-base, we now have *n* robots with *n* knowledge-bases to keep consistent both with the world and with one another. While others have shown that there exist cases in which agents can achieve cooperation despite discrepancies in their world models [22], there is currently no concrete theory on what tasks are achievable despite inconsistent world models. Therefore, we conservatively assume that failure to properly coordinate the knowledge-bases could lead to *system delusion* [12], i.e. the databases are now inconsistent, and there is no obvious way to repair them, resulting in failure to coordinate activity.

An analysis of asynchronous peer-to-peer replicated databases by Gray et al. [11] suggests that a potential problem exists. A *conflict* occurs when two different databases attempt to update the same object, or race to install their updates at other databases. Whenever conflicts occur, the replication mechanism must detect this and somehow reconcile the two transactions so that their updates are not lost. Under the following simple assumptions

- The databases are updated through lazy group replication, i.e. the originating database updates its entries, and then propagates the update to other replicas asynchronously.
- Each node updates any other database location with equal probability.
- All nodes impose an equal load on the system.
- There are a fixed number of objects per transaction.

Gray et al. were able to show that the conflict rate per second is:

$$\mathrm{O}\left(\frac{r^2a^3tn^3}{s}\right),$$

where *r* is the number of transactions per second initiated by each node, *a* is the number of locations updated per transaction, *t* is the time required to complete an update, *s* is the number of distinct entries in the database and *n* is the number of nodes (which, in our case, is equivalent to the number of robots) in the system.

The critical point here is that the number of conflicts encountered by the system increases with the third power of the number of nodes or robots. As Gray et al. point out, “having the reconciliation rate rise by a factor of 1000 when the system scales up by a factor of 10 is frightening”. While the two models are not exactly analogous, there is sufficient overlap between the problem of synchronizing different knowledge-bases and the issue of distributed database replication to elicit concern.

Furthermore, note that message propagation times are not presently part of the conflict model as presented above. If message delays were added to the model, then each transaction would last longer, hold more resources and generate more conflicts. Moreover, mobile robots necessarily communicate via wireless links, which are well known to have higher error rates [9,26], and hence higher message delays, than their wired counterparts. This analysis suggests that we could potentially face serious scalability issues for any physical multi-robot system with a database-driven knowledge model. The work necessary to reconcile the conflicts that could arise as team members tried to communicate knowledge to other members could eventually overwhelm the robots, or leave them badly out of synch.

3. Related work

Recent progress has been made towards the development of a formal framework for anchoring symbols [7,8]. However, most implemented physical systems take the approach of equipping the symbolic system with a set of domain-dependent epistemic actions that fire task-specific perceptual operators to update specific parts of the knowledge-base. The programmer designing the knowledge-base is responsible for ensuring that the proper updates are done, i.e. the right epistemic actions are fired at the appropriate times. This alleviates some of the difficulties of getting information into the knowledge-base in a timely manner. However, any mistakes by the programmer will lead to inconsistencies between the knowledge-base used by the symbolic system and the external environment.

Tiered architectures, such as [2,4,6], that combine symbolic and behavior-based systems inherit these model coherency issues, because their symbolic layer still relies on a database-driven world model for its reasoning process. As we pointed out in the previous section, these issues are exacerbated in a cooperative environment where multiple knowledge-bases have to be synchronized.

We feel that these knowledge-base synchronization issues have led to a paucity of physical multi-robot systems utilizing symbolic reasoners. There has been excellent work done on coordination protocols for co-operative agents [5] in the multi-agent community, and these protocols have been successfully used for agent teamwork in simulation environments [23]. However, the only physical multi-robot team that utilizes both a symbolic reasoner (in a tiered architecture) and active communication that we know of is the DIRA system [21].

Most existing multi-robot controllers implemented on physical systems focus on extending traditional behavior-based techniques [1] to a team environment (for example see [3,10,19]). Behavior-based systems allow rapid response to changes in the environment due to tight sensor–actuator integration. Many of these behavior-based systems also obey circuit semantics, which means their control programs are generally implemented as feed-forward circuits. This simplifies the communication structure necessary to maintain coordination between team members. Essentially communication in these behavior-based multi-robot controllers is reduced to virtual wires connecting the appropriate circuitry on one team member to another's (see Fig. 1). The wires carry relevant information from a robot to its counterparts. Conversely, each robot views its teammates simply as additional sensory input, and integrates the incoming information as appropriate. Conveniently, virtual wires can be simulated on physical robots using a broadcast communication mechanism such as User Datagram Protocol (UDP).

However, this convenience is not without cost. The strengths of the behavior-based approach are also its weakness. Circuit semantics impose a propositional representation on the reasoning system, i.e. representations without predicate/argument structure. Propositional representation makes most reasoning and planning tasks both difficult and clumsy since they require redundant copies of the system for each possible argument to a predicate or action [18]. Since most multi-robot controllers are extensions of behavior-based techniques, they inherit the same issues from the basic underlying architecture.

Some systems have attempted to solve the synchronization problem through techniques other than

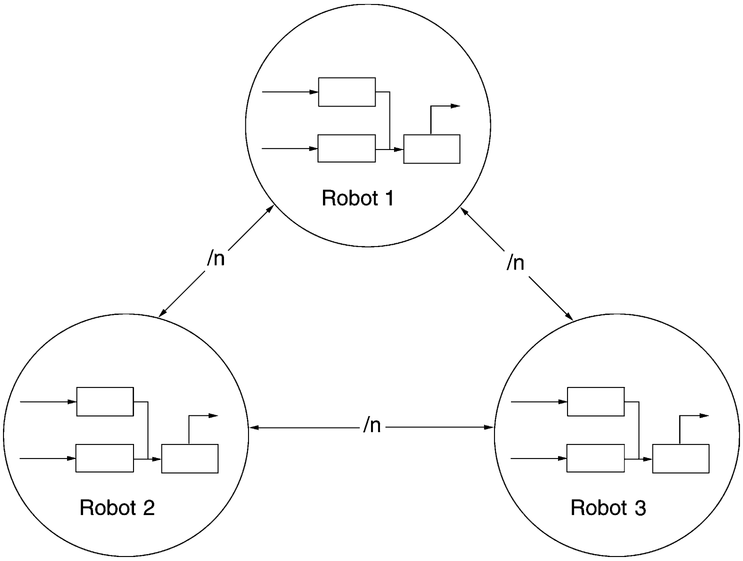


Fig. 1. Communication via virtual wires.

active communication. Wie [25] proposes an approach that achieves coordination through teammate observation and plan inference. Ronald Kube and Zhang [17] utilizes stochastic techniques that allow the individual robots to achieve a global goal through the use of simple non-interference behaviors. Some systems in the robocup small-sized robot league utilize a traditional symbolic reasoner that relies on a central shared world model [24]. In this case, reasoning is performed at a central server location where the master knowledge-base is located, and then actions are transmitted to the individual robots. Little, if any, reasoning is done on the client side.

In this paper, we only consider team models where members are fully autonomous entities with independent decision making ability, i.e. the robots are not reliant upon a central reasoner. Each robot is responsible for deciding its own course of action. Furthermore, we are focusing on domains where passive communication or stochastic techniques are insufficient. That is, team members will have to coordinate through the use of explicit communication.

4. HIVEMind

While designing our current multi-robot control architecture, we wanted to utilize as many useful features of traditional symbolic AI systems as possible on our robots. Specifically, we would like to have the ability to utilize predicate/argument structure in our representations. However, we also wanted to avoid importing the model coherence and database synchronization issues that symbolic systems encounter. That is, the symbols utilized in our inference rules should be tightly anchored to updates from the sensory systems as well as incoming information from other team members.

Our efforts in this direction have resulted in HIVEMind (Highly Interconnected Very Efficient Mind), a multi-robot control architecture that supports very efficient sharing of symbolic information between team members. The HIVEMind architecture is built on role-passing [13], a type of tagged behavior-based system [16]. Role-passing provides the developer with a limited set of domain-independent indexical variables (called roles) such as *agent*, *patient*, *source*, *destination*, etc. When a role is bound to an object, a tracker is dynamically allocated to it and tagged with the name of the role. Since the number of roles is relatively small, we can represent the extensions of unary predicates as bit-vectors, with one bit representing each role. This representation allows inference to be performed using bit-parallel operations in a feed-forward network.

Alternatively, for commodity serial hardware, we can represent a unary predicate extension using a single machine word. Inference rules can then be compiled directly into straight-line machine code consisting only of load, store, and bit-mask instructions [13]. While more limited than a full logic-programming system, it does allow us to express much of the kinds of inference used on physical robots today. The inference rules can be completely rerun on every cycle of the system's control loop, allowing the robots to respond to contingencies as soon as they are sensed. The compiled code is sufficiently efficient that inference is effectively free—1000 Horn clauses of five conjuncts each can be completely updated at 100 Hz using less than 1% of a current CPU. In short, role-passing affords us the ability to implement traditional inference rules using circuit semantics.

In addition to allowing very fast inference, this representation allows for very compact storage of a robot's current set of inferences. Unary predicates are stored in one machine word. Function values are represented using small arrays indexed by role. This compactness, combined with the circuit semantic nature of role-passing, allows us to take full advantage of simplified communication mechanism described previously, i.e. virtual wires connecting team members. In fact, for the kinds of tasks currently implemented by multi-robot teams, the representation we use is sufficiently compact to allow all function and predicate values of a robot to fit into a single UDP packet. Robots can therefore share information by periodically broadcasting their entire knowledge-base, or at least all those predicates and functions that might be relevant to other team members.

Knowledge-base broadcast is a simple communication and coordination model that provides each robot with transparent access to every other robot's state, establishing a kind of "group mind". It allows the team to efficiently maintain a shared situational awareness and to provide hard real-time response guarantees; when a team member detects a contingency, other members are immediately informed and respond within one update cycle without the need for negotiation protocols. Moreover, since HIVEMind systems are based on role-passing, multi-robot controllers implemented using this architecture have greater representational power and flexibility than pure behavior-based systems with propositional representations. That is, our communication is not based on passing propositional values such as `see-blue-object` or `see-red-object`, but rather predicates such as `see-object(X)`. Furthermore, since all relevant team knowledge is continuously being rebroadcast, each member's knowledge-base converges to the same state within O(1) time of joining the HIVEMind. This means that team members can be brought online and integrated into the HIVEMind very easily, allowing us to add or subtract team members dynamically. This also implies that, should communication fail for some time, the team would very rapidly return to a common state when it is restored.

Fig. 2 shows an abstract HIVEMind configuration for a two-robot team. Each team member has its own inference network. The network is driven both by its own sensory system and by the incoming data from the other team members. Outputs from the current robot's sensory systems are fed into aggregation functions on other team members. The output from those aggregation functions is then fed into the inference rules which drive the motor behaviors.

The aggregation functions are used to combine information from teammates and sensors into a single coherent output for the inference rules to reason over. In an n robot team, each robot's inference network has n distinct sets of inputs, one generated internally, and the rest received from the robot's teammates. These distinct inputs are first fused into a single set of inputs:

$$K = \beta(k_1, k_2, \ldots, k_n),$$

where the k_i are the tuples of inputs from each robot, K is the final fused tuple, and β is the some aggregation function that performs the fusion. For example, if a particular component of the input was a proposition, the aggregation function might simply OR together the corresponding components of the k_i. Thus the robot would believe the proposition if and only if some robot had evidence for it. In more complicated cases, fuzzy logic or Bayesian inference could be used. Real-valued data is likely to require task-specific aggregation. For example

- The team is assigned to scout an area and report the number of enemies observed. Each team member has a slightly different count of enemy troops. In this case, the best solution is probably to average the disparate counts.
- The task is "converge on the target". Each robot's sensors report a slightly different position for the target. In this situation, it appears to make sense that each team member rely on its own sensor values to track the target and only rely on other robots when the robot's own sensors are unable to track the target, e.g. the target is out of sight.

Fig. 3 shows how aggregation is performed in the actual system. As packets arrive on from other robots, they are unpacked into buffers for their respective

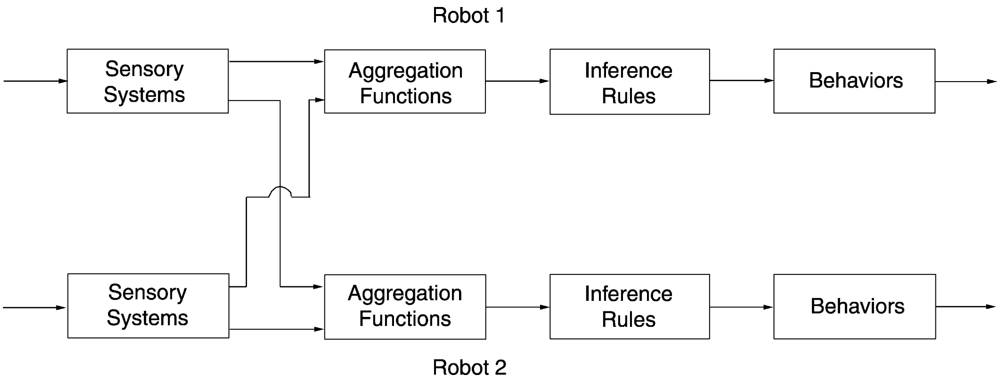


Fig. 2. Abstract view of HIVEMind.

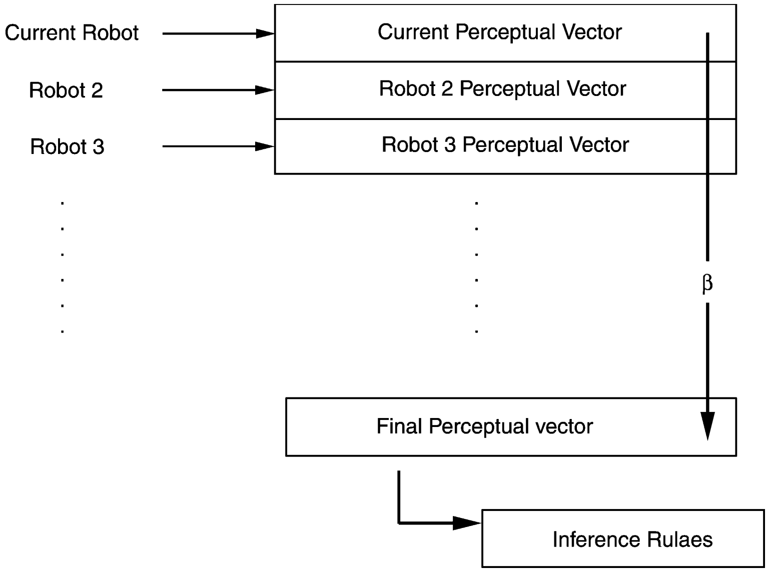

Current Robot
Robot 2
Robot 3
Current Perceptual Vector
Robot 2 Perceptual Vector
Robot 3 Perceptual Vector
β
Final Perceptual vector
Inference Rulaes

Fig. 3. Implementation of HIVEMind virtual wires.

robots, replacing whatever data had been stored previously for that robot. In parallel with this process, the main control loop of the robot aggregates the inputs from each robot and reruns the inference rules on the result. These inference rules then enable and disable low-level behaviors for sensory-motor control. Since the main control loop is performing real-time control, it runs much faster than the 1 Hz update used for communication (10 Hz in our current implementation).

The entire HIVEMind can be considered a single, parallel control network whose components happen to be distributed between the different robot bodies being controlled. Wires crossing between bodies are simulated using the RF broadcast mechanism, so that each member of the team is “connected” to every other member in a web-like structure of virtual wires. In our current implementation, each robot broadcasts its sensory data and state estimates in a single UDP packet at predefined intervals. Presently, broadcasts are made every second. Faster or slower rates could be used when latency is more or less critical. However, 1 Hz has worked well for our applications. To reiterate, we expect that currently implementable robot systems could store all the sensory inputs to the inference system in a single UDP packet (1024 bytes). Current autonomous robots are severely limited in their task capabilities, and hence their communication needs, by their sensors and actuators. Therefore, we feel that there is plenty of available bandwidth for communication in the foreseeable future. As robots develop more complicated sensoria, it may be necessary to use more complicated protocols, perhaps involving multiple packets, or packets that only contain updates for wires whose values have changed since the last transmission. For the moment, however, these issues are moot.

Given the current single-packet-protocol, the aggregate bandwidth required for coordination is bounded by 1 KB/robot/s, or about 0.1% of a current RF LAN per robot. Thus robot teams on the order of 100 robots should be practical from a communication standpoint. However, hardware failure limits most current robot teams to less than 10 members, so scaling limits are difficult to test empirically.

It may seem inefficient for each robot to have its own separate copy of the inference network. However, to have a single robot perform each inference and share the results would require much more complicated coordination protocols [5] analogous to the multi-phase commit protocols used in distributed database systems. Since communication bandwidth is a scarce resource and inference in our system is essentially free, it is more efficient for HIVEMind robots to perform redundant computation.

5. Implementation

5.1. Overview

We have implemented the HIVEMind system on a robot team that performs two tasks:

- Find object: The team systematically searches for a brightly colored object in a known environment. Team members explore the environment in a systematic manner until one of the team members locates the object or all searchable space is exhausted. When the object is found, all team members converge on its location.
- Town crier: This task involves making announcements in the same known environment. The team cooperatively travels to each landmark on a map and makes an announcement at every landmark.

In both cases a human user is responsible for indicating the current task to perform and supplying any required parameters for that task, e.g. the properties of the object to be found in the former task. The human interacts with the team through a user console, which appears as an additional, albeit non-performing, member of the team. When user input is entered into the console, that information is passed through the virtual wires to all team members. We have tested both tasks with a two robot team. The code for this system was written in a combination of GRL [14] and Scheme, although low-level vision operators were written in C++.

5.2. Hardware

The robotic bases used in this experiment are first generation Real World Interface (RWI) Magellan bases. The Magellan provides sonars, infrared sensors and bump switches; a total of 16 each, arrayed around the circular base. Vision is provided by a ProVideo CCD camera, connected by a Nogatech USB video capture adaptor cable to a laptop. The laptops are Dell Latitudes with Pentium II 450 MHz processors, 384 MB of RAM and 11 GB hard drives. They run Windows98, and communicate with the base through a serial cable. Remote communication is provided by Lucent Orinoco Silver wireless Ethernet cards that feature an 11 Mbps data transfer rate under the IEEE 802.11b standard.

5.3. User console

The Command Console for the HIVEMind team is based on the Cerebus project [15]. It provides a natural language interface for the human user and allows commands such as ‘find green ball’ or ‘announce “talk at 7!”’ to be entered. The task is bound to the `activity` role, and any arguments are bound to other appropriate roles, e.g. green would be bound to `object` in the former example. The current bindings are represented in a list form and transmitted on a virtual wire to all members to the team. The console appears as another robot to other team members, albeit one that is not doing any physical work. The user console also provides status information in the form of display windows based on the broadcast knowledge it is receiving from other team members. Using this interface, the human commander can inject new information into the team, as well as receive data about the current state of the “group mind”.

5.4. Perceptual systems

The sensory and memory systems are divided into “pools”, which are useful abstractions for grouping perceptual systems or descriptions of objects. Note that we do not make any unique claims about pools; they are simply convenient abstractions for implementing role-passing. The pools drive the inference rule network, which in turn drives the low-level behaviors that actually control the robot. Fig. 4 shows a high-level view of the system. The action pool stores a set of reified user-defined plans that can be bound to roles at runtime. These plans can then be run by calling the role to which they are bound. For example, the `find` plan is bound to the role `activity` when the user enters “find green ball” at the console. The binding is passed via virtual wire to the individual team members. So, when the control system calls `activity`, it would run the `find` plan. There are currently two plans in the action pool: `find` and `announce`.

The color pool stores color coordinates of different objects in a format suitable for use by the visual tracking system. The color of a desired object can be specified by binding a given color description in the pool to the role of the object. For example, when the user directs the team to seek a green ball, the term green

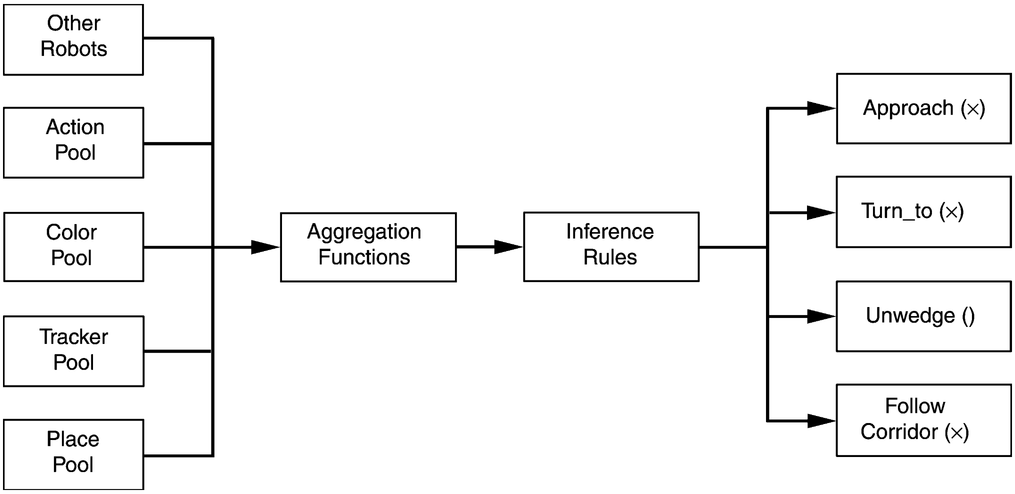


Fig. 4. Control flow from sensors to behaviors in a single robot.

is bound to an appropriate role. The bindings are then automatically passed over the network to the robots. The color pool presently contains descriptions for red, green, and blue objects, and is only used for the find object task.

The tracker pool consists of a set of trackers that utilize a variant of *k*-means clustering for tracking blobs of color in the robots visual image. These trackers can be allocated and bound to a role. The trackers can drive low-level behaviors with image-plane coordinate of the objects they track. In addition, they generate the low-level predicates `see-object(X)` and `near-object(X)` for input to the inference network. The trackers are used only in the find object task.

The place pool is a probabilistic localization system that uses a topological, i.e. landmark-based, map. Roles can be bound to landmarks and the system can determine the next appropriate waypoint in order to reach a landmark specified by role. The place pool also records the set of landmarks that have been visited with high probability and can determine the closest unvisited landmark. The current map contains 11 landmarks distributed over the west wing of the 3rd floor of the Northwestern Computer Science Department.

5.5. Communication

Both tasks require communication of the following:

- The current role bindings, including bindings for the current activity or task, and any bindings for pertinent arguments.
- A bit-vector specifying the set of landmarks that the robot has personally visited.
- The bit vector for the `see-object(X)` predicate.
- An array representing the `location(X)` function, which give the two nearest landmarks, if known, for any role `X`.

All of these are low-level outputs of the various pools, except for the current role bindings, which has to be stored on a separate latch on the user console. When the team is performing the town-crier task, the latter two communication structures, i.e. `see-object(X)` and `location(X)`, are not utilized for reasoning.

6. Inference rules

The inference rules for both tasks are fairly simple. This is partly due to the continual recomputation of inferences, which alleviates the need for some error detection and recovery logic that would otherwise be necessary. The inference rules for the find object task are:

1. If `see-object(X)` is true, then `goto(X)`.
2. If `location(X)` is known, and `see-object(X)` is false, then `goto(location(X))`.

3. If `location(X)` is unknown, and `see-object (X)` is false, then `goto(next-unsearched-location())`.

The inference rules for the town-crier task are:

1. If `at-landmark(X)` and `not-announced-at(X)`, then `speak-string()`.
2. If true, then `goto(next-unsearched-location())`.

The function `next-unsearched-location()` returns the current location if there are no new locations to travel to. `Goto()` is a polymorphic action keyed by the type of the argument passed to it. If the argument is bound to a location, then the robot will navigate to that landmark. If the argument is bound to a color in the color pool, then the robot approaches the largest object matching that color in its view. `Goto()` activates the four behaviors described below as necessary to accomplish its current task.

6.1. Behaviors

There are four motor behaviors that drive the robot:

- *Approach* drives to an object specified by role. It attempts to keep the object in the middle of its visual image.
- *Turn-to* swivels the robot to face a new direction. It is used when the robot arrives at a landmark and needs to turn in a new direction to reach another landmark.
- *Unwedge* activates when the robot becomes stuck in some corner unexpectedly. It swivels the robot in the direction in which it thinks has the greatest open space so the robot can continue moving.

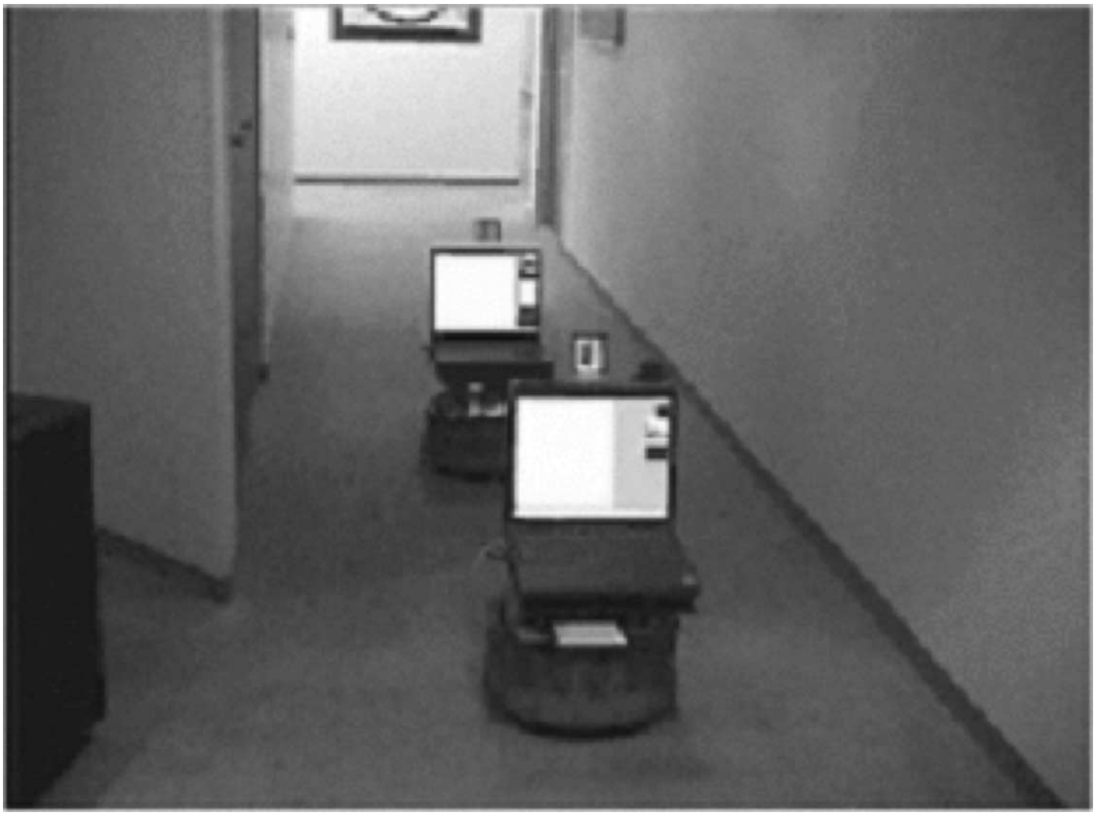


Fig. 5. Two robots leaving from starting point to perform a task.

- *Follow-corridor* navigates the hallways. It tries to remain centered in the middle of the corridor to facilitate easy recognition of environmental features.

The behaviors are arbitrated strictly through a priority stack. Behaviors that are higher on the stack have higher priority, and, if active, will be chosen to run over those of lower priority. Since HIVEMind always ensures that all team members are up-to-date on the current situation, each robot always knows the appropriate behavior to activate for the current situation and no conflict between team members arises.

6.2. Results

We have tested the system with a three-member team consisting of two robots and the command console (Fig. 5). The team was tested in the west wing of the 3rd floor of the Computer Science Department building. The wing consists of a network of six corridors spanning an area approximately 6 m × 20 m with an aggregate path length of 50 m. The network of corridors is represented by 12 landmarks in the topological map showing the locations of features such as corners and intersections. The robots drive at approximately 1 m/s on straight-aways, although stopping for ballistic turns at corners and intersections somewhat reduces their mean velocity. Sensing, inference and control decisions are each performed at 10 Hz.

In the find object experiments, all team members were started from a central point at the extreme east end of the wing (Fig. 6). The goal object, a green ball, was placed out of view, 15–20 m from the starting point. The object was always at least two corridors and three landmarks away from the starting point. When the command “find green” was entered on the command console, the robots begin a systematic search of the wing for the goal object. Unlike stochastic

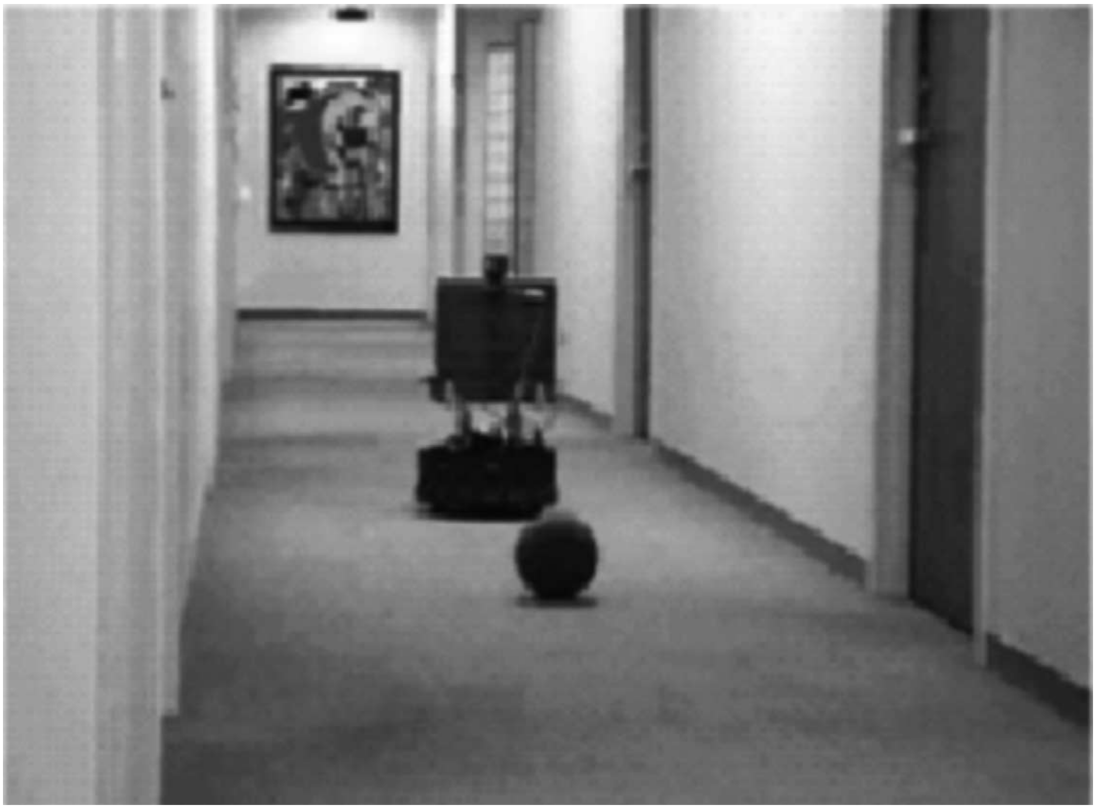


Fig. 6. One member of the team locating the target in the find object task.

search techniques such as foraging, the systematic search guarantees that each landmark is searched at most once and that all landmarks are guaranteed to be searched, if necessary. Using a greedy algorithm for landmark selection, the team was consistently able to find the landmark within 30 s provided that there were no catastrophic failures of the place recognition system. On typical runs, the team found the object in approximately 20 s.

For the town crier task, team members were again started from a central point at the extreme east end of the wing. The objective was for the robots to go through each landmark at least once, making the announcement at each landmark that the robots passed through. If a robot had already spoken at a particular landmark, then no further announcement should be made there, since we do not wish to inundate any nearby offices with multiple announcements. Again, barring any catastrophic failures of the place recognition system, the team was able to complete the task successfully.

The place recognition system is the weak point of the current implementation. Minor errors are common and occasional catastrophic failures can cause one of the team members to think that it has traversed its intended destination when in fact it has not. While we are working on improving the place recognition system, it should be stressed that the actual control and coordination architecture worked without error.

7. Conclusions

Grounding inference is a complex but unavoidable issue for systems embodied physically. Traditional symbolic reasoning systems face the issue of maintaining a world model that is coherent with the dynamic world. This issue is exacerbated in multi-robot systems, as we now have *n* knowledge-bases to synchronize with each other as well as the external environment. The multi-robot case shares some similarity to the problem of replicating *n* distributed databases; a problem that others have shown to very challenging, since the number of conflicts during attempted updates rises with the third power of the number of participating robots. We offer an alternative architecture that supports the useful features of a traditional symbolic reasoning system, in particular the ability to utilize predicate/argument structure, while avoiding the model coherence and database synchronization issues that traditional symbolic and tiered systems encounter.

The HIVEMind architecture allows behavior-based systems to abstract over both objects and sensors, while providing an anchoring approach that is efficient enough in both inference speed and bandwidth consumption to be usable on physical robotic teams. It presents multi-robot system designers with more powerful representations than behavior-based systems, and has a simple, efficient model for group coordination that consumes very little bandwidth while allowing team members to react to opportunities or contingencies within O(1) time. We believe that the right set of representational choices can allow the kinds of inference presently implemented on robots to be cleanly grounded in sensor data and reactively updated by a parallel inference network. By continually sharing perceptual knowledge between robots, coordination can be achieved for little or no additional cost beyond the communication bandwidth required to share the data. The effect is a kind of “group mind” in which robots can treat one another as auxiliary sensors and effectors. We have currently implemented two tasks utilizing HIVEMind: one that finds object, and another that makes announcements. Our current goal is to implement a system which finds and traps evading targets such as other robots or other humans. This is an especially interesting task because it requires non-trivial spatial reasoning about containment and visibility.

References

[1] R.C. Arkin, Behavior-Based Robotics, MIT Press, Cambridge, MA, 1998.

[2] R.C. Arkin, T.R. Balch, Aura: principles and practice in review, Journal of Experimental and Theoretical Artificial Intelligence 9 (2) (1997) 175–189.

[3] T.R. Balch, R.C. Arkin, Behavior-based formation control for multirobot teams, IEEE Transactions on Robotics and Automation 14 (6) (1998) 926–939.

[4] P. Bonasso, R.J. Firby, E. Gat, D. Kortenkamp, in: Hexmoor, Horswill, Kortenkamp (Eds.), Experiences with an Architecture for Intelligent Reactive Agents, Taylor & Francis, London, Journal of Theoretical and Experimental Artificial Intelligence 9 (1997) 2–3 (Special Issue on Software Architectures for Physical Agents).

[5] P.R. Cohen, H.J. Levesque, Teamwork, Nous 25 (4) (1991) 487–512 (Special Issue on Cognitive Science and AI).

[6] J.H. Connell, SSS: a hybrid architecture applied to robot navigation, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 92), Nice, France, 1992, IEEE Press, New York, pp. 2719–2724.

[7] S. Coradeschi, A. Saffiotti, Anchoring symbols to sensor data: preliminary report, in: Proceedings of the 17th AAAI Conference, Austin, TX, July 2000, pp. 129–135.

[8] S. Coradeschi, A. Saffiotti, Perceptual anchoring of symbols for action, in: Proceedings of the 17th IJCAI Conference, Seattle, WA, 2001, pp. 407–412.

[9] D. Eckhardt, P. Steenkiste, Measurement and analysis of the error characteristics of an in-building wireless network, in: Proceedings of the ACM SIGCOMM'96, August 1996, pp. 243–254.

[10] D. Goldberg, M.J. Mataric, Robust Behavior-Based Control for Distributed Multi-Robot Collection Tasks, USC Institute for Robotics and Intelligent Systems, Technical Report IRIS-00-387, 2000.

[11] J. Gray, P. Helland, P. O'Neil, D. Shasha, The Dangers of Replication and a Solution, Sigmod, 1996.

[12] J. Gray, A. Reuter, Transaction Processing: Concepts and Techniques, Morgan Kaufmann, San Francisco, CA, 1993.

[13] I. Horswill, Grounding mundane inference in perception, Autonomous Robots 5 (1998) 63–77.

[14] I. Horswill, Functional programming of behavior-based systems, in: Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, 1999.

[15] I. Horswill, R. Zubek, A. Khoo, C. Le, S. Nicholson, The Cerebus project, in: Proceedings of the AAAI Fall Symposium on Parallel Cognition and Embodied Agents, 2000.

[16] I. Horswill, Tagged behavior-based architectures: integrating cognition with embodied activity, in: Proceedings of the IEEE Intelligent Systems, September/October 2001, IEEE Computer Society, New York, pp. 30–38.

[17] C. Ronald Kube, H. Zhang, Collective robotic intelligence, in: Proceedings of the Second International Conference on Simulation of Adaptive Behavior, December 7–11, 1992, pp. 460–468.

[18] P. Maes, Situated agents can have goals, Robotics and Autonomous Systems 6 (1990) 49–70.

[19] L.E. Parker, ALLIANCE: an architecture for fault tolerant multirobot cooperation, IEEE Transactions on Robotics and Automation 14 (2) (1998).

[20] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice-Hall, Englewood Cliffs, NJ, 1995.

[21] R. Simmons, S. Singh, D. Hershberger, J. Ramos, T. Smith, First results in the coordination of heterogeneous robots for large-scale assembly, in: D. Rus, S. Singh (Eds.), Experimental Robotics VII (ISER, 2000), Lecture Notes in Control and Information Sciences 271, pp. 323–332, Springer, 2001.

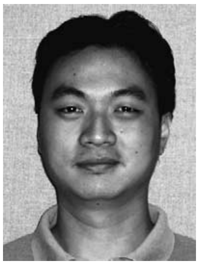
[22] L. Steels, Synthesising the origins of language and meaning using co-evolution, self-organisation and level formation, in: J.R. Hurford, M. Studdert-Kennedy, C. Knight (Eds.), Approaches to the Evolution of Language, Cambridge University Press, Cambridge, 1998, pp. 384–404.

[23] M. Tambe, Teamwork in real-world, dynamic environments, in: Proceedings of the Second International Conference on Multi-Agent Systems (ICMAS-96), Menlo Park, CA, December 1996, AAAI Press, pp. 361–368.

[24] M. Veloso, M. Bowling, S. Achim, K. Han, P. Stone, The CMUnited-98 champion small robot team, M. Asada, H. Kitano (Eds.), RoboCup-98: Robot Soccer World Cup II, Springer, Berlin, 1999, pp. 891–897.

[25] M. Van Wie, A probabilistic method for team plan formation without communication, Agents (2000) 112–113.

[26] G. Xylomenos, G.C. Polyzos, Internet protocol performance over networks with wireless links, IEEE Network 13 (1999) 55–63.



Aaron Khoo is a Ph.D. candidate at Northwestern University, where he is a member of the Autonomous Mobile Robotics Group. His primary research interests lie in the realm of multi-robot systems and human–computer interaction. He received a BA in computer science and mathematics from Knox College.

Ian Horswill is an Associate Professor of computer science at Northwestern University. His research focuses on integrating high-level reasoning systems with low-level sensory-motor systems on autonomous robots. He received his B.Sc. from the University of Minnesota and his Ph.D. in computer science from the Massachusetts Institute of Technology.

ELSEVIER

Robotics and Autonomous Systems 43 (2003) 133–147

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Multi-modal anchoring for human–robot interaction

J. Fritsch*, M. Kleinehagenbrock, S. Lang, T. Plötz, G.A. Fink, G. Sagerer

Applied Computer Science, Faculty of Technology, Bielefeld University, P.O. Box 100131, 33501 Bielefeld, Germany

Abstract

This paper presents a hybrid approach for tracking humans with a mobile robot that integrates face and leg detection results extracted from image and laser range data, respectively. The different percepts are linked to their symbolic counterparts *legs* and *face* by anchors as defined by Coradeschi and Saffiotti [Anchoring symbols to sensor data: preliminary report, in: Proceedings of the Conference of the American Association for Artificial Intelligence, 2000, pp. 129–135]. In order to anchor the composite object *person* we extend the anchoring framework to combine different component anchors belonging to the same person. This allows to deal with perceptual algorithms having different spatio-temporal properties and provides a structured way for integrating anchor data from multiple modalities. An evaluation demonstrates the performance of our approach.

Keywords: Anchoring; Multi-modal person tracking; Human–robot interaction

1. Introduction

The increasing availability of mobile robot platforms with good navigation capabilities provides a basis for the exploration of advanced human–robot interfaces (HRI). The development of systems with natural HRI is an important prerequisite for the widespread use of robots in home and office environments [1]. However, building powerful interfaces that go beyond a simple dialog-based interaction between user and robot is challenging. Due to the nature of mobile systems it is necessary to use sensor devices that can be carried on-board a small robot for realizing an HRI. Additionally, the sensing techniques must be non-intrusive, i.e. a human must be allowed to interact with the robot without having to wear special equipment (e.g. markers, colored gloves) to enable the robot's sensors to observe him. Standard multimedia cameras are cheap sensors that can be used for observing a human instructor to track his position and recognize gestural instructions [3,22]. However, despite intensive research in computer vision, the variations in lighting conditions encountered in dynamic environments pose major problems for tracking humans based on their visual appearance. For example, the color of a human face changes significantly if the lighting conditions are varied. A face detection process based on color may therefore fail to always detect the face in the images of a sequence depicting a human moving through an office. At the same time there may be background objects entering the field of view of the camera that have a face-like color. Consequently, the feature sequence belonging to an image sequence may contain false positives (background objects) and false negatives (missed faces).

In order to enable the robot to track humans over time despite inaccuracies in the feature sequence, the tracking algorithm can make use of temporal information and context knowledge. These sources of

[*] Corresponding author. Fax: +49-521-106-2992.
E-mail address: jannik@techfak.uni-bielefeld.de (J. Fritsch).

0921-8890/03/$ – see front matter © 2003 Elsevier Science B.V. All rights reserved.
PII: S0921-8890(02)00355-X

information allow to (1) select the features matching an internal symbolic description of the object to be tracked, and (2) focus processing on a subset of all features. The latter is especially important if the sensor capability is limited, the processing power is small, or several objects are present.

The *anchoring* framework by Coradeschi and Saffiotti [4,5] aims at providing a method for tracking objects over time by defining a theoretical basis for grounding symbols to percepts originating from physical objects. The practical capability is demonstrated with examples dealing with a single type of percept obtained by processing camera images.

However, in complex environments several different sensors can generate different types of percepts originating from the same physical object. Additionally, the spatio-temporal properties of the different types of percepts can vary significantly. We propose a solution to these problems by anchoring a symbol denoting a *composite object* through anchoring the symbols of its corresponding *component objects*. In this solution, the composite anchoring module is responsible for fusing the data of the component anchors. Our approach to integrate several anchoring processes can be easily extended to other modalities and allows for parallel or distributed anchoring of component symbols. To demonstrate our approach we perform person tracking by anchoring the symbol *person* through anchoring the symbols *legs* and *face* to the corresponding percepts.

In extension to the original use of anchoring for connecting one symbol system to one perceptual system, our application concentrates on solving the challenging task of tracking composite objects, i.e. humans. Therefore, we use a symbol system that only contains predicate symbols describing the identity of persons to be tracked. The use of more predicate symbols in the symbol system to support more complex interactions using, for example, speech (e.g. ‘Follow the small person with the red shirt’) will be the focus of future work.

In the following section we will give a review of related work. The original anchoring framework will be described in Section 3. The basic idea of the proposed integration framework is presented in Section 4, while Section 5 describes some extensions to cope with multiple composite objects. The application to person tracking is described in Section 6. Section 7 presents an extensive evaluation of the complete system. The article concludes with a summary of the presented work.

2. Related Work

Our approach extends work by Coradeschi and Saffiotti [4,5], and therefore their anchoring framework is described in detail in Section 3. In this section we will concentrate on the related techniques of data association and fusion, as these techniques bear similarities to our approach.

Bar-Shalom and Li discuss in [2, Ch. 8.2] different types of configurations for multisensor tracking including a hybrid approach. The so-called Type I configuration denotes a standard single sensor tracking system. Type II configurations perform Type I tracking for several sensors and subsequently fuse the individual tracks, while Type III proposes a direct *synchronous* sensor data fusion across multiple sensors before performing tracking on the fused sensor data. The Type IV configuration, instead, uses local data association for the individual sensors but a global tracking. However, this configuration still requires synchronous sensor data. For fusing data originating from sensors at different sites, a hierarchical hybrid configuration for multisensor–multisite tracking is proposed.

For person tracking using different sensing modalities a variety of approaches and fusion methods have been developed. Darrell et al. [6] use a Type II data fusion method to integrate depth information, color segmentation, and face detection results. Fusing the individual tracks is done using simple rules. Likewise, Okuno et al. [11] use a Type II configuration to fuse auditory and visual information from talking persons. Track fusion is done rule-based, but differently from [6] thresholds on the track differences are used to avoid fusing different tracks. A Type III configuration is used by Feyrer and Zell [7] to track persons based on vision and laser range data. The two types of sensor data are fused by adding a two-dimensional Gaussian to a potential field representation for each potential person position. After initial selection of the person to be tracked, another Gaussian is added to the potential field at the Kalman filtered position estimate to maintain temporal coherence. Type IV configurations with sequential processing of the individual sensors are often implemented hierarchically. After associating coarse position estimates, a smaller search space is used for processing more precise sensor data. Schlegel et al. [15] propose vision-based person tracking that uses color information to restrict the image area that is processed to find the contour of a human. A more sophisticated method to realize the sequential search space reduction is proposed by Vermaak et al. [21]. In their approach sound and vision data are sequentially fused using particle filtering techniques.

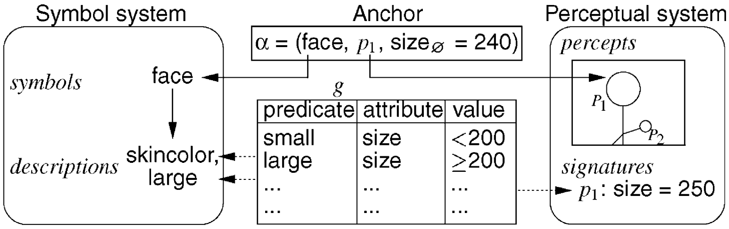
Although we perform person tracking using a camera and a laser range finder which are on-board a mobile robot, we have to perform multisite tracking in a hybrid configuration, as different components of a human are observed from different positions. In contrast to the intersite association and overall information fusion proposed in [2] we developed a model-based modular integration scheme that extends the anchoring framework described in Section 3. Besides enabling classical tracking with multiple sensors at different sites, anchoring allows to maintain representations for temporarily occluded objects and provides mechanisms for reacquiring the object. Therefore, anchoring can be understood as an extension to classical tracking approaches that defines a framework for dealing with missing sensor data in a structured way. The proposed multi-modal anchoring approach is easy to implement, has transparent structure, and exhibits efficient, low complexity performance.

3. Anchoring

The problem of recognizing objects by linking features extracted from sensor data to an internal symbolic representation is especially prominent in an autonomous system whose environment is constantly changing. Such a system needs to establish connections between processes that work on the level of abstract representations of objects in the world (symbolic level) and processes that are responsible for the physical observation of these objects (sensory level). These connections must be dynamic, since the same symbol must be connected to new sensor data every time a new observation of the corresponding object is acquired.

We follow the definition of anchoring proposed by Coradeschi and Saffiotti [5]. They define anchoring as the problem of creating and maintaining in time the correspondence between symbols and sensor data that refer to the same physical object. Basically anchoring incorporates a *symbol system* and a *perceptual system* that are linked by an anchor (Fig. 1). The symbol system includes a set of individual symbols and a set of unary predicate symbols. Each individual symbol has a symbolic description which is a set of predicate symbols. The perceptual system includes a set of *percepts* and a set of *attributes*. A percept is a structured collection of measurements assumed to originate from the same physical object. An attribute is a measurable property of a percept. The set of attribute-value pairs of a percept is called the *perceptual signature*.

The role of anchoring is to establish a correspondence between a symbol, which is used to denote an object in the symbol system, and a percept generated in the perceptual system by the same object. This is achieved by comparing the symbolic description and the perceptual signature via a predicate grounding relation g. This relation constitutes the correspondence between unary predicates and values of measurable attributes. For example, g could specify that a symbol with the predicate *large* corresponds to a percept, if the value of its attribute *size* is above a certain threshold. The relation g can be embedded in a function *match* that evaluates whether a given perceptual signature is consistent to a given symbolic description or not. The



predicate	attribute	value
small	size	<200
large	size	≥200
...	...	...
...	...	...

Fig. 1. Linking symbols to sensory data with anchors.

correspondence between symbol and percept is represented in an internal data structure α, called anchor. Since new percepts are generated continuously within the perceptual system, this relation is indexed by time.

At every moment t, the anchor $\alpha(t)$ contains three elements: a symbol, meant to denote an object inside the symbol system; a percept, generated inside the perceptual system by observing an object; a signature, providing the estimate of the values corresponding to the observable properties of the object. The anchor α is *grounded* at time t, if it contains the percept perceived at t and the updated signature. If the object is not observable at t and so the anchor is *ungrounded*, then no percept is stored in the anchor but the signature still provides the best available estimate.

In order to solve the anchoring problem for a given symbol x in a dynamic environment three main functionalities have been outlined in [4,5]:

- *Find*. Create a grounded anchor the first time that the object denoted by x is perceived. The function *match* is used to assure that the symbolic description matches the perceptual signature. In case of multiple matching percepts, a *selection* can either be made inside the find functionality or by the symbol system.
- *Track*. Continuously update the anchor while observing the object. In this case the prediction is achieved by a specific *one-step-predict* function. The predicted signature is compared to the perceived attributes with a *match-signature* function. This allows to find percepts compatible with the attributes of the percepts anchored to the symbol in the previous steps. In case of multiple matching percepts, the *select* function is used to choose one percept.
- *Reacquire*. Update the anchor when the object has to be reacquired, i.e. if the anchor is ungrounded. This is used to locate an object when there is a previous perceptual experience of it. The experience is used to *predict* a new signature which is then compared to newly acquired percepts. Here, the prediction is generally more complex than in the *track* case. If it is *verified* by using *match* that a percept is compatible with the prediction and the symbolic description then the current signature is *updated*. Again, in case of multiple matching percepts, a *select* function is used to choose one percept for updating.

For a detailed description of the formal anchoring framework the interested reader is referred to [4,5].

4. Multi-modal anchoring

Up to now the literature on anchoring considers only the special case of connecting one symbol to the percepts from one sensor. However, the real world contains objects that cannot be captured completely by a single sensor. If several sensors are used, the symbolic description of the object has to be linked to several different types of percepts acquired from different modalities.

One solution is the extension of the anchoring definition to link several percepts to a single symbol. However, with such an approach the integration of different types of percepts with different processing times makes it necessary to anchor the individual percepts asynchronously. Additionally, if the different percepts relate to different parts of the object the spatial relations between them need to be incorporated into the predicate grounding relation to obtain a consistent result. Consequently, the resulting algorithm for this solution may become very complex from an implementational point of view.

Therefore, we propose a modular approach (Fig. 2) that allows to anchor a symbol of a composite object by distributed anchoring of the corresponding component objects based on the related percepts originating from multiple modalities. This modular approach provides a structured way for simple integration of

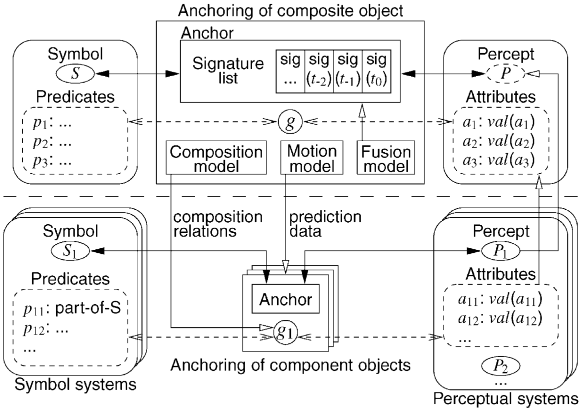


Fig. 2. Multi-modal anchoring.

additional component anchors and facilitates parallel anchoring of different types of percepts. The information provided by the individual perceptual systems of the component anchors is collected by a composite anchoring process for integration. The combined data is again stored in an anchor structure, the so-called composite anchor.

The main difference to original anchoring is that the symbol corresponding to the composite object has no direct perceptual counterpart. Every time a component anchoring process has chosen a new percept for updating its anchor, the percept is linked to the symbol of the composite object. The composite anchoring process then calculates its own perceptual signature by incorporating the signature of the component anchor. Usually, this signature can only be used to update a subset of the available attributes of the composite anchor, because the associated percept originates only from the perceptual system of a component object.

The main functionalities *find*, *track*, and *reacquire* defined in the original anchoring do not directly exist for the composite anchor module. These functions are carried out by the component anchoring processes that also initiate updates of the composite anchor. The composite object is anchored/grounded, if at least one component object is anchored/grounded. Because every component anchor module has different predicate symbols, it also contains its own predicate grounding relation. The predicate grounding relation of the composite anchor module embodies the correspondence between predicates concerning the composite object and attribute data calculated from attribute values originating from the different component anchoring processes.

In order to coordinate all component anchoring processes, it must be ensured that the different sensors observe the same composite object. The component anchoring processes have to be supplied with position estimates for the composite object, and the composite anchoring process has to fuse the information supplied by the component anchors. Therefore, a *composition model*, a *motion model*, and a *fusion model* are provided.

The composition model contains the spatial relationships between the composite object and its components. It ensures that the individual anchoring processes only select percepts that are compatible with the composite object. At startup, a component anchoring process establishes a grounded anchor simply if its symbolic description matches the perceptual signature. Hence, the composite anchor is initialized and from now on data about the composite object is provided to its component anchoring processes as follows: the *match* function of every component anchor is extended to additionally make sure that the composition relations provided by the composition model of the composite object are satisfied. Therefore, the predicate *part-of-S* is added to the symbolic description of the component anchors where *S* is the symbol of the corresponding composite object. After a component anchoring process has executed its extended *match*, the composite anchoring process can perform its own *match* to check whether its symbolic description matches the corresponding perceptual signature of the processed percept.

The motion model describes the motion behavior of the composite object and allows to predict its position. Together with the spatial relations provided by the composition model a component anchoring process can predict the position of its underlying component object. Especially for steerable sensors which allow to select the desired field of view it may be necessary to use information about the composite object. In this case a steerable sensor can be pointed into the direction where a percept is expected in order to establish the corresponding component anchor.

The fusion model is used for integrating the various signatures of the component anchors in the composite anchor. Every time a component anchoring process has processed new percepts, it sends its new signature to the composite anchor module. This signature refers to the point of time in the past when the corresponding sensor data was acquired. Since the different perceptual systems achieve different processing speeds, the composite anchor module does not always receive the attribute data from component anchors in correct temporal order. In order to ensure that the attribute data is fused to the signature of the composite anchor at the appropriate point of time, the composite anchoring process maintains a list containing all signatures sorted in chronological order. New attribute data is inserted in the list and the signature of the composite anchor is updated for the corresponding point of time based on the fusion model. If the list already contains entries that are newer than the inserted one, then the fusion of the signatures of the composite anchor is repeated for the subsequent points of time. The underlying specification of the fusion itself is domain dependent.

5. Anchoring multiple composite objects

Usually, more than one object has to be tracked simultaneously. Then, several anchoring processes have to be run in parallel to keep track of the different objects. In this case, multi-modal anchoring as described in the previous section may lead to the following conflicts between the individual composite anchoring processes:

- A percept is selected by more than one anchoring process.
- The anchoring processes try to control a steerable sensor contradictorily.

In order to resolve these two problems, a *supervising module* is introduced, which manages all composite anchoring processes. It coordinates the selection of percepts and schedules access to steerable sensors. The supervising module grants access to steerable sensors only to the composite anchoring process which holds the so-called *anchor of interest*. The decision which is the anchor of interest depends on the intended application.

In order to coordinate the selection of percepts the *select* functionalities of the individual component anchor modules have to be modified. These modules no longer select percepts individually. Instead, they assign to every percept a score, which is the higher the better a percept fits the anchor. Based on these scores an overall selection can be performed (Fig. 3). Any possible selection result can be expressed as a list of assignments, where the nth entry of the list contains the number of the percept which is selected for the nth anchor. Note, that the entries of the list have to be pairwise different in order to describe a consistent result. The total score of an overall selection is defined as the sum of the scores corresponding to the assignments. The aim is to find the optimal result, which is the selection yielding the maximum total score. The corresponding search is realized using a search tree: the root of the tree is given by the empty list, whose list entries are all undefined. For every successive node the number of undefined entries decreases by 1. The leaves of the search tree contain all possible overall selections. Since the maximum of all scores assigned by an anchor is known, the total score of a partially undefined list can be estimated optimistically. Hence, the search can be efficiently realized using the A*-algorithm.

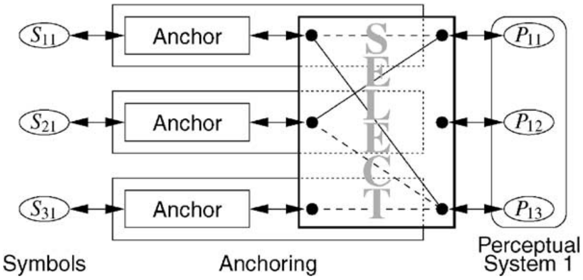


Fig. 3. Modification of *select* in component anchor modules.

However, the number of percepts not necessarily coincides with the number of anchors. If there are more anchors than percepts, not every anchor is assigned a percept and therefore is not updated. If there are more percepts than anchors, not every percept is assigned to an anchor. The remaining percepts are used to establish new anchors. Additionally, an anchor that was not updated for a certain period of time will be removed by the supervising module.

6. Person tracking in a dynamic environment

In order to prove the feasibility of our multi-modal anchoring approach, we demonstrate its use for person tracking with a mobile robot. Person tracking is a prerequisite for every HRI and has to be realized with the available on-board sensors which often can capture only a part of the human body due to the usually small distance between the human and the robot. Our robot can observe a person with a camera and a laser range finder. Based on the skin-colored regions extracted from camera images the face of a person can be detected and identified. The beam from the laser range finder is at leg-height and, consequently, human legs can be detected. In this section we will first present our mobile system. Then, the algorithms to extract the leg and face percepts will be described. Finally, component anchoring and anchoring of the composite object person is explained.

Our hardware platform (Fig. 4) is a PeopleBot from ActivMedia with two on-board PCs (Pentium III, 850

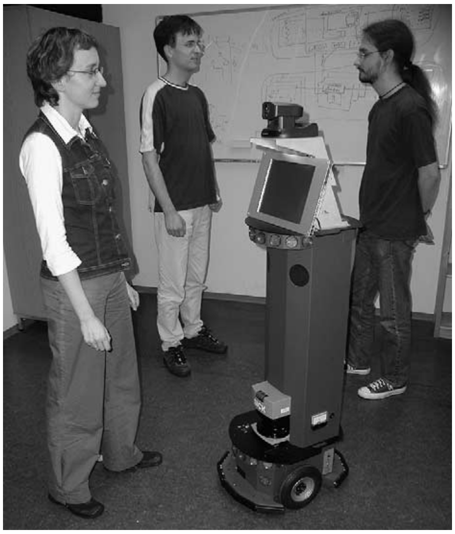


Fig. 4. Our PeopleBot following a person.

and 500 MHz, respectively). The first PC is used for controlling the motors and the on-board sensors while the second one is used for image processing. Both PCs run Linux and are linked with a 100 Mb Ethernet. A SICK laser range finder is mounted at the front at a height of approximately 30 cm. Measurements are taken in a horizontal plane, covering a 180° field of view. A pan-tilt color camera (Sony EVI-D31) is mounted on top of the robot at a height of 140 cm for acquiring images of the upper body part of humans interacting with the robot. For robot navigation we use the ISR (Intelligent Service Robot) control software developed at the Center for Autonomous Systems, KTH, Stockholm [10].

6.1. Detection of human pairs of legs in 2D laser scans

In mobile robotics 2D laser range finders are often used, primarily for robot localization and obstacle avoidance. A laser range finder mounted at the height of legs can also be applied to detect persons. Fig. 5 shows a sample laser scan with a person standing in front of the robot. The legs result in a characteristic pattern.

Detecting legs in laser scans was already considered for mobile systems. In [16] for every object features like diameter, shape, and distance are extracted from the laser scan. Then, fuzzy logic is used to determine which of the objects are pairs of legs. In [17] local minima in the range profile are considered to be pairs of legs. Since other objects (e.g. trash bins) produce patterns similar to persons, additionally moving objects are distinguished from static objects.

Our approach for the detection of human legs is based on laser scans with an angular resolution of 0.5°. Generally, persons can be located by two closely positioned segments. A segment within a laser scan consists of consecutive reading points with similar distance values, which usually result from a smooth

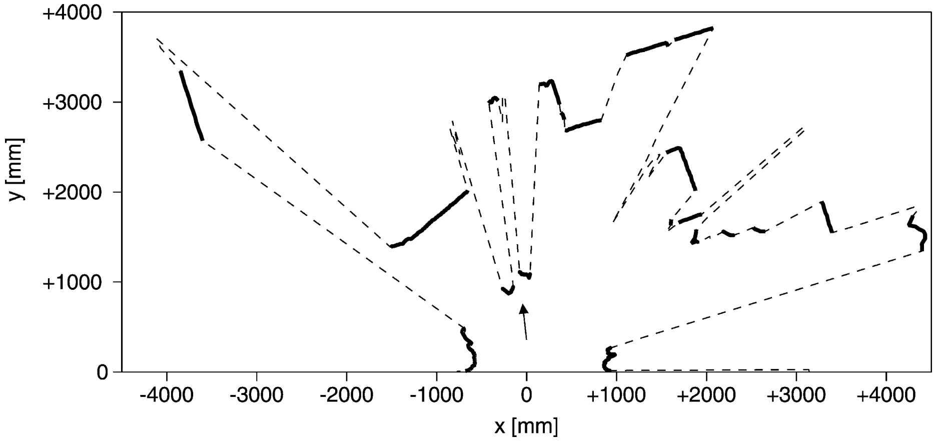


Fig. 5. A sample 2D laser scan. The arrow marks a pair of legs.

surface of a single object. Large differences of distance values are due to edges or occlusions. Thus, single human legs are mostly observed as single segments.

The detection of pairs of legs consists of three steps: *segmentation*, *classification*, and *grouping*. In the first step the laser scan is split into segments. Each segment consists of a maximum number of successive reading points, where the differences of the distance values of two consecutive points are below a given threshold (chosen as 75 mm). In the following step each segment is classified as *leg* or *non-leg*, based on the following features: number of reading points (n), mean distance (μ), standard deviation of the distances (σ), width in world coordinates in a direction perpendicular to the laser beam (w), and distances to the adjacent segments $(d_1$ and $d_2)$. We obtained satisfying results using the following conditions to classify a segment as *leg*:

$$\begin{aligned}&(n > 4) \wedge (\mu < 3000\,\text{mm}) \wedge (\sigma < 40\,\text{mm})\\&\quad \wedge (50\,\text{mm} < w < 250\,\text{mm})\\&\quad \wedge (\max(d_1, d_2) > 250\,\text{mm})\\&\quad \wedge (\min(d_1, d_2) > -50\,\text{mm}).\end{aligned}$$

In the final step single legs are grouped into pairs depending on their distance in world coordinates, which is chosen to be below 500 mm.

In certain cases one leg of a person is occluded by the other one, and thus only a single leg will be detected. In order not to discard this information, the *percepts* generated by this perceptual subsystem include all detected pairs of legs, and all single legs which are not part of a pair. The *attributes* computed for a percept are the direction given in the local coordinate system of the robot, the distance, and a flag, which indicates whether the percept is a pair of legs or not. The arrow in Fig. 5 marks a pair of legs detected with our approach in the sample laser scan.

6.2. Detection of human faces in color images

Face detection is very important for human–robot interaction: at first, the detection of a face is a reliable indicator for the presence of a person. In addition, much information is extractable from a face, e.g. person identity or gaze direction.

The perceptual subsystem that performs face detection processes color images from the pan-tilt camera mounted on top of the robot. The detection is modeled as an image scanning process, that repeatedly extracts sub-images for classification. To speed up the scanning process, the search space in the image is restricted to regions of skin color. Since we are dealing with images obtained from a camera on a mobile robot, the task of color segmentation is challenging:

- A moving robot encounters lighting conditions of high variability.
- There is no constant background in images as the robot acts in an unstructured environment.

In order to detect color regions under varying lighting conditions, an adaptive color segmentation algorithm has to be used. Probably the most famous adaptive image segmentation system is the Pfinder (person finder) system [23] for tracking a single, completely visible human wearing homogeneously colored clothes in front of a static background. In Pfinder, the color of every background pixel and each body part (head, torso, arms, hands, legs, and feet) is modeled as a Gaussian in YUV color space. Additionally, the positions of body parts are described by Gaussians in image coordinates.

For the task of skin color segmentation the related LAFTER system [12] uses similar techniques to track the face of a single user with a pan-tilt camera. Here a Gaussian mixture is used to model the background variations. In order to detect a face in arbitrary backgrounds captured by a moving camera, recent approaches avoid explicit background modeling [13,18]. However, these approaches are limited to single faces.

Different from the approaches above our goal is the tracking of *several* skin-colored image regions that may be subjected to different lighting conditions. This is realized by modeling every skin-colored region with a separate Gaussian distribution. In order to stabilize the adaptation step, we use context information from face detection to restrict updating to regions containing faces and select image areas of face size for adapting the color models. In the following we give a short overview of our adaptive skin color segmentation approach, more detailed information can be found in [8]. Note, that for skin color segmentation on the mobile robot no region-of-interest (ROI) is used and the complete image is segmented as the uncertainty for determining ROIs on a mobile robot is too high to be reliable enough.

For color representation the r–g color space is used as it is well suited for representing skin color over a wide range of lighting conditions [24]. For the special case of modeling a person's face a Gaussian distribution has been shown to be sufficient [14]. For every pixel the skin probability is calculated as the maximum of the individual probabilities of the Gaussian models. The resulting skin probability image is binarized with an empirically determined threshold of 0.2 and a connected components analysis yields skin-colored regions.

In order to prevent the color models from adapting to skin-colored background objects a face verification step is carried out on all regions found. For face detection we apply the *eigenface method*, operating on gray-level images. Any image with a size of $n \times m$ pixel can be considered as a point in an nm-dimensional space. Faces lie in a subspace of the overall image space. Kirby and Sirovich [9] demonstrated how Principal Component Analysis (PCA) can be used to efficiently represent human faces. Later, Turk and Pentland [20] applied this technique to face detection. PCA finds the principle components of the distribution of the face images, which are called *eigenfaces*. They span a subspace (face space) representing possible face images. We use a face space computed from a set of sample face images having a size of 37 × 43 pixel. These samples only contain the central parts of faces (eyes, nose and mouth) so that variances of the background are excluded. In addition, the images are preprocessed using histogram equalization in order to compensate varying lighting conditions.

Before a given image can be classified it has to be rescaled to the size of the sample images and preprocessed. The resulting image is then reconstructed by a weighted sum of eigenfaces. The resulting residual error is small if the given image is a face image and large otherwise. Hence, for classification an empirically determined threshold can be used to distinguish face from non-face images.

In order to decide whether a segmented region of skin color originates from a face, a sub-image at the position of the region has to be extracted and classified with the eigenface method. However, the center of mass (COM) of the region does not necessarily coincides with the center of the face due to inaccuracy of segmentation. Therefore, the area at the region has to be scanned at different positions and with varying scalings by using the following method: the center of the initial sub-image (x, y) coincides with the COM of the skin-colored region. There and at the two neighboring positions $(x+1, y)$ and $(x, y+1)$ the corresponding reconstruction errors for the extracted sub-images are computed. The next position of the scanning process is chosen according to *steepest gradient descent*. This process stops if a face is detected or a local minimum is reached. In the latter case the process continues with sub-images of a new size (±7.5%, followed by ±15%).

For all image regions that are found to contain a face updating of the color model is performed. In order to stabilize the updating process an empirically determined global skin color distribution is used for filtering out non-skin pixels. Based on a theoretical model Störring et al. [19] have shown that the overall skin color distribution occupies a shell-shaped area in r–g color space that is called *skin locus*. Similar to Soriano et al. [18] we determined the skin locus for our camera empirically with hand-segmented training images [8]. With all pixels in an elliptical image area at a detected face position that lie inside the skin locus, local Gaussian parameters are calculated and used to smoothly update the Gaussian model:

$$\vec{\mu}_{\mathrm{new}} = \gamma\vec{\mu}_{\mathrm{local}} + (1-\gamma)\vec{\mu}_{\mathrm{old}},$$
$$\mathbf{\Sigma}_{\mathrm{new}} = \gamma\mathbf{\Sigma}_{\mathrm{local}} + (1-\gamma)\mathbf{\Sigma}_{\mathrm{old}}.$$

For our system running at approximately 3 Hz a learning rate of $\gamma = 0.6$ has been shown to provide good results for persons moving in a standard office domain.

The *percepts* generated by this perceptual subsystem are the skin-colored regions classified as face. For every percept a set of *attributes* is computed: with the position information from the pan-tilt camera the angle of the face relative to the robot is calculated. The detected face size is used to estimate the distance d of the person: assuming that sizes of heads of adult humans only vary to a minor degree, the distance is proportional to the reciprocal of the size. The height of the face above the ground is also extracted by using the distance and the camera position.

Additionally, a face identification step is performed with a slightly enhanced version of the method proposed in [20]. Each individual is represented in face space by a mixture of several Gaussians with diagonal covariances. Practical experiments have shown that the use of 4–6 Gaussians leads to satisfying results in discrimination accuracy requiring only small amounts of training material. The mixture densities are estimated from the projections of up to 50 sample images per individual. The performance of the identification process has been evaluated in an experiment with nine individuals. For a test set of 76 images a recognition rate of 89% could be achieved. When accepting a rejection rate of 20%, over 96% of the images classified were assigned to the correct individual.

6.3. Anchoring component objects

The characteristics of the anchoring processes for the components legs and face are reflected in their three main functionalities. The *find* functions of the leg and face anchor modules anchor only percepts in front of the robot, if their distance to the robot is less than 3 m. Additionally, the selected leg percept must match the predicate *is-pair*. If the face anchor module is linked to the anchor of interest, it is first checked in the *find* function whether the field of view of the camera overlaps with the person position provided by the anchor of interest. If necessary, the camera is pointed to the direction where the face percept is expected. The functionalities *track* and *reacquire* of the anchor modules for legs and face are rather similar. All these functions try to anchor percepts close to the predicted position while considering restrictions given by the composition model of the person. More specifically, the *track* functions predict the current percept's position based on the last known position. In contrast, the prediction of the *reacquire* functions is based on the person position obtained from the person anchor module. If the face anchoring process tracks or tries to reacquire the face of the person of interest, the camera is steered to make sure that the position of the predicted percept does not move out of the field of view.

6.4. Anchoring composite objects

The person anchoring module receives individual signatures originating from the leg and the face anchoring processes. It is important to note that this data is processed asynchronously by the composite anchoring process. Fig. 6 shows the framework for anchoring the composite object *person*. The *composition model* used describes empirically defined person relations (Fig. 7).

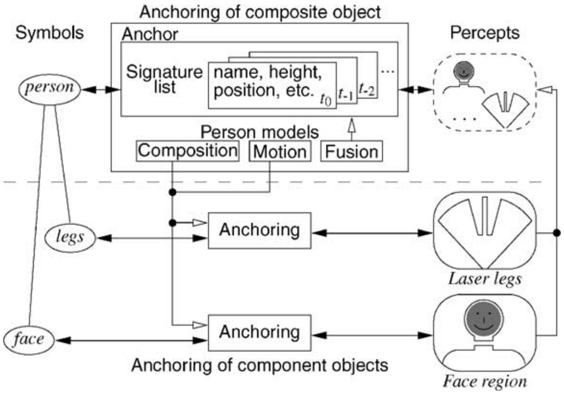


Fig. 6. Anchoring a person by anchoring the legs and the face.

All attributes of the multi-modal anchoring of persons that correspond to spatial positions are described by Gaussian distributions instead of scalar values. This allows to model uncertainty for positions. For the attributes of percepts the variance of the Gaussian can be determined from the measuring inaccuracy of the corresponding sensors. The *motion model* defines how a position can be predicted for time $t(i+1)$ based on the known position at time $t(i)$: the mean value remains unchanged (no velocity assumed) while the variance increases linearly with time, expressing increasing uncertainty.

The attribute values contained in the signature list of the composite anchor module are updated by multiplying the Gaussian of each attribute value with the Gaussian representation of the corresponding attribute values from new percepts. This results in the following update formulas that are calculated in the *fusion*

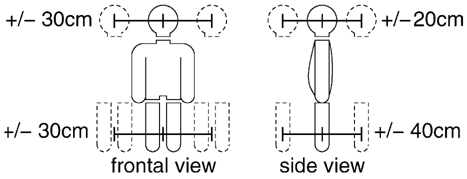


Fig. 7. The composition model for matching consistent percepts.

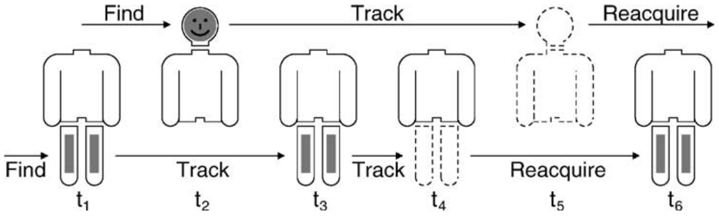


Fig. 8. A schematic example for anchoring a person.

model:

$$\mu_{t(i)} = \frac{\mu_{t(i-1)}\sigma_{\mathrm{p}} + \mu_{\mathrm{p}}\sigma_{t(i-1)}}{\sigma_{t(i-1)} + \sigma_{\mathrm{p}}},$$

$$\sigma_{t(i)} = \frac{\sigma_{t(i-1)}\sigma_{\mathrm{p}}}{\sigma_{t(i-1)} + \sigma_{\mathrm{p}}}.$$

The resulting mean value $\mu_{t(i)}$ is a weighted sum of the old mean value $\mu_{t(i-1)}$ and the mean value of the percept μ_{p}. Since the weights are given by the variances of the old position in the signature list and the percept, the mean value corresponding to the smaller variance (more certainty) has a greater effect.

The person attribute values that are updated with the signatures of the grounded component anchors are the angle ϕ_{p} and distance d_{p} relative to the robot, the face height h_{p} and the person name N_{p}. The initialization of the values ϕ_{p} and d_{p} is carried out if a component anchor is grounded for the first time. The attribute values h_{p} and N_{p} can only be initialized after receiving the first signature from the face anchoring process. During normal operation the person's fusion model makes sure that the person's position is smoothly updated by anchored legs and faces. In contrast, h_{p} and N_{p} can only be updated by processing face signatures.

In order to illustrate the concept a schematic example for anchoring one person is shown in Fig. 8 depicting six consecutive time steps at the beginning of an anchoring process:

t_1 : Person anchoring is started and all component anchoring processes perform their *find*. The leg detection generates a leg percept and the legs are anchored for the first time. The leg anchoring process switches from *find* to *track*. Subsequently, the person position contained in the composite anchor module is initialized and the person becomes grounded. Now, the *find* of the face anchor module is able to point the camera into the right direction.

t_2 : The face detection generates a face percept and the face anchor becomes grounded. The face anchoring process switches from *find* to *track* and the person anchor is updated accordingly.

t_3 : Again, the leg detection generates a leg percept. Based on the *track* function, the leg anchor as well as the person anchor are updated.

t_4 : In this time step, new laser range data is processed but no matching leg percept is found by the leg anchoring process. Therefore, it switches from *track* to *reacquire*. No updating of the person anchor takes place.

t_5 : A new camera image is processed but no face percept matching the prediction of the person position is found. Thus, the face anchoring process also switches from *track* to *reacquire*. Now the person is ungrounded since neither the legs nor the face are grounded.

t_6 : In the new laser range data a leg percept matching the predicted person position is found. Now the legs as well as the person are grounded again.

7. Results

We implemented the extended anchoring framework in an object-oriented manner using C++ and added the person tracking functionality to the ISR software [10] on the behavior level. When the robot is instructed to track persons the tracking behavior is started in parallel with other behaviors necessary, for example, obstacle avoidance. The tracking behavior initializes the person anchoring process.

The evaluation of our system was carried out in an office room, more specifically in an area having a size of approximately 4.60 m × 3.40 m. The room was equipped with wooden furniture, which was

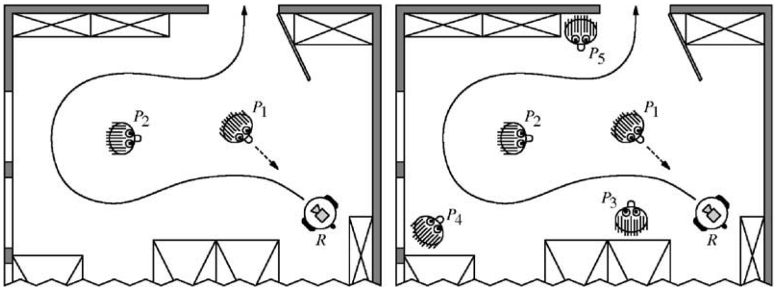


Fig. 9. Scenario: first setup (left); second setup (right).

challenging for the face recognition, because the color of wood is similar to skin color. We realized two setups (Fig. 9). In the first setup only two persons were present, one in the middle of the room standing still (P_2) and one guiding the robot (P_1). The task for P_1 was to become the person of interest by approaching the robot (<1 m). Then, P_1 had to guide the robot around P_2 and to leave the room through the door, while looking towards the camera as long as possible. The resulting trajectory had a length of approximately 7.5 m. The second setup was similar to the first one, but three additional persons P_3–P_5 were placed at predetermined locations in the room, not affecting the trajectory resulting from the first setup. P_1 was instructed to try to regain the attention of the robot in case that the robot lost P_1. If this was not possible, because the robot tried to follow one of the other persons, then the experiment was interpreted as failure. Both experiments were carried out with ten different subjects.

Throughout the tests, the laser range finder provided new laser range data at a rate of 4.6 Hz to the leg detection algorithm. The processing time necessary for generating leg percepts and anchoring was negligible. The adaptive skin-color segmentation processed images with a size of 189 × 139 pixels. For each skin-colored region the face detection was carried out. The processing time of the face detection and identification system depends on the number of skin-colored regions present in the image. On average the face percepts were provided at a rate of 3.1 Hz. Again, the time necessary for updating component and composite anchor was negligible. Together, the person attributes were updated with an average rate of 7.7 Hz due to the asynchronous anchoring of the different types of percepts.

The first setup (Table 1) was accomplished after an average time of 55 s. The robot lost three people once, but they were able to regain the attention of the robot to complete the run. On average 95.3% of the time a person was grounded. The legs were grounded 92.1%

Table 1
Results of the first setup with P_1 and P_2

Run	t (s)	$v_{ø}$ (m/s)	Lost	Person grounded (%)	Legs grounded (%)	Face grounded (%)	Legs/step	Face/step
1	39	0.19	0	99.7	98.9	63.1	1.78	0.76
2	62	0.12	0	96.6	93.8	36.4	1.71	0.90
3	52	0.14	1	95.4	83.5	51.0	1.72	0.57
4	56	0.13	0	99.3	93.8	54.1	1.79	0.59
5	81	0.09	1	96.4	95.7	34.7	1.63	0.40
6	32	0.23	0	99.2	98.7	51.1	1.87	0.78
7	90	0.08	1	80.9	73.2	22.0	1.94	0.49
8	51	0.15	0	99.2	98.8	56.5	1.75	0.70
9	42	0.18	0	98.0	97.9	35.7	1.79	0.45
10	44	0.17	0	88.6	87.1	16.3	1.60	0.39
Average	55	0.14	–	95.3	92.1	42.1	1.76	0.60

Table 2
Results of the second setup with P_1–P_5

Run	t (s)	$v_ø$ (m/s)	Lost	Person grounded (%)	Legs grounded (%)	Face grounded (%)	Legs/step	Face/step
1	60	0.13	2	93.6	91.5	27.7	2.63	0.41
2	43	0.17	0	96.7	95.0	20.7	2.61	0.32
3	The robot lost P_1 and tried to follow P_3							
4	51	0.15	0	98.7	90.4	66.0	2.49	0.74
5	47	0.16	0	96.2	94.5	7.1	2.52	0.20
6	The robot lost P_1 and tried to follow P_2							
7	77	0.10	0	99.8	97.5	72.0	2.59	0.85
8	74	0.10	0	93.4	92.6	20.3	2.63	0.22
9	61	0.12	0	97.7	96.1	36.4	2.55	0.56
10	42	0.18	0	86.1	84.2	11.9	2.73	0.26
Average	57	0.13	–	95.3	92.7	32.8	2.59	0.45

of the time, the face 42.1%. On average 1.76 legs and 0.6 faces were processed in every computation step by the corresponding perceptual systems.

The time needed to successfully perform the task of the second, more complex setup (Table 2) took only 2 s more per run on average. For this setup we expected more percepts to be computed, because more persons were present. This was in fact true for the legs, but not for the face. The persons guiding the robot were taking care of not colliding with one of the persons P_2–P_5 and, therefore, looked at the camera less often. This resulted in a correspondingly lower face detection rate. On average the face was grounded only 32.8% of the time. The legs and the whole person were grounded for approximately the same time (95.3 and 92.7%) as in the first setup. Runs 3 and 6 resulted in a failure. A recovery was not possible even though the face identification would have indicated the mistake. This is because an active search for a specific person, which goes beyond the reacquire functionality of anchoring, is not part of the current implementation.

8. Summary

We presented a method for anchoring composite symbols through anchoring component symbols to their associated percepts and subsequently fusing the resulting data of the component anchors. This modular approach facilitates multi-modal anchoring and can easily be extended with additional anchoring processes. We demonstrated the performance of our approach with a person tracking application for a mobile robot. In the current implementation laser range data and color images are processed to find percepts for the symbols *legs* and *face*. Our extended anchoring framework allows for multi-modal tracking of humans. Through taking advantage of the different sensor capabilities in terms of precision and information content a more complete representation of tracked persons is maintained. Therefore, our approach forms the basis for more advanced human–robot interaction.

Acknowledgements

This work has been supported by the German Research Foundation within the Collaborative Research Center ‘Situated Artificial Communicators’ and the Graduate Programs ‘Task Oriented Communication’ and ‘Strategies and Optimization of Behavior’.

References

[1] A. Agah, Human interactions with intelligent systems: research taxonomy, Computers and Electrical Engineering 27 (1) (2000) 71–107.

[2] Y. Bar-Shalom, X. Li, Multitarget–Multisensor Tracking: Principles and Techniques, YBS, Storrs, CT, 1995.

[3] H.-J. Böhme, U.-D. Braumann, A. Brakensiek, A. Corradini, M. Krabbes, H.-M. Gross, User localisation for visually based human–machine interaction, in: Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, 1998, pp. 486–491.

[4] S. Coradeschi, A. Saffiotti, Anchoring symbols to sensor data: preliminary report, in: Proceedings of the Conference of the American Association for Artificial Intelligence, 2000, pp. 129–135.

[5] S. Coradeschi, A. Saffiotti, Perceptual anchoring of symbols for action, in: Proceedings of the International Conference on Artificial Intelligence, 2001, pp. 407–412.

[6] T. Darrell, G. Gordon, M. Harville, J. Woodfill, Integrated person tracking using stereo, color, and pattern detection, International Journal of Computer Vision 37 (2) (1998) 175–185.

[7] St. Feyrer, A. Zell, Robust real-time pursuit of persons with a mobile robot using multisensor fusion, in: Proceedings of the International Conference on Intelligent Autonomous Systems, Venice, 2000, pp. 710–715.

[8] J. Fritsch, S. Lang, M. Kleinehagenbrock, G.A. Fink, G. Sagerer, Improving adaptive skin color segmentation by incorporating results from face detection, in: Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication (ROMAN), Berlin, Germany, September 2002, IEEE, pp. 337–343.

[9] M. Kirby, L. Sirovich, Application of the Karhunen–Loeve procedure for the characterization of human faces, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (1) (1990) 103–108.

[10] M. Lindstrom, M. Andersson, A. Orebäck, H.I. Christensen, ISR: an intelligent service robot, in: H.I. Christensen, H. Bunke, H. Noltmeier (Eds.), Sensor Based Intelligent Robots, Proceedings of the International Workshop on Sensor Based Intelligent Robots, Selected Papers, vol. 1724, Dagstuhl Castle, Germany, September–October 1998, Lecture Notes in Computer Science, Springer, New York, 1999, pp. 287–310.

[11] H.G. Okuno, K. Nakadai, K. Hidai, H. Mizoguchi, H. Kitano, Human–robot interaction through real-time auditory and visual multiple-talker tracking, in: Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Maui, HI, 2001, pp. 1402–1409.

[12] N. Oliver, A. Pentland, F. Berard, LAFTER: a real-time face and lips tracker with facial expression recognition, Pattern Recognition 33 (2000) 1369–1382.

[13] Y. Raja, S.J. McKenna, S. Gong, Colour model selection and adaptation in dynamic scenes, in: Proceedings of the European Conference on Computer Vision, Freiburg, Germany, 1998, pp. 460–474.

[14] Y. Raja, S.J. McKenna, S. Gong, Segmentation and tracking using colour mixture models, in: Proceedings of the Asian Conference on Computer Vision, vol. 1, Hong Kong, 1998, pp. 607–614.

[15] C. Schlegel, J. Illmann, H. Jaberg, M. Schuster, R. Wörz, Vision based person tracking with a mobile robot, in: Proceedings of the British Machine Vision Conference, Southampton, UK, 1998, pp. 418–427.

[16] R.D. Schraft, B. Graf, A. Traub, D. John, A mobile robot platform for assistance and entertainment, Industrial Robot 28 (1) (2001) 29–34.

[17] D. Schulz, W. Burgard, D. Fox, A.B. Cremers, Tracking multiple moving objects with a mobile robot, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, vol. 1, Kauwai, HI, 2001, pp. 371–377.

[18] M. Soriano, B. Martinkauppi, S. Huovinen, M. Laaksonen, Skin detection in video under changing illumination conditions, in: Proceedings of the International Conference on Computer Vision and Pattern Recognition, vol. 1, Barcelona, Spain, 2000, pp. 839–842.

[19] M. Störring, H.J. Andersen, E. Granum, Physics-based modelling of human skin colour under mixed illuminants, Robotics and Autonomous Systems 35 (3–4) (2001) 131–142.

[20] M. Turk, A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuro Science 3 (1) (1991) 71–86.

[21] J. Vermaak, A. Blake, M. Gangnet, P. Perez, Sequential Monte Carlo fusion of sound and vision for speaker tracking, in: Proceedings of the International Conference on Computer Vision, vol. 1, 2001, pp. 741–746.

[22] S. Waldherr, S. Thrun, R. Romero, A gesture based interface for human–robot interaction, Autonomous Robots 9 (2) (2000) 151–173.

[23] C. Wren, A. Azarbayejani, T. Darrell, A. Pentland, Pfinder: real-time tracking of the human body, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 780–785.

[24] J. Yang, W. Lu, A. Waibel, Skin-color modeling and adaptation, in: Proceedings of the Asian Conference on Computer Vision, vol. 2, Hong Kong, 1998, pp. 687-694.

J. Fritsch received the Diploma in Electrical Engineering from the Ruhr-University Bochum, Germany, in 1996. He joined the research group for Applied Computer Science at Bielefeld University, Germany, in 1998. There he is working in the Collaborative Research Center 'Situated Artificial Communicators'. His research interests are adaptive color segmentation, the recognition of manipulation actions based on symbolic and sensory data, and the realization of advanced interfaces for human–machine interaction.

M. Kleinehagenbrock received the Diploma in Computer Science from the RW-TH Aachen, Germany, in 2001. He joined the Research Group for Applied Computer Science at Bielefeld University, Germany, in 2001, as a Ph.D. student in the graduate program 'Strategies and Optimisation of Behavior'. His primary interest is the integration of vision and speech modules on a mobile system to realize an advanced human–robot interface.

S. Lang received the Diploma in Computer Science from University of Bielefeld, Germany, in 2000. He is currently pursuing a Ph.D. program in Computer Science at the University of Bielefeld in joint affiliation with the Applied Computer Science Group and the graduate program 'Task-oriented Communication'. Sebastian Lang is interested in image processing, pattern recognition and human–machine interaction.

T. Plötz received the Diploma in Technical Computer Science from the University of Cooperative Education, Mosbach, Germany, in 1998. In 2001 he received the Diploma in Computer Science from Bielefeld University, Germany. He joined the Research Group for Applied Computer Science at Bielefeld University, Germany, in 2001. Thomas Plötz is interested in HMM-based pattern recognition in the field of speech-processing and bioinformatics.

G.A. Fink received the Diploma in Computer Science from the University of Erlangen-Nürnberg, Erlangen, Germany, in 1991 and the Ph.D. degree (Dr.-Ing.) also in Computer Science from Bielefeld University, Germany, in 1995. In 2002 he received the Venia Legendi (Habilitation) in Applied Computer Science from the Faculty of Technology of Bielefeld University. In 1991 he joined the Applied Computer Science Group at the Faculty of Technology of Bielefeld University where he is currently an Assistant Lecturer. His fields of research are speech and handwriting recognition, spoken language understanding, man–machine interaction, and distributed systems for pattern analysis applications. He has published various papers in these fields, and is author of a book on the integration of speech recognition and understanding. Dr. Fink is a Member of the Institute of Electrical and Electronics Engineers (IEEE).

G. Sagerer received the Diploma and the Ph.D. (Dr.-Ing.) Degree in Computer Science from the University of Erlangen-Nürnberg, Erlangen, Germany, in 1980 and 1985, respectively. In 1990 he received the Venia Legendi (Habilitation) in Computer Science from the Technical Faculty of this university. From 1980 to 1990 he was with the Research Group for Pattern Recognition at the University of Erlangen-Nürnberg, Erlangen, Germany. Since 1990 he is a Professor of computer science at the University of Bielefeld, Germany, and Head of the Research Group for Applied Computer Science. His fields of research are image and speech understanding including artificial intelligence techniques and the application of pattern understanding methods to natural science domains. He is author, coauthor, or editor of several books and technical articles. Dr. Sagerer is a Member of the German Computer Society (GI), the European Society for Signal Processing (EURASIP) and the Institute of Electrical and Electronics Engineers (IEEE).

ELSEVIER

Robotics and Autonomous Systems 43 (2003) 149–162

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

A meta-learning approach to ground symbols from visual percepts

Nicolas Bredeche[a,b,*], Yann Chevaleyre[a], Jean-Daniel Zucker[a], Alexis Drogoul[a], Gérard Sabah[b]

[a] *LIP6-CNRS, Université P&M Curie, Boite 169, 4, Place Jussieu, 75232 Paris Cedex 6, France*
[b] *LIMSI-CNRS, Université Paris XI, BP 133, F-91403 Orsay Cedex, France*

Abstract

There is a growing interest in both the robotics and AI communities to give autonomous robots the ability to interact with humans. To efficiently identify properties from its environment (be it the presence of a human, or a fire extinguisher or another robot of its kind) is one of the critical tasks for supporting meaningful robot/human dialogues. This task is a particular anchoring task. Our goal is to endow autonomous mobile robots (in our experiments a Pioneer 2DX) with a perceptual system that can efficiently adapt itself to the context so as to enable the learning task required to physically ground symbols. In effect, Machine Learning based approaches to provide robots with an ability to ground symbols heavily rely on ad hoc perceptual representation provided by AI designers. Our approach is in the line of meta-learning algorithms, that iteratively change representations so as to discover one that is well-fitted for the task. The architecture we propose is based on a widely used approach in constructive induction: the Wrapper-model. Experiments using the PLIC system to have a robot identify the presence of humans and fire extinguishers show the interest of such an approach that dynamically abstracts a well-fitted image description depending on the concept to learn.

Keywords: Anchoring; Meta-learning; Change of representation; Object identification

1. Introduction: anchoring symbols and identifying objects

Recent works in both robotics and artificial intelligence have shown a growing interest in providing mobile robots with the ability to interact and communicate with humans. One of the main challenges in designing such robots is to give them the ability to perceive the world in a way that is useful or understandable to us. One approach is to give the robot the ability to identify physical entities and relate them to perceptual symbols that are used by humans (to refer to these same physical entities). To perform this task, the robot has to ground these symbols to its percepts (i.e., its sensor data). Recently, the term of *anchoring* [1] has emerged to describe the *building and maintenance of the connection between sensor data and the symbols used by a robot for abstract cognition*. As a matter of fact, anchoring is an important issue for any situated robot performing abstract reasoning based on physically grounded symbols. Amongst others, anchoring plays an important role to communicate or relate to either other robots or humans.

There are tasks, such as object manipulation or functional imitation, where anchoring requires

[*] Corresponding author. Present address: LIP6-CNRS, Université P&M Curie, Boite 169, 4, Place Jussieu, 75232 Paris Cedex 6, France.
E-mail address: nicolas.bredeche@lip6.fr (N. Bredeche).

0921-8890/03/$ – see front matter © 2003 Elsevier Science B.V. All rights reserved.
PII: S0921-8890(02)00356-1

explicitly recognizing objects and localizing them in the three-dimensional space. Fortunately, such an *object recognition* task is not necessarily required to achieve anchoring. In applications such as human/object tracking, face and object identification, or grounded robot–human communication, *object identification* is enough. Informally, to *recognize* an object often requires from the robot to match its percepts with a known model of the object [2]. This task has been studied for a few decades now and is known to be difficult in unknown environments. On the contrary, *identifying* the sole presence of an object is simpler since its goal is to *classify* or to *name* an object [3]. As a matter of fact, there exist many easy to use and reliable descriptions for characterizing the presence of an object. To identify the presence of a fire in a room, one does not have necessarily to visually recognize it. Smelling smoke, hearing cracks, feeling heat, seeing dancing shapes on a wall are different ways of identifying the presence of a fire. For an autonomous robot, the ability to identify objects is a first step towards more complex tasks and may be built by regularly checking for the object. Identifying objects is therefore a simple form of anchoring symbols (such as *fire*) to its percepts.

In this paper, we are concerned with a practical task, where a Pioneer 2DX mobile robot has to rely on its limited visual sensors to anchor symbols such as *human being*, *mobile robot* or *fire extinguisher* (etc.) that it encounters while navigating in our laboratory. Anchoring is then used to support human/robot or robot/robot communication. For instance, an interaction may be engaged if a *human being* is identified, or a rescue operation may be initialized if a non-responding Pioneer 2DX is identified. Identifying a *fire extinguisher* may allow the robot to respond to a query formulated by a human. To design an autonomous robot, living in a changing environment such as our laboratory, with the identification ability described above is a difficult task to program. As such it is a good candidate for a Machine Learning approach, which may be easily recasted as a classical *concept learning task*. To teach the robot to anchor symbols using Machine Learning has proven successful [4]. To use Machine Learning techniques, the designer has to both define learning examples and a representation language based on the robot percepts to describe them.

It is clear that a great part of the success of the learning task per se depends on the representation chosen [5]. Having an AI designer providing the robots with an adequate representation has a major drawback: it is a fixed, ad hoc representation. Any change of setting (a museum instead of an AI lab) may require a new perceptual description. In order to overcome this drawback, our main objective is to endow an autonomous robot with the ability to dynamically abstract from its percepts different representations, well suited to learn different concepts. The intuitive idea is to have the robot explore the space of possible examples descriptions (with various colors, resolution, representation formalisms, etc.) so as to discover for each concept a well-fitted representation. The underlying intuition being that for anchoring the symbol *human being* a robot does not need the same visual *stimuli* that might be necessary to identify a *power-plug* on a wall.

Section 2 presents a concrete setting in which this problem occurs and pinpoints why adapting one's representation may be useful to increase learning accuracy. Section 4 explains our approach based on abstraction operators applied to visual information provided by the robot. Finally in Section 5 and Section 6, a set of real-world experiments describes the interest of such an approach and outlines the difference between three representations, each one fitted to a different concept (the presence of a human, a fire extinguisher or a box).

2. Problem settings

2.1. The MICRobES project

The practical task we are concerned which takes place in a wider project called MICRobES [6], which is a collective robotics experiment started in 1999 and involving more than 10 people. This project aims at studying the long-term adaptation of a micro-society of autonomous mobile robots in an environment populated by a human collectivity: the LIP6 laboratory in Paris. The robots, 10 Pioneer 2DX, have to survive in this environment as well as cohabit harmoniously with its inhabitants.

From an individual point of view, they need to recharge themselves autonomously, build the map of their environment in order to memorize its characteristics and localize themselves, avoid the mobile obstacles (human beings, other robots) and the potentially dangerous places (stairs, lifts).

From a collective point of view, they have to solve the spatial conflicts (access to the charging stations, coordination in navigation), cooperate by sharing information about the environment (open or closed rooms, etc.) and abide by some individualized constraints in their interactions with human beings (e.g. learning individual schedules and respect for privacy).

The colony of robots does not have, then, a functional goal, apart from being able to survive in an eco-system in which it must implement a robust and adaptive social structure. Thus, by studying robots that are physically as well as socially situated, MICRobES works towards two main goals: design a sufficiently autonomous and versatile robotics basis that can be used in different applications (distributed surveillance of buildings, guidance of visitors, etc.) and study, in collaboration with sociologists, the conditions required for immerging autonomous mobile robots in a larger public.

2.2. Anchoring and the building of a perceptual system

Inside the MICRobES project, we are concerned with providing the robot with the ability to perform robot–human communication about objects in the world. However, from the robot's point of view, using a shared lexicon of human symbols requires some prerequisites such as grounding these symbols in order to make sense in the world [7].

In this paper, we aim at providing each Pioneer 2DX autonomous mobile robot with the ability to identify (i.e. correctly classify) objects or living beings encountered in its environment thanks to mechanisms inspired from perceptual learning. As stated in Section 1, there is a strong difference between object *recognition* and *identification* as stated in [3]:

- *Object recognition* consists in finding a familiarity with an object for which there already exists a (*usually 3D*) model known by the system.
- *Object identification* consists in classifying or naming an object, i.e. it requires neither a model of the object nor complex scene reconstruction algorithms.

This identification ability will serve to build a lexicon of grounded human symbols in order to provide a basis for human–robot interaction-based behaviors (e.g. *dialogue* using the lexicon, request to *track* or *follow* an anchored object, etc.). This paper focus on the anchoring process while the use of a lexicon for such tasks will not be described here. It is important to understand that the anchoring process described here is independent of any behavior.

In practical, each robot navigates in the environment during the day and takes snapshots of its field of vision with its video camera according to three possible behaviors:

- *Wander behavior*. The robot explores its environment and takes snapshots from time to time. This behavior is useful to get a set of images that is representative of the environment.
- *Attention behavior*. The robot takes a snapshot upon a request. This enables a supervisor to show specific scenes.
- *Active learner behavior*. The robot explores its environment and takes snapshots that are supposed to be interesting according to what it already knows. In Machine Learning, such *active learning* techniques can greatly improve the accuracy of new classifiers by selecting examples based on the performance of previously learnt classifiers.

At the end of each day, the robot may report to a supervisor and "ask" her/him what objects (whose symbols may or may not belong to a predefined lexicon) are to be identified on a subset of taken pictures (without the supervisor pointing at them). It then performs a learning task in order to create or update the connection between sensory data and symbols which is referred to as the anchoring process. From a Machine Learning point of view, the learning tasks produces classifiers that should then be used to identify symbols from the sensory data. Fig. 1 describes this process. The learning task associated to the anchoring is therefore characterized by a set of image descriptions and attached labels. It corresponds to a multi-class concept learning task.

A key aspect of the problem lies in the definition of the learning examples (i.e. the set of descriptions extracted from the images) used by the robot during the anchoring process. In effect, a first step in any anchoring process is to identify (relevant) information out of raw sensory data in order to reduce the complexity of the learning task.

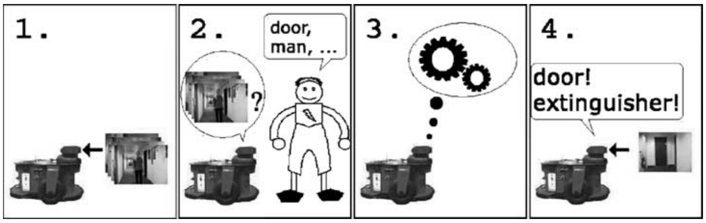


Fig. 1. The four steps toward lexicon anchoring. As a first step, the robot takes snapshots of its environment which are *labeled* by a supervisor. The robot tries to associate (learn) the provided label(s) with its percept, and, after a number of such steps take place, it shall be able to autonomously label a new environment.

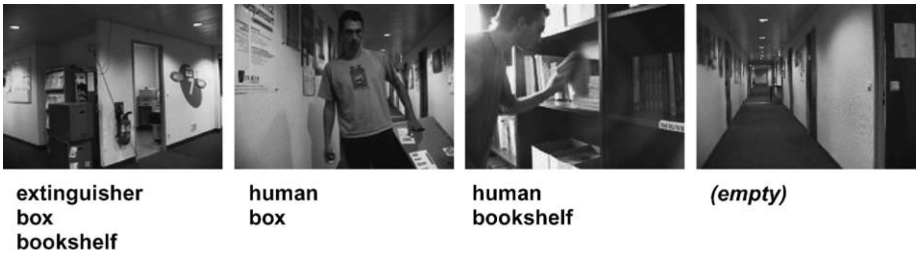


Fig. 2. Examples of the robot's visual experience.

The Pioneer2DX mobile robot provides images thanks to its LCD video camera while navigating in the corridors. The images are 160 × 120 wide, with a 24 bits color information per pixel. Humans, robots, doors, extinguishers, ashtrays and other possible targets can be seen among the images as shown in Fig. 2. All these possible targets, as they appear in the images, are of different shape, size, orientation and sometimes they are partially occluded. Finally, each image is labeled with the names of the occurring targets.

3. Related works

In this section, we will briefly review two research domains that are more or less related to our problem setting. We will try to highlight both the specificity of our task and the common concerns between problem settings. Firstly, we will state the differences with works studying the *emergence of language* in a society of robots. Then, we will study common concerns between *content-based image retrieval* and our identification task.

3.1. Emergent adaptive lexicon and language

Lexicon anchoring is mainly concerned with studying the evolution of a language in a society of agents through the emergence of a shared grounded lexicon. In order to build a shared lexicon, a group of agents may require a combination of individual adaptation, cultural evolution, and auto-organization [8].

These works do not focus much on the problem of extracting visual percepts. The world is perceived through few "channels" (such as *color*, *localization*, *height*, *width*) and *discrimination trees* are built incrementally to disambiguate words [9]. In fact, the problem of perceptions is simplified so that it is possible to study the evolution of language on a large scale (i.e. grounding meaning in a society of agents).

While these works achieved very interesting results and deals with the grounding of a lexicon, we are not concerned with the same issues. As a matter of fact, we consider the anchoring of a *given* lexicon, i.e. how to extract relevant information from complex images, instead of the *emergence* of such a lexicon, i.e. lexicon adaptation, evolution of grammar or syntax, etc.

3.2. Content-based image retrieval

Our problem setting shares much more in common with that of *content-based image retrieval* (CBIR) [10]. However the goal in CBIR is (roughly) to compute a similarity measure between two images, the question as to how the information is extracted remains central in both cases. As a matter of fact, we can learn much by studying the popular approaches used in CBIR to describe an image. There are three main approaches based on:

- *Global color histogram description* [11]. Matching image's histogram descriptions achieved surprisingly good retrieval results and is considered as a benchmark to evaluate other approaches. This approach is simple yet efficient.
- *Region-based similarity* [12,13]. The similarity measure is computed by matching regions grown according to various properties of the images (e.g. *color and texture properties*). However good results where achieved using this approach, there are known drawbacks such as the complexity of matching between images described as sets of regions and the unreliability of region-growing algorithms.
- *Configural recognition* [14]. This approach provides an efficient way to compare images using spatial properties between regions while limiting the matching complexity. Only a *fixed* number of regions according to a *given* configuration are taken into account. Since the *template* is given by the supervisor,[1] this approach is fitted for retrieval of images with constant overall organizations (i.e. scenes (e.g. mountain, sea, etc.) vs. objects).

The cited approaches can help us defining an image description mechanism but we should also take into account that there are strong differences between anchoring and CBIR. These fundamental differences are that:

(1) *Retrieval is not identification*. CBIR uses a similarity measure that do not explicitly classify the example. Moreover, learnt classifiers (set of rules, decision trees, trained neural networks, etc.) are faster to apply than computing any similarity measure (i.e. one-pass test vs. complex matching phase). Such classifiers enable nearly costless image classification and can easily be implemented in a real-time operating mobile robot.

(2) CBIR *is not a long-term behavior*. The robot is supposed to navigate in the environment and constantly update its anchors. Since the world is dynamic and subject to *concept drifts*, the robot requires to be able to learn and adapt its anchors through time (e.g. if a new example of the "chair" symbol may appear someday).

(3) *The images are not collected thanks to a situated behavior*. The data collected by the robot are specific to its location. Due to the properties of such images, we are concerned with checking if there is a specific property hidden in the image that would help to identify an object. As a matter of fact, the environment of the robot provides very similar images where global variations are not bounded to a given object. On the other hand, CBIR is about retrieving globally similar images among a set of very different images.

4. Changing the representation of images

4.1. Initial perceptual representation

We define the role of the robot's perceptual system as to extract *abstract percepts* out of *low-level percepts*, such as a set of pixels, from the video camera or sonar values. These abstract percepts provide a representation of the perceived world on which further computation will be based. They can be anything from sets of clustered colored regions to a matrix resulting from a Hough transform. The choice of a representation is motivated by finding a good trade-off that reduces the size of the search space and enhances the expressiveness of the abstract percepts.

As mentioned in Section 2, the problem we consider is that of automatically finding a representation of a set of labeled images that is well adapted to the learning of concepts. Let us underline that our goal is not to achieve the best performance on the particular learning task mentioned in the previous section. To obtain the best performance would require that experts in the field build an ad hoc representation for each concept to learn. On the contrary, we are interested in having

[1] The values for each component within the fixed template can also be learnt [15].

a robot find by itself the good representation, so that, if the context changes or the concept to learn is different, it has the ability to discover by himself the good level of representation. We therefore consider the representation provided by the sensors as an *initial* representation.

From the robot's point of view, each pixel from the camera is converted into a *low-level percept*. In the initial image representation, where each pixel is described by its position (*x*, *y*), its *hue* (the tint of a color as measured by the wavelength of light), its *saturation* (term used to characterize color purity or brilliance) and its *value* (the relative lightness and darkness of a color, which is also refereed to as "tone"). The initial description of an image is therefore a set of 19 200 (160 × 120 pixels) 5-tuple (*x*, *y*, *h*, *s*, *v*). Each image is labeled by symbols following the process described in Section 2 (see also Fig. 2). The *positive* examples of a given concept (e.g. presence of a fire extinguisher) to learn correspond to all images labeled positively for *this* concept. The *negative* examples are the images not labeled for *this* concept. As a matter of fact, a negative example for a given concept can be a positive example for another concept. The number of positive examples for each concept may vary greatly depending on the environment, the exploration of the robot, etc.

The initial representation of images, consisting of hundreds of thousands of pixels, is clearly a too low-level representation to be used by Machine Learning algorithms. We shall now analyze different representations that have been considered in the field of Computer Vision from the Machine Learning point of view. These different representations will provide some directions for investigating automatic changes of representation to improve the learning accuracy.

4.2. Representation languages in Machine Learning

In the traditional setting of Machine Learning, each object is represented by a *feature-vector* *x*, to which is associated a label *f*(*x*). The supervized learning task consists in finding a classifier *h* which minimizes the misclassification probability $\Pr[f(x) \neq h(x)]$ on a newly observed example (*x*, *f*(*x*)).

Within the *multiple-instance* setting [16], objects are represented by *bags of feature-vectors*. Feature-vectors are also called *instances*, as in the traditional setting features may be numeric as well as symbolic features. Again, the associated learning task consists in finding a classifier h_{multi} such that most bags are correctly classified. In this setting, multiple-instance classifiers are of the form $h_{\text{multi}}(b) = h(x_1)\text{v}\cdots\text{v}h(x_r)$ where $b = \{x_1, \dots, x_r\}$ is a bag containing *r* instances. Thus, an object represented by a bag *b* will be classified positively by h_{multi} iff at least one of its instances fires *h*. Multiple-instance learning has been successfully applied to various domains including the prediction the chemical activity of molecules [16], and the classification of natural scenes [15].

Within a *relational setting* the objects are represented by a set of components objects, their features, and relations between components. In particular, in Inductive Logic Programming [17] Prolog facts are used to describe objects and Background Knowledge *B* encodes deductive rules.

To summarize, in Machine Learning the languages used to represent examples fall into three broad categories:

- *Feature-vector*. The most widely used and for which efficient algorithms have been devised.
- *Relational description*. The most expressive representation but whose inherent complexity [18] prevents from efficient learning.
- *Multiple-instance*. An in-between representation, more expressive than feature-vector but for which efficient algorithms do exist.

4.3. Dimensions of abstraction

In the perspective of automatically exploring the set of possible representations of an image, we propose to identify particular operators and to experiment with them. There are countless operators that could be applied to an image hoping for more accurate learning. Operators changing the *contrast*, the *resolution*, the *definition* are all possible candidates.

To improve the learning of concepts, we are interested in transformation that are *abstractions* in the sense that they decrease the quantity of information contained in the image [5]. Abstraction is considered as a specific *change of representation* that is an *homomorphism* from one representation to another (here:

from an image to its description). Starting from the initial *low-level percepts* (i.e. the pixels of the image), the elements obtained after applying the *abstraction operators* will be referred to as *abstract percepts*, since they will be used as representative percepts for further processing.

The two main dimensions of abstraction that we shall study are *granularity* and *structure*. Granularity corresponds to the resolution of the image. Structure corresponds to the basic element of the image as the smallest individually accessible portion of the image to consider, be it a pixel or a complex region. Fig. 4 depicts the space of representation changes associated to these two dimensions and their corresponding abstraction operators that we define as:

- The *associate operator* (for granularity). It consists in replacing a set of pixels with a unique (mega)pixel that has for its (h, s, v) values the average of the pixels that were associated. This operator is a built-in operator for the robot as it corresponds to a particular *sub-sampling*. The resulting *abstract percepts* will be referred to as *r-percepts*.[2]
- The *aggregate operator* (for structure). It consists in grouping a set of pixels or regions to form a pattern. This operation is also referred to as "term construction" in the literature [19]. The pattern does not replace the pixels or regions it is composed of, and therefore the resolution or granularity of the image is not changed. What changes is the structure of the image. The aggregate operator may be either data-driven (e.g. growing patterns) or model based (e.g. applying a predefined mask). For reasons of efficiency required by the use of a robot we have considered an aggregate operator that is applied to contiguous pixels forming a particular shape (we do not consider region-growing algorithms because of their versatility when using fixed thresholds). The resulting *abstract percepts* will be referred to as *s-percepts*.[3]

Fig. 3 illustrates how one can use these operators by showing a practical example where specific instances of the associate and aggregate operators are sequentially applied to extract a new description from an image.

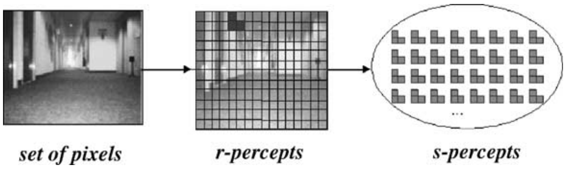


Fig. 3. An example where specific instances of the operators *associate* and *aggregate* are sequentially applied to an image.

5. Automatically changing the representation for learning

In the previous section two abstraction operators to change the representation of images were presented. The parameter of the associate operators we have considered is the number of pixels that are associated to form a (mega)pixel. The parameter of the aggregate operator is the pattern or region structure.

With respect to the learning task described in Section 2, a key issue is to analyze the impact of representation changes on learning. The main question is related to the choice of one operator and its parameters. In Machine Learning, the abundant literature on feature selection shows that approaches fall in two broad categories: the `wrapper` and the `filter` approach [20]. Intuitively, the `wrapper` approach uses the performance of the learning algorithm as a heuristic to guide the abstraction. In the following, we present how the `wrapper` approach can be used to choose the most fitted abstraction. As it is an approach that attempt to learn from the learning process itself it is also referred to as a *meta-learning* approach.

We have developed the PLIC system, which is both an image description toolkit, a data reformulation tool, and a Wrapper. PLIC interacts with RippMi, a *multiple-instance rule learner* that generates classifiers as decision rules (see [21] for a full description of RippMi). For example, a typical classifier would be (using a s-percept such as the one seen in Fig. 3:

- HYPOTHESIS: HUMAN.
- TRUE:- P3VALUE<=9, P2SATURATION>=27.
- TRUE:- P2HUE<=203, P1SATURATION<=3, P3VALUE<=165.
- TRUE:- P3HUE<=198, P1X>=6, P1Y>=2.
- DEFAULT FALSE.

[2] r-percepts, as in *resolution percept*.

[3] s-percept, as in *structural percept*.

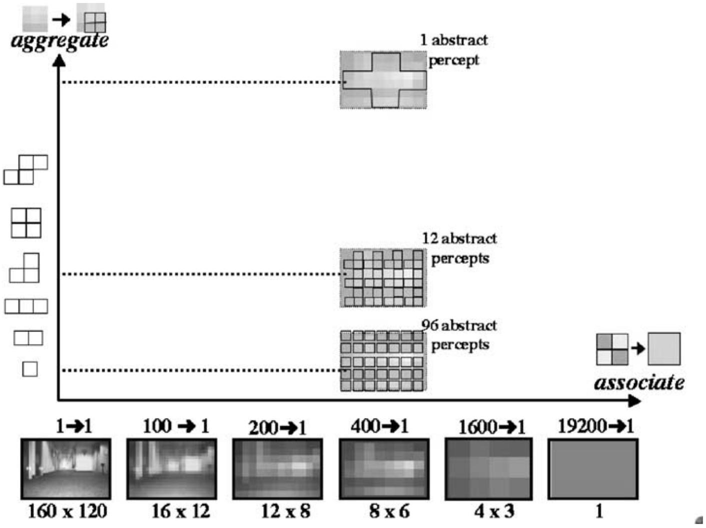


Fig. 4. The space of image representation obtained by applying the associate operator (changing the *resolution*) and the aggregate operator (changing the *structure*).

where P1, P2, P3 are the corresponding embedded r-percepts. RippMi cross-validates learnt classifiers in order to evaluate the average error rate on unknown data. This is generally a reliable estimation of the classifier's accuracy on future data.

With the help of RippMi, PLIC applies the operators as follows:

(1) *Association operator*. The horizontal dimension in Fig. 4 is difficult to explore since object identification is independent of scaling. Since there is no "better" resolution and that every resolutions should be useful, PLIC describes each image by using the associate operator for several image resolutions (namely 1 × 1, 4 × 3, 8 × 6, 16 × 12, 32 × 24[4]). The idea behind this *multi-granularities approach* is to learn classifiers that are invariant to resolutions and object size variations.

(2) *Aggregation operator*. PLIC uses its Wrapper component in order to explore the vertical dimension in Fig. 4. Exploring the vertical dimension is used to select between different structural patterns to apply with the aggregate operator. The Wrapper-based component explores different s-percepts iteratively as synthesized is shown in Fig. 5. An initial s-percept is chosen (at first, it embeds only one r-percept), and the image is reformulated in a multiple-instance representation using this structure; then, the concepts are learnt using this representation. Based on the results with cross-validation of the learning algorithm, a new structure is devised by adding a contiguous r-percept. The heuristic for creating a new structure is based on the fact that for the current s-percept, all the embedded r-percepts are used in at least one decision rule of the rule set with the accuracy being better than at the previous level. For example, the rule set we saw before would be extended (all three embedded r-percepts are used).

PLIC uses RippMi to learn several *classifiers* for each object to be identified (e.g. one classifier for each s-percepts). Each classifier is learnt thanks to a fixed number of positive and negative examples during a *batch learning session*. However these classifiers are cross-validated, it is possible that the robot may encounter new occurrences of the object. For example, people may change clothes, objects may be moved or replaces, the environment can vary greatly during the day (e.g. daylight vs. artificial light), etc.

Fortunately, all these classifiers can be combined in order to evaluate which one have to be replaced. PLIC

[4] We were not able to go further due to memory limitations.

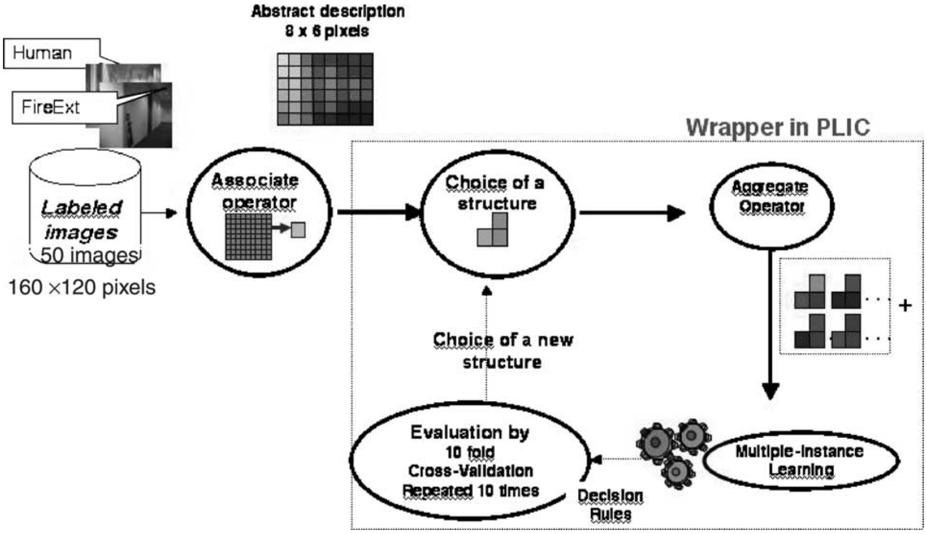


Fig. 5. The PLIC system Wrapper component.

addresses *anchoring in the long-term* by increasing or decreasing each classifier's *weight* depending on the accuracy of its prediction when new images are presented. Many on-line learning algorithms can be used such as the weighted majority algorithm [22] or even a simple perceptron. As a consequence, we can easily replace outdated classifiers by the newly learnt classifiers whenever there is a batch learning session. For a given concept, such a session can be launch once *n* new images have been labeled with this concept (choosing *n* is free but may take into account memory limitation since this is a batch learning task where all positive and negative examples are handled at the same time as opposed to on-line learning).

6. Experiments

6.1. Experimental setup

To evaluate the interest of abstracting visual percepts from a Machine Learning point of view, a number of different experiments have been carried out. The experiments presented are based on the images acquired by a PIONEER2DX mobile robot in the corridor of the *LIP6* laboratory (Paris, France). The objects as they appear in the images are different in shape, size, orientation and are sometimes partially occluded. The lexicon contains three symbols to anchor:

(1) *Human*. A single person with different kind of clothes.
(2) *Fire extinguisher*. They can be found in the corridor of our lab.
(3) *Box*. Various boxes that stand alone or piled up.

As explained in Section 2, a supervisor "names" the occurring targets. Given the symbol to anchor, we have decided that every batch learning session would be based on the first 25 positively labeled examples and 25 randomly selected negative examples (no bias). The size of the corresponding descriptions depends on the operators. For example, a learning set may vary from approximately 2 KB (with a global histogram description for each image) to 3 MB (with an associate operator set to 32 × 24 and an aggregation operator with s-percepts that embed 4 r-percepts). Among the positive examples, about 50% are labeled with one object, 15% with two objects and 5% with three objects.

Three independent sets of experiments are presented. The first one illustrates the impact of the operator associate used to build a multi-granularities descriptions. The second studies the impact of the aggregate operator based on an arbitrarily selected granularity. The multiple-instances rule learner RIPPERMI was used on the descriptions obtained from these

Table 1
Object detection accuracy (%) and granularity

	Human	Extinguisher	Box
Global histogram (baseline)	61.83	64.72	77.78
4 × 3	50	63.89	69.17
8 × 6	50.28	66.67	86.11
16 × 12	63.61	58.61	82.5
32 × 24	72.5	70.28	56.67
Multi-granularities	72.8	75.28	75.8

images with a 10-fold cross-validation.[5] Moreover, each experiment is repeated 10 times in order to get a good approximation of the results. In Machine Learning, such a validation is known to compute a good approximation of what will be the real accuracy of the classifiers (i.e. the object identification accuracy). RipperMi returns a set of rules (i.e. a classifier) that covers the positive examples. Finally, we describe the use of weights to evaluate classifiers obtained at the previous steps and show the benefits of updating the anchor of an object through time.

6.2. Evaluating automatic changes of granularity

To begin with, we performed a simple learning task using Ripper [23], a well-known supervized learning algorithm, with a learning set consisting of the popular[6] global histogram descriptions of the images. This will serve as a baseline to evaluate the impact of choosing a specific granularity.

Table 1 shows the object identification accuracy for the three concepts.[7] According to the results, it is not clear which resolution is better. The experiment with the global histograms sometimes yields better results that experiment with finer grain. Moreover, the multi-granularities approach seems to yield only slightly better results than other approach and is even worse for the easy-to-learn "box" concept. Nevertheless, the multi-granularities approach produces classifiers that are resolution independent: each classifier is learnt on a dataset where each image is described in four different ways (i.e. 4×3, 8×6, 16×12 and 32×24 representations) that generates four distinct examples.

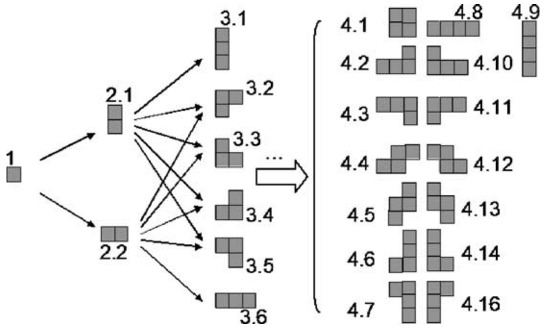


Fig. 6. Four levels of structural configurations (i.e. s-percepts) generated by PLIC.

Clearly, object identification depends on the object and its accuracy is subject to change through time and experience. While the multi-granularity approach do not always yield the best results, there are good chances that its classifiers will prove more robust in time than other classifiers.

6.3. Experiments on automatic changes of structure

PLICs Wrapper tool was used with the heuristic described in Section 5 in order to generate up to a maximum of 4 r-percepts per s-percept. The possible structural configurations are shown in Fig. 6. Each structural configuration is applied from every single r-percept to generate the learning sets. The 32 × 24 resolution was chosen in order to show the potential of structural reformulation. Table 2 shows the best results achieved for each structural level of complexity.

Results from the experiments show that for all the objects, the highest accuracy is achieved by one of the most complex structural configurations, which is not

Table 2
Object detection accuracy (%): best results for each structure level

	Human	Extinguisher	Box
Global histogram (baseline)	61.83	64.72	77.78
Level 1	72.5	70.28	56.67
Level 2	73.5 (2.2)	73.33 (2.2)	60 (2.2)
Level 3	76.67 (3.5)	73.33 (3.4)	76.7 (3.4)
Level 4	80.8 (4.10)	80.8 (4.3)	85.33 (4.16)

[5] Cross-validation is a widely used data-oriented evaluation of the learning generalization error. The dataset is divided into a learning and a training set.

[6] At least in cbir.

[7] Learning duration is less than 10 s standard deviation is about 1%.

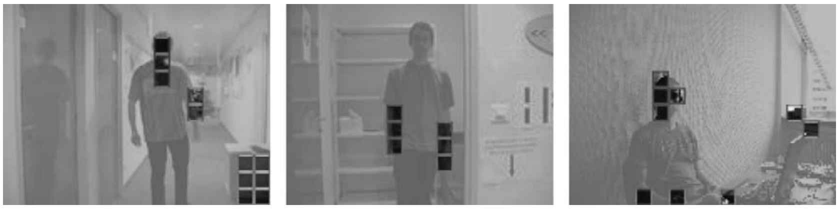


Fig. 7. Three snapshots taken during a tracking behavior (with identification).

surprising. However the structural configurations are still quite simple, the identification accuracy for each object rose between 8 points (human detection) and 29 (!) points (box detection).

Eventhough the impact of modifying the aggregation operator depends on the concept to learn (same as the association operator), structural reformulation is clearly an efficient way to improve classifiers for anchoring. The classifier shown in Section 5 was learnt from a reformulated dataset using the "3.3" s-percept. This classifier demonstrates that relations between the embedded r-percepts are taken into account.

6.4. On-line learning and updating anchors

We saw previously that PLIC and RipperMi require a fixed number of examples during a *batch learning session*. As a consequence, these *batch learning algorithms* build efficient classifiers as long as examples are representative of the world. Given the world is dynamic and unstable, we have to update the anchors from time to time. An interesting approach is to combine an on-line learning algorithm such as the well-known *weighted majority algorithm* [22] with our batch learning algorithms. Such an on-line learner would:

- Provide a global detection prediction by aggregating the weighted classifiers predictions.
- Improve the global performance of a set of classifiers in the long-term by evaluating them (classifiers with a low accuracy are replaced by new ones).

We experimented this algorithm during several batch learning sessions and it proved to be efficient thanks to the following characteristics:

(1) *It naturally improves classifiers*. Each learning session is based on a specific set of data. If the robot's environment is (more or less) stable, it is possible to grasp new object's properties or to take better snapshots. This sometimes results in learning better classifiers that can slightly increase the global identification accuracy for an object. In our experiments, we empirically evaluated this as less than a 5% growth in object identification accuracies for different kind of redundant objects.

(2) *It performs long-term adaptation to concept drifts*. We experimented on the tracking by identification of a human being dressed in grey and black. The robot built its classifiers during two learning sessions. Then, the robot was made unable to track the target because the human dressed in blue and white. After two other learning sessions, new classifiers were built and the robot could track the human target again. What is important here is that some of the old classifiers still remained. These few classifiers relied on *skin* and *hair* colors, which are constant human features.

Fig. 7 shows three snapshots taken during tracking human (and other objects). Different classifiers were used depending on the image. On the first image, classifiers identified a human based on *t-shirt*, hair and skin colors using different structures. A box is also identified. On the second image, human detection relies simply on the color of the skin. Finally, the third image shows an example of wrong detection on the right part of the picture due to an unknown environment (bureau vs. corridor). Nevertheless, the human is also detected thanks to skin-based classifiers and a "t"-like structure classifier that covers the face (skin and hair).

7. Conclusion

In this paper we have addressed the problem of using automatic abstraction of visual percepts by an autonomous mobile robot to improve its ability to learn *anchors* [1]. This work finds its application in a real-world environment within the MICRobES multi-robots project [6], where anchors provides a basis for communication between the Pioneer 2DX robot and its human interlocutor. In the approach we proposed the robot starts with the initial low-level representation of the images it perceives with its LCD video camera, and iteratively changes their representation so as to improve the learning accuracy. Between the low-level pixel representation and a global histogram representation there is an immense space of possible representations. To explore part of this abstract space of representation we have identified two operators. A first one changes the resolution and loose information by averaging the color of squares of pixels. A second one that groups pixels without changing the resolution.

To guide the exploration of the space of possible abstractions, we have developed the PLIC system which uses the learning results in order to select the abstract operators to be applied. From a Machine Learning point of view, this architecture is based on a widely used approach in feature selection: the Wrapper-model. The set of experiments that have been conducted show that both operators do impact on the learning accuracy. It is interesting to notice that the best resolution and structure (sort of coordinates in the abstract space) found by the system depends of the concept.

It is also clear that as the number of examples increases, different reformulations might perform better. Creating high-level abstract percepts does not only improve accuracy, it makes object identification faster for the robot. This is true as long as the abstraction process do not takes itself too much time. This is a known trade-off in the field of abstraction [24]. As a matter of fact, abstracting regions by using region-growing algorithm was a candidate abstract operators but its computation is too costly for on-line identification.

This study shows that for learning anchors, an approach that periodically searches for the most accurate representation, given the examples at hand, is a promising direction. Moreover, it appears that for each anchor that needs to be learnt, different abstractions might be more appropriate. These findings raise several questions with respect to the robot architecture. The search for a better representation should be triggered by a decrease of performance of the acquisition of new examples? How to compare the application of operators that change the resolution and operators that change the structure? A central question for any lifelong learning system, integrating abstraction abilities, is to decide whether to continue to *exploit* its current representation or *explore* new representations at the risk of loosing resources if no better ones is found.

Acknowledgements

Pr. Zucker is supported by a research grant from the CNRS (délégation), and a cognitique project (on spatial cognition). The MICRobES project is supported by three grants from the French Department of Higher Education and Research, by a grant from the LIP6 for the collaboration between research teams, and by a research contract between the LIP6 and France Telecom R&D.

References

[1] S. Coradeschi, A. Saffiotti, Anchoring symbols to sensor data: preliminary report, in: Proceedings of the AAAI-2000, Austin, Texas, 2000, pp. 129–135.

[2] D. Marr, Vision, Freeman, Oxford, 1982.

[3] J. Stone, Computer vision: What is the object? in: Prospects for AI, Proceedings of the Artificial Intelligence and Simulation of Behaviour (AISB), IOS Press, Birmingham, England, 1993, pp. 199–208.

[4] V. Klingspor, K. Morik, A.D. Rieger, Learning concepts from sensor data of a mobile robot, Machine Learning 23 (2–3) (1996) 305–332.

[5] L. Saitta, J.-D. Zucker, A model of abstraction in visual perception, Applied Artificial Intelligence 15 (8) (2001) 761–776.

[6] S. Picault, A. Drogoul, The microbes project, an experimental approach towards open collective robotics, in: Proceedings of the Fifth International Symposium on Distributed Autonomous Robotic Systems, Springer, Tokyo, 2000.

[7] S. Harnad, The symbol grounding problem, Physica D 42 (1990) 335–346.

[8] L. Steels, Emergent adaptive lexicons, in: P. Maes (Ed.), Proceedings of the Simulation of Adaptive Behavior Conference, MIT Press, Cambridge, MA, 1996.

[9] L. Steels, Perceptually grounded meaning creation, in: Proceedings of the First International Conference on Multi-Agent Systems, 1996, pp. 338–344.

[10] J. Eakins, M.E. Graham, Content-based image retrieval, in: Report to JISC Technology Applications Programme, Institute for Image Data Research, University of Northumbria at Newcastle, 1999.

[11] M. Stricker, M. Swain, The capacity and the sensitivity of color histogram indexing, Technical Report 94-05, Communications Technology Lab, ETH-Zentrum, 1994.

[12] J.Z. Wang, Integrated Region-based Image Retrieval, The Kluwer International Series on Information Retrieval, vol. 11, Oxford, 2001.

[13] S. Belongie, C. Carson, H. Greenspan, J. Malik, Color- and texture-based image segmentation using em and its application to content-based image retrieval, in: Proceedings of the International Conference on Computer Vision, 1998.

[14] P. Lipson, E. Grimson, P. Sinha, Configuration based scene classification and image indexing, in: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR'97), IEEE Press, New York, 1997.

[15] O. Maron, A.L. Ratan, Multiple-instance learning for natural scene classification, in: Proceedings of the 15th ICML, 1998, pp. 341–349.

[16] T.G. Dietterich, R.H. Lathrop, T. Lozano-Pérez, Solving the multiple-instance problem with axis-parallel rectangles, Artificial Intelligence 89 (1–2) (1997) 31–71.

[17] S. Muggleton, Inductive logic programming, New Generation Computing 8 (4) (1991) 295–318.

[18] A. Giordana, L. Saitta, Phase transitions in relational learning, Machine Learning 41 (2) (2000) 217–241.

[19] A. Giordana, L. Saitta, Abstraction: a general framework for learning, in: Working Notes of the AAAI Workshop on Automated Generation of Approximations and Abstraction, Boston, MA, 1990, pp. 245–256.

[20] R. Kohavi, G. John, The Wrapper approach, in: H. Liu, H. Motoda (Eds.), Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publishers, Dordrecht, 1998, pp. 33–50.

[21] Y. Chevaleyre, N. Bredeche, J.-D. Zucker, Learning rules from multiple instance data: issues and algorithms, in: Proceedings of the Ninth International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU02), Annecy, France, 2002.

[22] N. Littlestone, M. Warmuth, The weighted majority algorithm, in: Proceedings of the IEEE Symposium on Foundations of Computer Science, 1989, pp. 256–261.

[23] W.W. Cohen, Fast effective rule induction, in: Proceedings of the 12th International Conference on Machine Learning, Morgan Kaufmann, Los Altos, CA, 1995.

[24] F. Giunchiglia, Using abstrips abstractions: Where do we stand?, Artificial Intelligence Review 13 (1996) 201–213.

Nicolas Bredeche received his master degree in Computer Science from the University of Pierre et Marie Curie (Paris 6), France, and in 1999, his DEA (French MSc) in cognitive sciences from the University of Paris XI. He is currently completing his Ph.D. in Artificial Intelligence at the AI lab of Paris 6, Machine Learning team. His research interests include machine learning, robotics, abstraction, vision, perception, perceptual learning, anchoring, symbol grounding and human-robot communication.

Yann Chevaleyre is an associate professor in Computer Science at University of Paris 9. He received his Ph.D. and his DEA (French MSc) in Artificial Intelligence from the University of Paris 6. He also has worked as a postdoctoral fellow at university of Pernambuco, Brazil. His research interests include multiple-instance learning, computational learning theory, and learning in multiagent systems.

Jean-Daniel Zucker is a former Aeronautical Engineer (Sup'Aéro, 1985). He worked in R&D for six years. He got his Ph.D. in 1996 in Machine Learning from Paris 6 University where he became an associate professor focusing on representation changes and abstraction in learning. In 2002 he became full professor at Paris University where he leads a Machine Learning team. His research focus includes now DNA chips data mining.

Alexis Drogoul is a full professor in the LIP6 laboratory. He leads the MIRIAD team, which focuses on multi-agent simulation, reactive agents, collective problem solving and collective mobile robotics within the OASIS theme. He is interested in the study of the emergence of spatial, temporal, behavioral, and/or social structures within reactive multi-agent systems. His research is thus covering a wide range, from the simulation of animal societies to the design of multi-robots systems, problem-solving systems, and agent-oriented methodologies. He is member of the program committee of the major conferences in these domains (IJCAI, AAMAS, SAB, etc.) and expert for the NSF and EEC on agent-based technologies.

Gérard Sabah graduated from the 'École Polytechnique' in 1971 and entered the CNRS in the same year. Besides his engineering degree from Polytechnique, further degrees include a 'Doctorat d'État ès Sciences' (1978) on Natural Language understanding by Computers. He is currently a Research Director at CNRS. His present interest is the study of the cognitive processes in natural language understanding, acquisition and generation, and its relation with reasoning and consciousness. Until recently he was responsible of the natural language pole of the Man-Machine Communication PRC (National Programme of Co-ordinated Research) and responsible for the Cognitive Science Network in southern Paris. He has also been the president of the French Association for Cognitive Research and member of the AFIA (French Association for Artificial Intelligence—member of the ECCAI federation) bureau and editor in chief of its journal.

ELSEVIER

Robotics and Autonomous Systems 43 (2003) 163–173

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Shared grounding of event descriptions by autonomous robots

Luc Steels [a,b,*], Jean-Christophe Baillie [a]

[a] *SONY Computer Science Laboratory, Paris, France*
[b] *Artificial Intelligence Laboratory, Vrije Universiteit Brussel, Brussels, Belgium*

Abstract

The paper describes a system for open-ended communication by autonomous robots about event descriptions anchored in reality through the robot's sensori-motor apparatus. The events are dynamic and agents must continually track changing situations at multiple levels of detail through their vision system. We are specifically concerned with the question how grounding can become shared through the use of external (symbolic) representations, such as natural language expressions.

Keywords: Autonomous robots; Event descriptions; Open-ended

1. Introduction

The work reported in this paper is part of a larger research effort towards grounded open-ended self-generated communication among robots [22] and grounded open-ended natural language-like communication between humans and robots [19]. *Grounded* means here that the communication is about the shared environment in which speaker and hearer are situated and which has to be perceived and interpreted autonomously by both participants through a perceptual apparatus. This contrasts with natural language interfaces to purely symbolic information systems like databases [3] or communication systems for software agents [4] in which the agents have full access to an accurate and complete representation of the environment and each others' internal states. *Open-ended* means that the communication conventions are not fixed in advance but they are negotiated and adapted to suit the communication needs of the partners. This appears necessary because human communication is open-ended as well. Humans may invent new meanings and new expressions for these meanings or adapt existing expressions to serve new purposes as part of a normal conversation [9].

Grounded verbal communication is an enormously challenging task which requires the integration of many capabilities, including speech and language processing. We believe that there is no single sweeping principle that will make a non-grounded AI system grounded. It is definitely not the case that one can simply attach a vision/action module to a logic-based reasoning system to obtain a grounded agent, nor that one can simply put a conceptual system on top of a behaviour-based robot. Instead, grounding is a matter of embodiment and very careful design, as well as tight integration of many components at many different levels. Nevertheless, it is possible to identify a set of issues that need to be dealt with and general design principles or strategies.

This paper starts with an introduction into the grounding issue in an attempt to clear up possible terminological confusions. It then describes very briefly a system that we have built which is a combination

[*] Corresponding author. Present address: SONY Computer Science Laboratory, Paris, France.
E-mail address: steels@arti.vub.ac.be (L. Steels).

0921-8890/03/$ – see front matter © 2003 Elsevier Science B.V. All rights reserved.
PII: S0921-8890(02)00357-3

and integration of two subsystems reported on earlier: the PERACT system [5], designed for the visual recognition of actions, and the EVOLAN system [20], designed for exploring the symbolisation of event description in multi-agent simulations. This paper focuses mainly on the visual processing and event categorisation that anchor symbols in the world. We next turn to a discussion of the underlying design principles and end with some conclusions.

2. Terminological issues

There has been a big debate in AI and cognitive science on whether intelligence requires symbolic representations [8], and if so how these representations are supposed to be related to the world through a sensory apparatus and how this relation is acquired [11,13].

For the purposes of clarifying this discussion, we have found it useful to make a number of new distinctions. Generally speaking, representations code meanings, i.e. features of the environment relevant to the agent. We can distinguish internal and external representations. Internal representations occur when the physical structures are located inside computer memories or in brains. External representations are physical structures outside the individual: marks on a piece of paper, sounds, gestures, objects. Communication between two agents always requires external representations.

Another useful distinction which has been introduced by Sperber and Wilson, is that between Shannon coding and inferential coding, which gives rise to a distinction between information representations (I-representations) and expressive representations (E-representations), respectively. Natural language utterances are clear examples of inferential coding and thus of E-representations [18]. The interpreter is assumed to be intelligent and capable to infer meaning from mere hints. As a consequence the representation can be more compact because the interpreter shares sufficient context and background knowledge to make the appropriate inference. Most importantly, the representation need not be exclusively based on established conventions but can be the outcome of a negotiation process. Thus analogy is heavily used in natural language to invent a way for expressing a new meaning. For example, soon after Douglas, Engelbart developed the new concept of an "*x*–*y* position indicator for a display system" in the form of a box rolling over the table, it was called a mouse by analogy with the shape of a real mouse and now the whole world calls it that way. This shows that inferential coding can potentially express an open-ended set of meanings because the coding conventions can be adapted as the needs arise.

The information structures typically used in computers are examples of Shannon codings, and further called I-representations. No intelligent interpreter is assumed and so interpretation is straightforward and automatic. There is even a question whether one can speak of interpretation. All the information is in the message itself and the coding is fixed. There is no need to go through a complex process of disambiguation or the guessing of meaning. The production and interpretation of E-representations clearly requires information processing, both to code the meaning that needs to be expressed and the partial structures (such as syntactic structures) generated as part of the production and interpretation process (which sometimes even involves a model of the listener). But this does not necessarily imply that the brain internally uses E-representations. We do not want to go into that discussion here, except to point out that often philosophers, anthropologists, artists, etc. use the term representation in the sense of (external) E-representation whereas computer scientists or AI researchers use it in the sense of (internal) I-representations. Our more precise terminology is proposed to avoid this confusion.

The term grounding applies to all possible representations, and the opposite of a grounded representation is a formal representation, like an uninterpreted algebra. A grounded representation has intentionality; it is about objects and situations in the world. This implies that the agent needs processes to establish and maintain this relation. For example, there might be internal I-representations in the form of data structures (or states in neural networks) that code for the colour, position, shape, size, trajectory, speed of movement, etc. of an object in the world and these could be constructed and maintained by a vision system that is segmenting images, tracking them, and computing their properties in real time. External representations could also be grounded, in the sense that a description produced by one agent could be about an object or situation in reality and the other agent has to ground the meaning of this description in his own perception of reality in order to understand it.

A key issue, and the one we try to solve in the experiments discussed in this paper, is how *shared grounding* can occur. This is a big problem because the agents do not have access to each other's internal states (i.e. each other's internal representations). We argue that shared grounding can be established through a negotiation process embedded in language games. So the 'symbols' that we are trying to see grounded are external symbols used in language-like communication, they are not internal symbols used in some cognitive process. Not only are we interested in single symbols (words) but also, and particularly, in the shared grounding of the meaning of grammatical structures. By negotiation we mean that agents invent representational conventions, try them out with others, and adapt their set of conventions based on the feedback on success or failure in communication.

There has been quite a lot of work (as illustrated by this special issue, as well as the papers in [10]) on how a single agent can ground his internal I-representations in reality by a sensory-motor apparatus. But there has been little work so far on shared grounding through external E-representations, i.e. how a population of agents, each with a grounded representation system, can evolve agreement on how their respective internal representations are coordinated through external representations. Other papers describing our approach (see, for example [19,22]) have focused mainly on the language part, whereas this paper focuses exclusively on the anchoring components, i.e. the vision and tracking system that generates the internal representations to be expressed.

3. Grounded language communication

The robotic installation used for the present paper is displayed in Fig. 1 and similar to that used in earlier 'talking heads' experiments [22]. It consists of two SONY pan-tilt cameras (EVI-D31) each connected to a computer, which runs the PERACT system. The computers are Bi-Xeon 1.7 GHz machines running Linux Redhat 7.1. The language-specific aspects of the system (parsers, producers, etc.) run on a third computer (Mac G4 with Common LISP) with communication through a local area network.

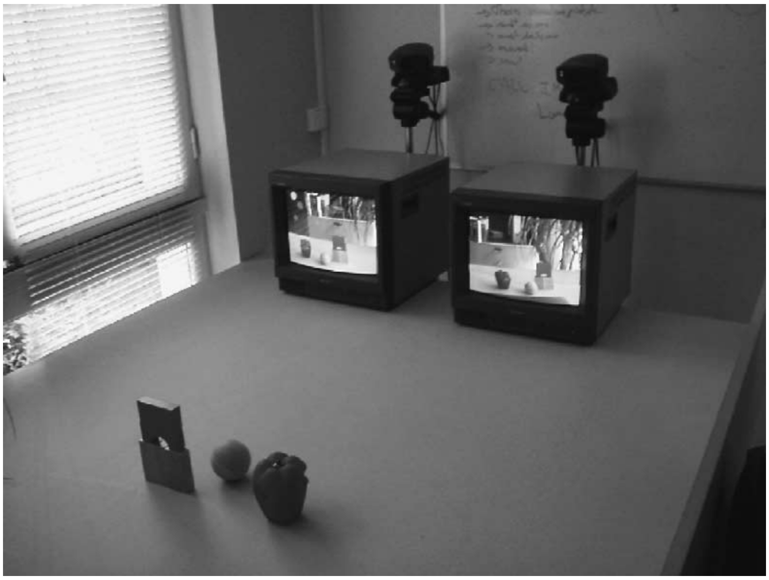


Fig. 1. Robotic installation used for the experiments reported in this paper. It consists of two steerable cameras capturing images of dynamic scenes. The captured images are shown on separate monitors.

3.1. Language games

The robots engage in interactions which we call language games [19]. A language game is a routinised, situated interaction between two agents. The interaction not only involves verbal communication, i.e. the parsing and producing of utterances, but also the grounding of internal representations through sensory processing, and, most importantly, steps for learning new aspects of language if necessary: new words, new meanings for existing words, new phrases. We believe that human–robot communication is best structured in terms of language games because human language interpretation requires a strong sense of context. The utterance does not contain all the information necessary for its interpretation. Words are ambiguous, many things are left unsaid, and the speech signal is notoriously difficult to decode. Because the language game makes the communication more predictable and provides a framework for semantic inference, it produces the strong constraints needed to make verbal communication doable and enables social learning [21].

The present paper focuses on one game only, namely a description game in which one agent (the speaker) describes to another agent (the hearer) an event in the world. The hearer gives feedback whether he agrees with the description or not. Some snapshots of a typical example of a scene is shown in Fig. 2. There is a hand which moves towards a (red) object and picks it up. An adequate description is: “The hand picks up the red object”. Notice that the background consists of an unaltered typical office environment with different sources of light (daylight and artificial light). The action takes place at a normal pace and the dialog takes place as a commentary on the actions in real time.

Additional typical events handled by the system are: the red ball rolls against the green block. The hand slides the pyramid against the blue box. The hand puts the red cube on top of the green one. A yellow ball rolls down a ramp. Because the world consists of dynamically changing situations, classified as events, this work is strongly related to other research on visual event classification [14,17], temporal world modelling [2], and the conceptual analysis of event expression in natural languages [23].

3.2. The semiotic cycle

To play a description game requires that the speaker perceives the situation by capturing streams of images with the camera, represent the result of sensory processing as a series of facts in memory, and then conceptualise the event and the objects in terms of roles and event types. Next the speaker must map this conceptualisation into an utterance, which includes choosing words for the predicates identifying the objects and the event, and applying the rules of grammar. The hearer must lookup the words and decode the grammatical structures, reconstruct a semantic structure, and interpret it in terms of his own world model. The language game succeeds if the utterance produced by the speaker describes an event in the recent past. The whole process is called the semiotic cycle (an extension and adaptation of the well-known ‘semiotic triangle’) and displayed in Fig. 3.

The internal conceptual representation consists of a series of facts represented in first-order predicate–calculus, following standard practices in symbolic AI [15]. A typical set of facts generated from visual processing for an event in which a red ball moves away from a green ball is:

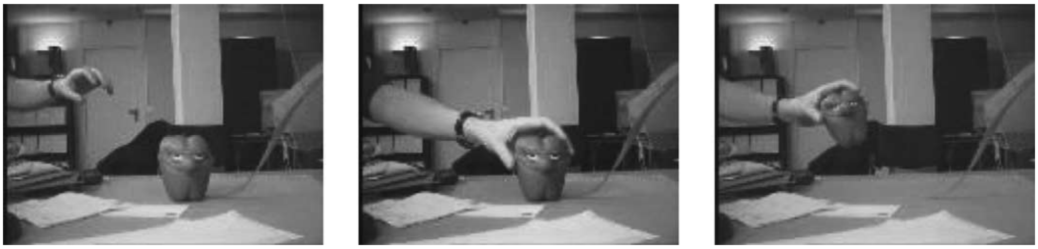


Fig. 2. Snapshots of a typical event handled in the experiment: a hand grasping an object.

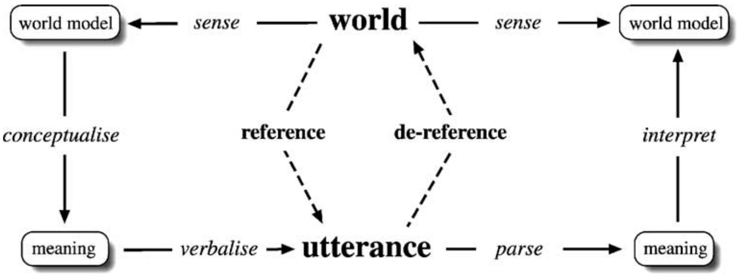


Fig. 3. The semiotic cycle: left, processes carried out by the speaker; right, processes carried out by the hearer.

```
(move-away ev-1) (move-away-patient
   ev-1 obj-1)
(move-away-source ev-1 obj-2)
  (green obj-2)
(ball obj-2) (hand obj-4) (red obj-1)
  (ball obj-1)
(larger obj-1 obj-2) (box obj-3)
  (next-to obj-1 obj-3)
```

Speaker and hearer see the situation from different angles, which often implies that there is no complete equivalence in their world models.

Each fact has three additional information items:

- A *time stamp* indicating when the fact arrived into memory. This is used for implementing forgetting: after a certain time period 'old' facts are erased and can no longer be the subject of a language communication.
- A *time period* which specifies the start and end point of a fact (when known). This makes it possible to use the temporal interval calculus [2] for representing and reasoning about actions and time.
- A *certainty* indication assigned by the vision system to this fact.

Before making an utterance, the speaker chooses (randomly) a recent event plus a set of objects related to this event (for example, ev-1, obj-1 and obj-2). For each object and for the event itself, the speaker then seeks a predicate or conjunctive combination of predicates, that are distinctive for the object or the event. Distinctive means that the predicate (or combination) is only valid for the intended object but not for any other object in the context. Thus (ball obj-3) is not distinctive for obj-3 in the above example, because both obj-1 and obj-2 are described as such. However, (green obj-3) is distinctive because obj-3 is the only green object in the context. The lexicon associates words with predicates (for example, the word "green" with the predicate 'green') and grammatical rules map additional aspects of meaning such as the predicate–argument relations into syntactic structures. The speaker assembles all of this into a complete utterance and the hearer uses the same rules in reverse to come up with a semantic structure.

The semantic structure as reconstructed by the hearer from parsing the utterance consists of a predicate–calculus expression with variables that can be matched against the facts in fact memory, again following standard practises in natural language semantics. For example, for the utterance "the red ball moves away", this expression looks as follows:

```
(move-away ?event) (move-away-
   patient ?event ?object)
(red ?object) (ball ?object)
```

When this expression is matched against the fact memory shown earlier, a unique coherent set of bindings of all variables is obtained:

```
((?event . ev-1) (?object . obj-1))
```

This is considered as an appropriate interpretation and therefore the game succeeds. The game fails when there is no such interpretation or when there is more than one set of possible bindings for all the variables.

This paper does not further discuss the (very complex) language component, nor how words or grammatical rules are invented and learned as part of the game (see [19] for more details). Instead, we focus on how agents establish the relation between the real world as captured by the cameras and their internal world models.

3.3. Issues in grounding

There are a number of very difficult grounding issues which need to be handled within the context of this application:

- First of all, the environment which generates input to the system consists of the dynamically changing unpredictable real world. Agents have to keep up with the dynamics of the environment and produce responses within available sensory and computational resources. It follows that not just anything can be computed but resources need to be allocated in a dynamic fashion depending on the requirements of the communicative situation.
- Second, because the images are unconstrained, with natural and changing daylight, the results of visual processing are necessarily going to be noisy. For example, segments found on the basis of colour segmentation may suddenly disappear or change when light conditions are slightly changing, a condition in the world (like a hand touching an object) may during a short instant of time change because of unstable segmentation, part of an event may not be perceived due to failures in lower level visual processing, etc. So we need a way to handle noise, for example by using top-down expectations.
- Third, because the communication is open-ended, it will be necessary to adapt the visual processing to the needs of the communication partners, which implies that at least some of it will have to be learned. Another thing which has to be learned is what the vision system 'tells' the language system.

4. Visual processing

We now focus on the set of visual interpretation processes that the agents use to relate the external dynamically changing real world with the internal conceptual world models and with the meaning of natural language expressions. Rather than detailing the many vision algorithms that have been used (which are most of the time well-known state of the art algorithms [16]), we focus on the general architectural principles. More information on the PERACT system can be found in [6].

The vision system can be decomposed into three subsystems. The first one attempts to detect and track visual units at different hierarchical levels. The second subsystem detects and tracks events, again at different hierarchical levels. The third subsystem consists of feature detectors that attempt to find qualitative descriptions for units at different levels of the object or event hierarchy. The result of all these processes is a set of streams, reporting objects and their properties dynamically in response to a changing world. There is a (short term) memory of these streams that is kept as they unfold. This is called the visual memory. Some of the descriptions flow automatically into the robot's fact memory (particularly those that are at a higher level and whose certainty is beyond a threshold) and these are used by the conceptualisation system to construct or interpret semantic structures.

4.1. Detecting and tracking spatio-temporal units

The first step in grounding is to detect and 'latch onto' regions in the image that are generated by objects of interest in the environment. This results in a deictic representation [1] which establishes and monitors indexical references between internal symbols and external objects. A first innovation of the work presented here is that this tracking not only takes place for a single object, but for an open-ended set of objects at different hierarchical levels—as long as they are part of the same spatio-temporal context. The detection and tracking of units at different hierarchical levels constitutes the backbone of the vision system. It starts in a bottom-up manner from the images captured by the camera, and goes through various processing steps, first to extract spatial regions, and then to connect them in time to get spatio-temporal continuities (see Fig. 4).

More concretely the following layers are present:

(1) *Image streams*: At the first bottom layer, there is an influx of images (at a rate of 24 s and with a size of 160 × 120 pixels) supplied by the camera in the LUV colour space.

(2) *Figure/ground separation*: The next step is to identify regions that may correspond to objects in the scene, thus distinguishing figure(s) from

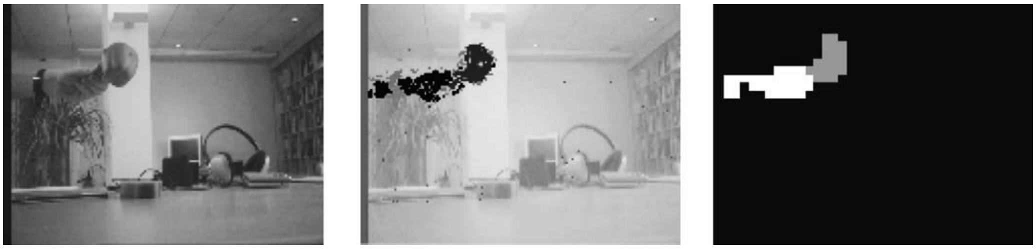


Fig. 4. Flow of treatment from raw image to segmented objects.

background. This is currently implemented by comparing the captured image with a stored image of the background. All zones where the pixels are not identical to the background are marked as zones of interest. This avoids processing the complete image in subsequent layers. The background needs to be learned prior to further visual processing but is updated whenever there are significant changes.

(3) *Occupancy grid*: Constructing an occupancy grid is a well known technique used in mobile robotics for navigation and path planning. An occupancy grid is a cellular representation of the environment that contains in each cell information about the probability that there is an object present [12]. We use a similar technique here. Prior to routine visual processing, probabilistic colour histograms are learned for each of the possible objects that may appear in a scene [7]. These histograms make it possible to calculate the probability that a certain pixel belongs to a particular object. The occupancy grid collects these probabilities for all pixels in each zone of interest and assigns the pixel to the object with the highest probability, if it is greater than a minimal threshold. Note that relying on colour histograms for object recognition clearly limits the types of scenes and object sets we can handle, but as our main goal is on exploring shared grounding, we do not seek absolute competence in vision.

(4) *Spatial region growing*: The resolution of the occupancy grid is then reduced (for efficiency) and next used by a region growing algorithm [24] to group the zones of interest into regions that correspond to objects. The result of this layer is therefore a stream of 'best' hypotheses for each object's coarse spatial occupancy.

(5) *Temporal tracking*: The first four layers all work on streams of single images. The next layers work on multiple images with the goal of tracking the same object over time. Because objects are identified using the histogramming technique, it is relatively easy to know whether they re-occur in the image and to compute their centre so that a trajectory can be established. No sophisticated region tracking (by trying to match parts of regions from one image to the next) is performed, which of course restricts the number of objects of the same colour that can be handled.

Obviously, this vision system has a number of clear limitations. The background has to be learned prior to visual processing and has to be updated when there is a significant change. The frame rate of the camera is low (24 images/s), so that fast actions (such as a ball being dropped onto the table) cannot be detected. Objects which can participate in scenes have to be learned in advance. There is only a small set of objects (typically seven depending on their colour histograms) that can be used simultaneously in the same set of scenes. Nevertheless, this system is adequate enough for our main purposes (namely experiments in grounded language communication). Work continues to improve its reliability and speed by integrating additional vision algorithms, but the strong temporal constraint imposed by the real world puts a limit on how much visual processing can be done with available hardware. Right now the system computes all visual information at a rate that keeps up with the frame rate of the camera.

4.2. Event detection and tracking

The next set of visual processes is concerned with the detection and tracking of events. The task is similar to that of detecting objects, in the sense that deictic representations are constructed and maintained. The main difference is that grouping is based on changes in the properties of objects rather than on invariances. Event detection is organised in different layers:

(1) *Detecting change*: The first layer produces a stream of properties of objects that change over time. Specifically, it produces qualitative descriptions for:

- Movement of an object, which is signalled if the centre of gravity of an object has changed significantly in between two image frames.
- Contact between two objects which is signalled if the regions of the objects concerned touch each other for a significant time period.
- Approach between objects which is triggered if the distance between the centres of gravity of two objects is becoming significantly smaller between image frames.
- Front positioning of one object x with respect to an object y which is signalled if x is moving towards y, and y is located within a cone emanating from x.

A stream of Boolean values for these descriptions are produced for all the objects in the scene, together with an indication of their certainty.

(2) *Detecting events*: The next layer groups these results in time. Moments when the same configuration of qualitative descriptors holds are grouped together in blocks. For the scenes shown in Fig. 2, two objects are being tracked: 0 (a hand) and 1 (a red object), and the following properties: the hand moves (M0), object-1 moves (M1), object-1 and the hand touch each other (T10), they approach each other (A10), object-1 moves in front of the hand (F10) or the hand moves in front of object-1 (F01). The description streams generated in conjunction with Fig. 2 is as follows, where the number in front of each line indicates the number of time steps that the same configuration of descriptions holds.

```
   [M0  M1  T10 A10 F10 F01]
13 [ 1           1         ]
12 [         1             ]
 7 [ 1   1   1           1 ]
 6 [         1             ]
```

Blocks of time in which the same configuration holds are called micro-events. For example, the first micro-event is one where the hand moves and approaches an object. The second micro-event is one where the hand is touching the object. In the third micro-event the hand and the object move, the hand still touches the object and the hand is moving fronted with respect to the object. In the final micro-event the hand still touches the object but neither the hand nor the object move. Micro-events are generated as soon as they have been found, in other words when the configuration of qualitative descriptors changes for a significantly long time.

(3) The final layer is concerned with the detection of events. Events are sequences of micro-events. For example, the pick-up event in Fig. 2, involving a hand and an object, is defined as a sequence of three micro-events: (1) the hand moves towards the object, (2) the hand touches the object, and (3) both move away together. Processes concerned with recovering such events use a library of event definitions which is matched against the stream of micro-events.

4.3. Qualitative descriptors

The final set of processes consists of pattern detection algorithms which detect significant features about the objects or events at different levels of the hierarchy. These algorithms look at the stream of units and compute properties of single objects (such as size, shape, colour, texture, etc.) or properties of multiple objects (such as respective geographical locations and change in locations). They output the result as streams of qualitative descriptions with a certainty indication. The algorithms use standard techniques from computational geometry and pattern recognition [6].

The qualitative descriptors are integrated in a flexible architecture that makes it possible to add new detectors at any time at any level of the object or event hierarchy or reschedule their usage, partly driven by top-down expectations. Concretely each descriptor runs as a separate parallel process (implemented as POSIX threads). Each process gets time-slices to advance its computation. Certain algorithms require more resources than others, and so ‘quick’ algorithms yield early results which can already be used at higher levels and may be sufficient for the purpose of language communication. Processes may be pre-empted when their results are no longer relevant.

4.4. Top-down information flow

In the discussion so far we assumed that information flows only in a bottom-up manner: from the images captured by the camera via a whole set of processes to the facts in memory. But this is a simplification that does not work for two reasons:

1. Each of the processing steps discussed may yield an unreliable result. For example, it is seldom the case that the qualitative descriptors which provide the basis for the detection of micro-events yields a clean set of outputs so that the micro-event is neatly defined as a block in time. Instead, the configuration is interspersed with very short moments when some of the descriptors do not hold. If we would only perform strict bottom-up processing we cannot deal with this kind of noise. Our solution has been to introduce for each pattern detector a top-down influence from the next level up. The user of the results of a pattern detectors monitors the certainty of recognition and the constancy of a pattern over time, so that small glitches can be eliminated and weak hypotheses discarded.
2. There is so much visual information in the image that it is impossible to extract fast enough everything that could possibly be extracted. Moreover, as little as possible should be put into the fact memory to avoid overloading or slowing down symbolic processing. However, occasionally more processing at lower levels is necessary: because an object being tracked on the basis of colour disappears temporarily from view, or because the continuation of an action which was taking place in fact does not take place, because the listener needs to use information about the shape of an object which was not yet computed by the vision system, etc. In these situations, it must be possible to assign more resources to the processes taking place at a specific layer and perform additional computation.

We have addressed these issues by introducing the notion of requests. Requests can be sent from the language system to the vision system, and the vision system can internally also generate requests. Requests trigger the activation or re-activation of pattern detectors on specific stretches of the object or event streams. They may also cause the reconfiguration of pattern detectors to change priorities and give more computer time to requested information. Finally, they can change the set of descriptors that is sent by default from the vision system to the fact memory.

Here are two concrete examples where this facility is used:

(1) In deciding what to say, the speaker must find a distinctive description to refer to an object. Suppose that there are two objects in memory, obj-1 and obj-2, and that facts in the fact memory only say that they are both red, perhaps because colour was the only property computed so far with sufficient certainty. The speaker cannot discriminate between the objects and so a request is issued to the vision system to stimulate computation of other qualitative descriptors for obj-1 and obj-2, that might yield a distinctive description. There are pattern detectors for colour, shape, texture, size, position, etc. with default priorities. Some of them may not have had enough resources to come up with a reliable conclusion, others may have such a low priority that they were not started at all. When the request comes, more computational power is given to these pattern detectors. Moreover, they do not necessarily operate over the image stream as it is entering the system but on past stretches as recorded in visual memory. When the pattern detectors produce more results, they are sent to the language system, turned into facts in memory and used in a new attempt for discrimination.

(2) The hearer may be sent an utterance that uses a set of properties which are not yet in the fact memory. For example, the hearer may have been asked to identify “the ball next to the green cube” but his fact memory may have recorded only that there is a green and a red object. Again a request is generated to the hearer’s vision system to go after more information. This request can be precise: compute information about shape for these specific objects given the hypothesis that it can be ball or a cube.

It would be desirable to enrich the power of the top-down flow of constraints on vision processing, for example, by predicting the position of objects in future time steps and use that as hypotheses for the pattern detectors, but this is not done yet in the current implementation.

5. Design principles for grounding

We now attempt to extract some of the lessons learned from our designs and experimentations with the grounded communication system briefly described in the previous section. We do not claim that these principles are unique to the system discussed here, on the contrary, we try to capture the 'best practice' in the field.

1. *Indexical representations*: The first important principle is to introduce a continuous detection and tracking of objects and events. The vision system described here latches onto an object or event and keeps tracking it as much as possible. This results in a dynamic deictic representation which maintains streams of indexical references, even if objects or events change.
2. *Description streams*: The second important principle is the introduction of description streams which produce and monitor properties of objects and events in time. The streams start from the images flowing in through the camera at a steady rate and continues all the way up to facts spilling into the fact memory. The units to which the descriptions apply are assembled, first spatially then temporally, at many different hierarchical levels.
3. *Noise reduction*: It is well known that real images taken from relatively unprepared real world situations always yield noisy processing results. This motivates the next design principle: noise at one hierarchical level can be reduced by preferring the most coherent analysis at the level just above it. We apply this principle through the whole system and at all levels.
4. *Top-down information flow*: It is clearly not enough to have information flowing from the sensory data to the conceptual world model, partly because there is not enough time to compute everything that could be computed. So there must also be a steady top-down flow of requests and expectations from the 'cognitive' layers to sensory processing.
5. *Attention*: Finally, we believe that an attention mechanism is unavoidable. The attention mechanism is responsible for allocating scarce computation resources. The system discussed in this paper uses a variety of means to achieve this: figure/ground computation at a very early stage, a thread-based implementation of feature detectors with varying and dynamically modifiable priorities partly steered by the vision system.

6. Conclusions

Grounded robots that engage in communication using external representations not only need a physical body and low-level behaviours but also a conceptual world model which must be anchored firmly and dynamically by the robot in the environment through its sensori-motor apparatus. We argued that there is not a simple sweeping theoretical principle to turn a system that uses conceptual world models into a grounded system. Instead many processes must be carefully integrated. We described an implemented system that has attempted to do so in the context of experiments in grounded open-ended language communication among robots as well as between humans and robots. We also proposed a set of design principles that capture the principles that we have used in our design.

Acknowledgements

This research was done at the Sony Computer Science Laboratory in Paris. We are grateful to several members of the laboratory for feedback and comments on the work discussed here. The research is also part of a French CNRS OHLL research project on the Origins of Language.

References

[1] P. Agre, D. Chapman, Pengi: An implementation of a theory of activity, in: Proceedings of the Sixth National Conference on Artificial Intelligence, AAAI Press, Anaheim, CA, 1987, pp. 268–272.

[2] J.F. Allen, G. Ferguson, Actions and events in interval temporal logic, Journal of Logic and Computation 4 (5) (1994) 531–579.

[3] I. Androutsopoulos, G.D. Ritchie, P. Thanisch, Natural Language Interfaces to Databases—An Introduction,

Cambridge University Press, Cambridge, Journal of Natural Language Engineering 1 (1) (1995).

[4] N. Badler, M. Palmer, R. Bindiganavale, Animation control for real-time virtual humans, Communications of the ACM 42 (8) (1999) 64–73.

[5] J.-C. Baillie, J.-G. Ganascia, Action categorization from video sequences, in: W. Horn (Ed.), Proceedings of the ECAI 2000, IOS Press, Amsterdam, 2000.

[6] J.-C. Baillie, Apprentissage et reconnaissance d'actions dans des sequences video, Ph.D. Thesis, Universite de Paris.

[7] J.-C. Baillie, Object tracking using certainty color maps, in: Proceedings of the ECCV'2002, submitted for publication.

[8] R.A. Brooks, Intelligence Without Representation: Artificial Intelligence Journal 47 (1991) 139–159.

[9] H.H. Clark, S.A. Brennan, Grounding in communication, in: L.B. Resnick, J.M. Levine, S.D. Teasley (Eds.), Perspectives on Socially Shared Cognition, APA Books, Washington, 1991.

[10] P. Cohen, et al., Proceedings of the AAAI Spring Symposium on Grounding, AAAI Press, Anaheim, CA, 2001.

[11] S. Coradeschi, D. Driankov, L. Karlsson, A. Saffiotti, Fuzzy anchoring, in: Proceedings of the IEEE International Conference on Fuzzy Systems, Melbourne, Australia, 2001.

[12] A. Elfes, Using occupancy grids for mobile robot perception and navigation, Computer 22 (6) (1989) 46–57.

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

[14] Y. Kuniyoshi, H. Inoue, Qualitative recognition of ongoing human action sequences, in: Proc. IJCAI-93, 1993, pp. 1600–1609.

[15] N. Nilsson, Artificial Intelligence: A New Synthesis, Morgan Kaufmann, Los Altos, CA, March 1998.

[16] L. Shapiro, G. Stockman, Computer Vision, Prentice-Hall, Englewood Cliffs, NJ, 2001.

[17] J. Siskind, Visual event classification through force dynamics, in: Proceedings of the AAAI Conference 2000, AAAI Press, Anaheim CA, 2000, pp. 159–155.

[18] D. Sperber, D. Wilson, Relevance: Communication and Cognition, Harvard University Press, Cambridge, MA, 1986.

[19] L. Steels, Language games for autonomous robots, IEEE Intelligent Systems (2001) 16–22.

[20] L. Steels, Simulating the origins and evolution of a grammar of case, in: Proceedings of the Evolution of Language Conference, Harvard, April 2002, in press.

[21] L. Steels, F. Kaplan, AIBO's first words, The social learning of language and meaning, Evolution of Communication 4 (1) (2001).

[22] L. Steels, F. Kaplan, A. McIntyre, J. Van Looveren, Crucial factors in the origins of word-meaning, in: A. Wray (Ed.), The Transition to Language, Oxford University Press, Oxford, UK, 2002.

[23] L. Talmy, Toward a Cognitive Semantics: Concept Structuring Systems (Language, Speech, and Communication), The MIT Press, Cambridge, MA, 2000.

[24] M.G. Montoya, C. Gil, I. Garcia, Implementation of a region growing algorithm on multicomputers: analysis of the work load balance, in: Proceedings of the AI'2000, Canada, 2000.

Luc Steels is professor of Artificial Intelligence at the University of Brussels (VUB) and director of the Sony Computer Science Laboratory in Paris. He is one of the pioneers of artificial life approaches to robotics, particularly in the domain of behavior-based architectures. More recently he has developed evolutionary approaches to language, in which robots evolve communication systems through self-organisation and emergence. He published a dozen books in several areas of Artificial Intelligence.

Jean-Christophe Baillie studied physics and computer science at the Ecole Polytechnique, France, and holds a Ph.D. thesis on Artificial Vision Computer Science from the Computer Science Laboratory of Paris 6. He conducted the research for this paper at the Sony Computer Science Laboratory in Paris. At the moment he is Associate Professor at ENSTA (Ecole Nationale Supérieure des Techniques Avancées).

Robotics and Autonomous Systems 43 (2003) 175–188

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Anchoring symbols to conceptual spaces: the case of dynamic scenarios

A. Chella [a,c,*], M. Frixione [b,c], S. Gaglio [a,c]

[a] *Dipartimento di Ingegneria Automatica Informatica, Università di Palermo, Viale delle Scienze, Palermo 90128, Italy*
[b] *Dipartimento di Scienze della Comunicazione, Università di Salerno, Salerno, Italy*
[c] *ICAR-CNR, Palermo, Italy*

Abstract

This paper deals with the anchoring of one of the most influential symbolic formalisms used in cognitive robotics, namely the *situation calculus*, to a conceptual representation of dynamic scenarios. Our proposal is developed with reference to a cognitive architecture for robot vision. An experimental setup is presented, aimed at obtaining *intelligent monitoring* operations of a robotic finger starting from visual data.

Keywords: Anchoring; Robot vision; Conceptual spaces; Action representation; Situation calculus

1. Introduction

A cognitive architecture for robot vision has been proposed by the authors in [4–6]; it is aimed at the representation of knowledge extracted from visual data in both static and dynamic scenarios. One of the main assumptions underlying the design of this architecture is the need of a principled integration of the approaches developed within the artificial vision community and the symbolic, propositional systems developed within symbolic knowledge representation (KR) in AI. Such an integration is based on the introduction of a *conceptual level* of representation, intermediate between the processing of visual data and declarative, propositional representations.

This paper deals with the anchoring of one of the most influential symbolic formalisms adopted in cognitive robotics, namely the *situation calculus*, to the conceptual representation of dynamic scenes. We discuss in particular how *actions*, *situations* and *fluents* may be anchored (in the sense of anchoring developed by Coradeschi and Saffiotti [9,11]) to the representations at the conceptual level, which are in turns generated starting from the robot perceptions (for an up to date survey on different perspectives on anchoring see [10]).

The main motivation for choosing the situation calculus lies in the fact that it is one of the simplest, more powerful and best known logic formalisms for the representation of knowledge about actions and change. It was primarily developed by McCarthy and Hayes [22]; for up to date and exhaustive introductions see [28,27]. Nowadays, it is a widely adopted formalism in the *cognitive robotics* literature; efficient Prolog implementations have been proposed [12,13,21]; simplified versions of the situation calculus are used by working mobile robots [3,14,18].

The following discussion is based on an experimental setup aimed at obtaining an intelligent visual control of a robotic finger starting from visual data. The

[*] Corresponding author.
E-mail addresses: chella@unipa.it (A. Chella), frix@dist.unige.it (M. Frixione), gaglio@unipa.it (S. Gaglio).

0921-8890/03/$ – see front matter © 2003 Elsevier Science B.V. All rights reserved.
PII: S0921-8890(02)00358-5

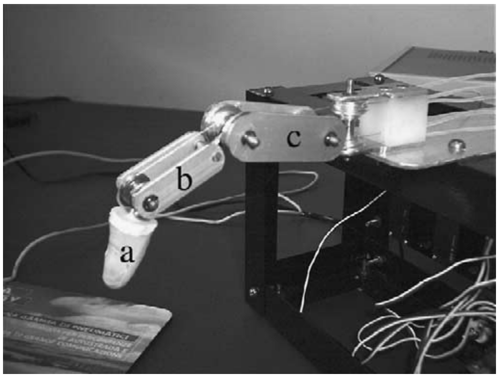


Fig. 1. The robotic finger used in the experimental setup. The terminal phalanx *a*, the middle phalanx *b* and the upper phalanx *c* are shown.

finger has been entirely developed at the Robotics Laboratory, Department of Computer Engineering, University of Palermo. It is made up by three phalanxes: a terminal phalanx *a*, a middle phalanx *b* and an upper phalanx *c* (see Fig. 1).

The finger is driven by schematic behaviors [1], and performs articulated movements, such as pushing a ball (Fig. 2) or picking up torus-shaped objects (Fig. 3). The system is equipped with a video camera that acquires the movements of the objects and of the finger itself, in order to perform *intelligent monitoring* operations. The acquired visual data are anchored to symbolic descriptions of the finger operations.

The system takes in input a sequence of images corresponding to subsequent phases of the evolution of the scene (the movements of the robotic finger and their effects on the whole scene), and produces in output a declarative description of the scene, formulated as a set of assertions written in the formalism of the situation calculus.

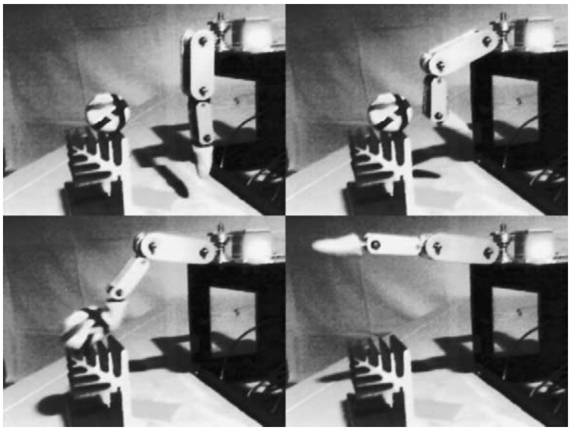


Fig. 2. The robotic finger pushes a ball.

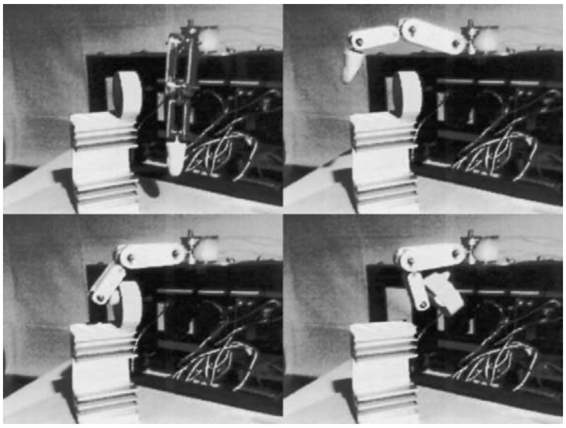


Fig. 3. The robotic finger picks up a simple torus-shaped object.

Such a symbolic description may be employed to perform high-level inferences, e.g. those needed to generate complex long-range plans, or to perform causal and diagnostic reasoning about the system operations. Symbolic assertions may also be used to generate explanations of the operations of the finger, in order to perform high-level teleautonomy [8].

The paper is organized as follows. In the next section, the main assumptions underlying the cognitive architecture are summarized. The third section is devoted to a synthetic description of the conceptual level representation of motion. The fourth section shows in details how the situation calculus is anchored to the conceptual representation. The last section discusses the proposed framework, and compares it to some relevant frameworks for anchoring described in the literature. Short conclusions follow.

2. The cognitive architecture for visual perception: an overall view

The existing attempts to integrate visual perception with propositional KR are mostly oriented towards natural language interpretation, with particular emphasis on man–machine interaction. They face only in a

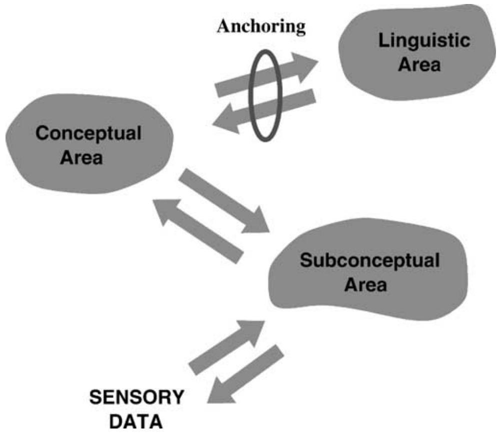


Fig. 4. The three areas of representation and the relations among them.

marginal way the general aspects of knowledge representation (see [6] for a review).

Our proposal is based on the hypothesis that a principled integration of the approaches of artificial vision and of symbolic KR requires the introduction of an intermediate representation between these two levels. Such a role is played by a *conceptual space*, according to the approach proposed by Peter Gärdenfors [15]. In our architecture, this intermediate, conceptual representation is the place where the anchoring occurs, and where the anchoring procedures operate.

The architecture is organized in three *computational areas*. Fig. 4 schematically shows the relations among them. The *subconceptual* area is concerned with the low-level processing of perceptual data coming from the sensors. We call it "subconceptual" because here information is not yet organized in terms of conceptual structures and categories. The subconceptual area includes a 3D model of the perceived scenes. Even if such a kind of representation cannot be considered "low-level" from the point of view of artificial vision, it still remains below the level of conceptual categorization.

In the *linguistic* area, representation and processing are based on the formalism of the situation calculus. In the *conceptual* area, the data coming from the subconceptual area are organized in conceptual categories, which are still independent from any linguistic characterization. The symbols in the linguistic area are anchored to sensory data by mapping them on the representations in the conceptual area. The purpose of the subsequent discussion is to show how a conceptual representation can be used to anchor the sentential representations of the situation calculus to the perceptual activities of a robotic system in a theoretically well founded way.

3. Conceptual spaces for representing motion

As previously stated, representations in the conceptual area are couched in terms of a *conceptual space* [15] that provides a principled way for relating high level, linguistic formalisms on the one hand, with low level, unstructured representation of data on the other. In this sense, we claim that conceptual spaces are a valuable tool for anchoring [7]. A conceptual space CS is a metric space whose dimensions are in some way related to the quantities processed in the subconceptual area. Dimensions do not depend on any specific linguistic description. In this sense, a conceptual space comes before any symbolic-propositional characterization of cognitive phenomena. In particular, a conceptual space devoted to the representation of the motion of geometric shapes is taken into account in the present paper.

3.1. Dynamic conceptual space

The term *knoxel* denotes a point in a conceptual space. From the mathematical point of view, a knoxel **k** is a vector in CS; from the conceptual point of view, it is an epistemologically simple element at the considered level of analysis. In the case of static scenes [4], a knoxel coincides with a 3D primitive shape, described in terms of some constructive solid geometry (CSG) schema. For example, the robotic finger (Fig. 1) may be described by three knoxels, corresponding respectively to the terminal phalanx *a*, the middle phalanx *b* and the upper phalanx *c*.

In order to represent dynamic scenes, we adopted an intrinsically *dynamic conceptual space*. The main assumption behind such a dymamic CS is that simple motions are categorized in their wholeness, and not as sequences of static frames. According to this hypothesis, every knoxel corresponds to a simple motion of a 3D primitive.

Formally, a knoxel $\mathbf{k}$ can be decomposed in a set of components $x_i(t)$, each of them associated with a degree of freedom of the moving primitive shape. In other words:

$$\mathbf{k} = [x_1(t), x_2(t), \dots, x_n(t)], \tag{1}$$

where n is the number of degrees of freedom of the moving 3D primitive (e.g. a phalanx of the finger). In turn, each motion $x_i(t)$ may be considered as the result of the superimposition of a set of elementary motions $f_j^i(t)$:

$$x_i(t) = \sum_j X_j^i f_j^i(t). \tag{2}$$

In this way, it is possible to choose a set of basis functions $f_j^i(t)$, in terms of which any simple motion can be expressed. Such functions can be associated to the axes of the dynamic conceptual space as its dimensions. Therefore, from the mathematical point of view, the resulting CS is a *functional* space.

In the domain under investigation, the chosen set of basis functions are the first low frequency harmonics, according to the well-known Discrete Fourier Transform (DFT, see [25]). By a suitable composition of the trigonometric functions of all of the geometric parameters, the overall motion of a 3D primitive is represented as a point in the functional space.

A single knoxel in CS therefore describes a *simple motion*, i.e. the motion of a primitive shape. A *composite simple motion* is a motion of a composite object (i.e. an object approximated by more than one primitive shapes, as is the case of the robot finger). A composite simple motion is represented in the CS by the set of knoxels corresponding to the motions of its components. For example, the first part of the trajectory of the whole finger shown in Fig. 3 is represented as a composite motion made up by the knoxels $\mathbf{k_a}$ (the motion of the terminal phalanx a), $\mathbf{k_b}$ (the motion of the middle phalanx b) and $\mathbf{k_c}$ (the motion of the upper phalanx c). Note that in composite simple motions the (simple) motions of their components occur simultaneously. In this case, the configuration of the conceptual space is completely described by the three knoxels participating to the motion of the finger:

$$\text{CS} = \{\mathbf{k_a}, \mathbf{k_b}, \mathbf{k_c}\}. \tag{3}$$

To consider the composition of several (simple or composite), motions arranged according to some temporal relation (e.g. a sequence), the notion of *structured process* is introduced. A structured process corresponds to a series of different configurations of knoxels in the conceptual space. In the transition between two subsequent different configurations, there is a change of at least one of the knoxels in the CS which is the consequence of a change in the motion of the corresponding 3D primitives. We call “scattering” such a transition from one knoxel to another. It corresponds to a discontinuity in time, and is associated with an instantaneous event.

In the case of the finger, a scattering occurs, e.g. when the finger has reached its upmost position, and begins to move downwards to pick up the object. In the CS representation, this amounts to say that knoxel $\mathbf{k_a}$ (i.e. the upward motion of the terminal phalanx) is replaced by knoxel $\mathbf{k}'_\mathbf{a}$, and, similarly, knoxels $\mathbf{k_b}$ and $\mathbf{k_c}$ are replaced by $\mathbf{k}'_\mathbf{b}$ and $\mathbf{k}'_\mathbf{c}$. The new CS′ configuration is

$$\text{CS}' = \{\mathbf{k}'_\mathbf{a}, \mathbf{k}'_\mathbf{b}, \mathbf{k}'_\mathbf{c}\}. \tag{4}$$

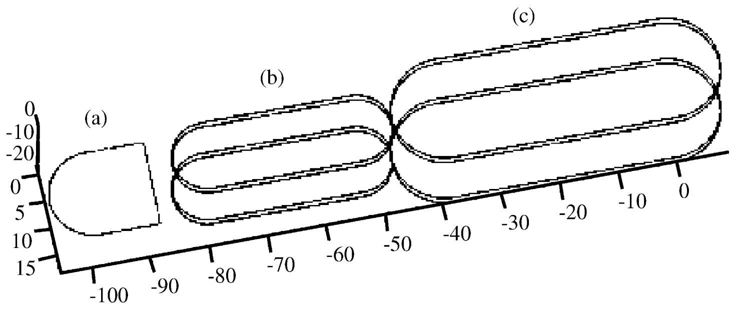


Fig. 5. The CAD model of the robot finger.

The occurred scattering may be described by the ordered set of the two CS configurations, before and after the scattering:

$$(\mathrm{CS}, \mathrm{CS}') \equiv (\{\mathbf{k_a}, \mathbf{k_b}, \mathbf{k_c}\}, \{\mathbf{k}'_\mathbf{a}, \mathbf{k}'_\mathbf{b}, \mathbf{k}'_\mathbf{c}\}). \tag{5}$$

3.2. Extraction of knoxels from image sequences

In the current experimental setup, the role of the *subsymbolic* area is the extraction of the knoxel parameters describing the 3D motion of the finger parts. This operation is based on an a priori 3D CAD model of the finger (Fig. 5).

In order to extract the finger contours, the images acquired by the camera are processed by an algorithm based on *snakes* [2]. A snake is a curve that searches the image under the influence of forces driven by the local distribution of the gray levels. Briefly, when the snake reaches the contour of an object, it is attracted by the contour itself, and it adapts its shape to the shape of the object. When the object moves or changes its shape, the snake continues to adapt itself in order to track the object.

Formally, a snake is described in a parametric form by the following equation:

$$v(s) = (x(s), y(s)), \tag{6}$$

where $x(s)$ and $y(s)$ are the coordinates along the shape contour and s the normalized arc length:

$$s \in [0, 1]. \tag{7}$$

The snake model adopted for this example reflects the geometric constraints imposed by the 3D model. The energy E_{snake} of a contour is defined as

$$E_{\mathrm{snake}}(v(s)) = \int_0^1 (E_{\mathrm{int}}(v(s)) + E_{\mathrm{image}}(v(s))\,\mathrm{d}s. \tag{8}$$

The energy integral is a functional since the variable s is in its turn a function (the shape contour). The internal energy E_{int} is formed from a Tikhonov stabilizer and is defined by

$$E_{\mathrm{int}}(v(s)) = a(s)\left|\frac{\mathrm{d}v(s)^2}{\mathrm{d}s^2}\right| + b(s)\left|\frac{\mathrm{d}v(s)^2}{\mathrm{d}s^2}\right|^2, \tag{9}$$

where $|\cdot|$ is the Euclidean norm.

The first-order continuity term, weighted by $a(s)$, let the snake behave elastically. The second-order curvature term, weighted by $b(s)$, let the snake be resistant to bending. For example, if we set $b(s) = 0$ at point s, the snake becomes second-order discontinuous at that point, and generates a corner.

The image functional determines which features have a low image energy, and hence attract the contours. In general, this functional is made up by three terms:

$$E_{\mathrm{image}} = w_{\mathrm{line}}T_{\mathrm{line}} + w_{\mathrm{edge}}E_{\mathrm{edge}} + w_{\mathrm{term}}E_{\mathrm{term}}, \tag{10}$$

where the w's are constant weights. The three terms respectively correspond to lines, edges and terminations. In the version of the model adopted for this example, only the edge functional is present, which attracts the snake to points with a high edge gradient:

$$E_{\mathrm{image}} = E_{\mathrm{edge}} = -(G_\sigma * \nabla^2 I(x, y))^2. \tag{11}$$

This corresponds to the image functional proposed by Kass et al. [19]. It is a scale-based edge operator that increases the locus of attraction of energy minimum. G_σ is a Gaussian of standard deviation sigma which controls the smoothing process prior to edge operator. Minima of E_{edge} lies on zero-crossing of $G_\sigma * \nabla^2 I(x, y)$ which defines the edges.

In order to extract the Regions of Interest (ROI) of the scenes before the application of the snake algorithm, some standard filtering operations are performed (see Fig. 6): starting from the acquired image (a), the noise is reduced by a 5 × 5 median filter (b), then the moving parts are detected by means of the Canny algorithm (c) and the image intensities between frames are subtracted in order to individuate the ROIs (d).

Fig. 7 shows the snake being attracted by the upper phalanx. As a first step, the snake initialize its position and dimensions by individuating the finger contours; then, it tracks the position of the finger during the evolution of the scene (Fig. 8). The geometric information obtained in this way is sent to a 3D CAD system that generates a VRML animated model of the evolution of the finger operations (Fig. 9).

Finally, the data concerning the movement of each phalanx are sent to a software module that performs the DFT, in order to generate the knoxel configuration of the conceptual space.

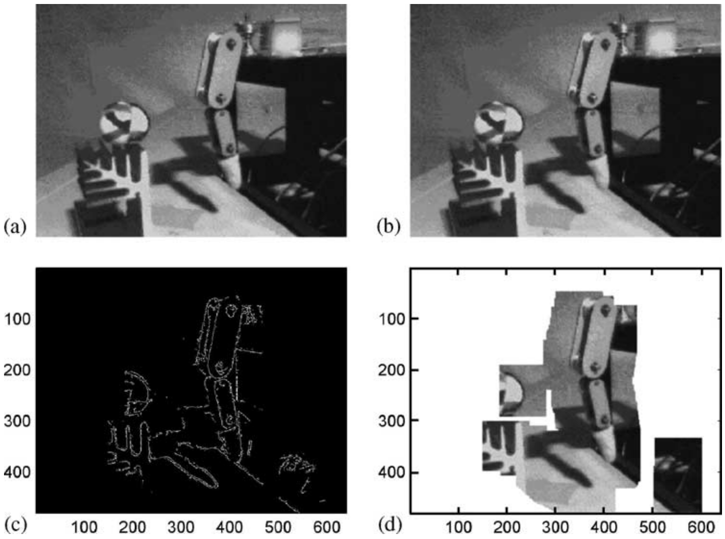


Fig. 6. The filtering operation to individuate the ROI of the finger.

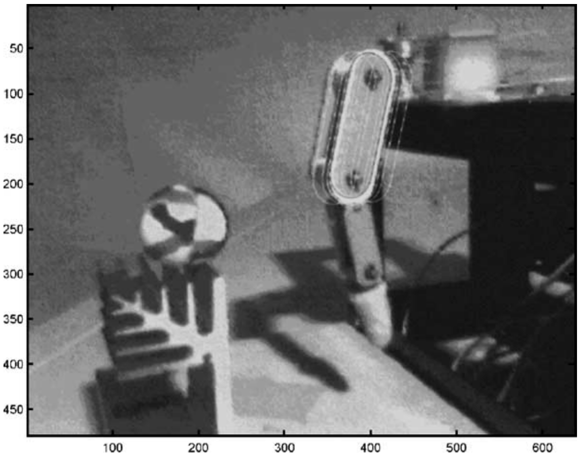


Fig. 7. The snake attracted by the upper phalanx.

4. Anchoring situation calculus to conceptual spaces

In the linguistic area, the evolution of the conceptual space is represented in terms of logic assertions expressed in the *situation calculus* formalism. Indeed, the representation adopted by the situation calculus is in many respects homogeneous to the conceptual representation described in the previous section.

In order to anchor linguistic area expressions to structures in the conceptual space, an *anchoring function Φ* associates expressions of the situation calculus to their counterpart in the conceptual space.

4.1. Anchoring actions and situations

The basic idea behind the situation calculus is that the evolution of a state of affairs is modeled in terms of a sequence of situations. The world changes when some *action* is performed. So, given a certain situation S_1, performing a certain action a will result in a new situation S_2. Actions are the sole sources of change of the world: if the situation of the world changes from, say, S_{i-1} to S_i, then some action has been performed. The initial situation S_0 models the initial state of the domain under consideration.

The situation calculus is based on the language of predicate logic. Situations and actions are denoted by first-order terms. The two place function *do* takes as its arguments an action and a situation: $S_i = do(a, S_{i-1})$ denotes the new situation S_i obtained by performing the action a in the situation S_{i-1}.

Classes of actions can be represented as functions. For example, the one argument function symbol $pick\_up(x)$ could be assumed to denote the class of the actions consisting in picking up some object x.

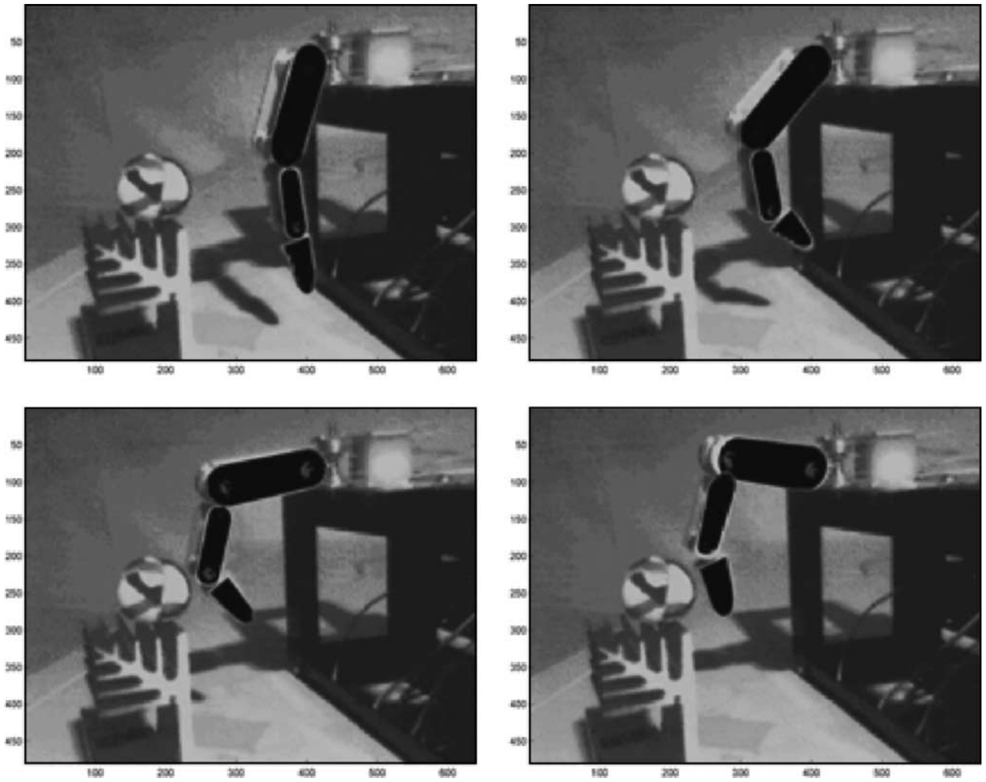


Fig. 8. The snake tracking the motion of the finger.

Given a first-order term o denoting a specific object, the term *pick_up*(o) denotes the specific action consisting in picking up o.

In terms of conceptual spaces, an action a is mapped on a suitable scattering of knoxels, corresponding to an ordered pair $(\mathrm{CS}_{i-1}, \mathrm{CS}_i)$, where CS_{i-1} and CS_i are the configurations of the knoxels, respectively, before and after the scattering:

$$\Phi(a) = (\mathrm{CS}_{i-1}, \mathrm{CS}_i). \tag{12}$$

The initial situation S_0 corresponds to the initial configurations of knoxels CS_0 in the conceptual space:

$$\Phi(S_0) = (\mathrm{CS}_0). \tag{13}$$

According to the situation calculus, a situation fully describes the state of affairs of the domain under consideration. Different sequences of actions lead to different situations. In other words, it can never be the case that performing some action starting form different situations can result in the same situation. If two situations derive from different situations, they are in their turn different, in spite of their similarity.

Therefore, a generic situation S_i is individuated by the unique sequence of actions $(a_0, a_1, \ldots, a_{n-1}, a_n)$ that generates the corresponding sequence of situations starting form the initial situation S_0. As a consequence, S_i is anchored to the sequence of knoxel configurations generated by the sequence of scattering corresponding to the actions:

$$\Phi(S_i) = (\mathrm{CS}_0, \mathrm{CS}_1, \ldots, \mathrm{CS}_{i-1}, \mathrm{CS}_i). \tag{14}$$

It should be noted that the formula $S_i = do(a, S_{i-1})$ means that the action a generates the new situation $(\mathrm{CS}_0, \mathrm{CS}_1, \ldots, \mathrm{CS}_{i-1}, \mathrm{CS}_i)$ starting from the old one $(\mathrm{CS}_0, \mathrm{CS}_1, \ldots, \mathrm{CS}_{i-1})$.

As an example, consider again the finger scenario. Suppose that the terminal phalanx of the finger initially rests in the position p_1. The initial situation S_0 is anchored to the configuration $\mathrm{CS}_0 = \{\mathbf{k_a}, \mathbf{k_b}, \mathbf{k_c}\}$,

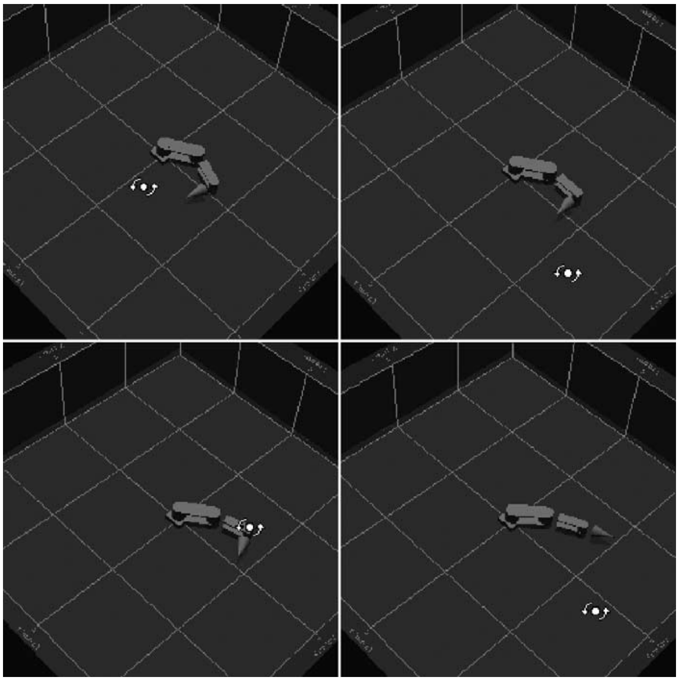


Fig. 9. The VRML model of the motion of the finger.

where $\mathbf{k_a}$ corresponds the terminal phalanx, and $\mathbf{k_b}$, $\mathbf{k_c}$ to the other to two phalanxes, all in rest state. Now the terminal phalanx begins to move from position p_1 towards position p_2 (suppose that the two other phalanxes do not change their position). When the motion of the phalanx starts, a scattering occurs in the conceptual space, and the knoxel $\mathbf{k_a}$ changes its position. Therefore the new configuration of knoxels is $\mathrm{CS}_1 = \{\mathbf{k}'_\mathbf{a}, \mathbf{k_b}, \mathbf{k_c}\}$ and the new situation S_1 is anchored to the sequence $(\mathrm{CS}_0, \mathrm{CS}_1)$.

Such a scattering corresponds to an (instantaneous) action that is represented by the formula *start_move_terminal_phalanx*(tph_1, p_1, p_2) (where tph_1 is the individual constant denoting the terminal phalanx). The knoxel $\mathbf{k}'_\mathbf{a}$ corresponds to the motion of the phalanx. During all the time in which the phalanx remains in such a motion state, the CS_1 remains unchanged (provided that nothing else is happening in the considered scenario), and $\mathbf{k}'_\mathbf{a}$ continues to be active in it.

When the motion of the phalanx ends in the position p_2, a further scattering occurs, $\mathbf{k}'_\mathbf{a}$ disappears, and a new knoxel $\mathbf{k}''_\mathbf{a}$ becomes active. Therefore, a new configuration $\mathrm{CS}_2 = \{\mathbf{k}''_\mathbf{a}, \mathbf{k_b}, \mathbf{k_c}\}$ is generated. This second scattering $(\mathrm{CS}_1, \mathrm{CS}_2)$ corresponds to the instantaneous action *end_move_terminal_phalanx*(tph_1, p_1, p_2). The knoxel $\mathbf{k}''_\mathbf{a}$ corresponds again to a rest state of the phalanx, but now in the position p_2. The new situation S_2 is now anchored to the configuration $(\mathrm{CS}_0, \mathrm{CS}_1, \mathrm{CS}_2)$.

4.2. Anchoring fluents

As a state of affairs evolves, it can happen that properties and relations change their values. In the situation calculus, properties and relations that can change their truth value from one situation to another are called (relational) *fluents*. An example of fluent could be the property of being in motion: it can happen that it is true that a certain object is in motion in a certain situation, and it becomes false in another. Fluents are denoted by predicate symbols that take a situation as their last argument. For example, the fluent corresponding to the property of being in motion can be represented as a two place relation *in_motion*(x, s), where *in_motion*(o, S_1) is true if the object o is in motion in the situation S_1.

Given a situation, a relational fluent f is, in general, anchored to a set of (sets of) knoxels. For example, the fluent *in_motion*(x, s) is anchored to the set of knoxels that correspond to moving shapes in the CS configurations that correspond to a situation s. A fluent *approaching*(x, y, s) is anchored to the set of ordered pairs of knoxels that represent pairs of shapes approaching each other in the CS configurations that correspond to a situation s. And so on.

In the finger example, a fluent *moving_terminal_phalanx*(x, p_1, p_2, s) is anchored to the set of knoxels moving from a point p_1 to a point p_2 in the CS configurations that correspond to a situation s.

In general, the anchoring function Φ for fluents behaves as follows:

$$\Phi(f(\bar{x}, s)) = \{\mathbf{kt_1}, \mathbf{kt_2}, \ldots, \mathbf{kt_n}\}, \tag{15}$$

where f is a fluent, $\bar{x}$ are all the arguments of f except for the last, and $\mathbf{kt_i}$ are all the knoxels, or knonxel t-uples, that satisfy the fluent f in the situation s.

The geometric structure of conceptual spaces, and the fact that the distance between points in a conceptual space can be interpreted as a measure of their similarity [15] make it possible to account for prototypical effects between the instances of a fluent: given a fluent f, and provided that the sets to which f is anchored in the different situations correspond to *natural concepts* [7], more central points correspond to “prototypical”, or “better” instances of f. Many forms of inference can take advantage from this feature of the conceptual representation.

4.3. Anchoring actions with temporal duration

In the ordinary discourse, actions may have a temporal duration. For example, the action of moving from a certain spatial location to another takes some time. In the situation calculus, all actions in the strict sense are assumed to be instantaneous. Actions that have a duration may be represented as processes, that are initiated and are terminated by instantaneous actions (see [26; 27, Chapter 7]). Suppose to represent the action of moving the robot finger f_1 from point p_1 to point p_2. In the terminology of the situation calculus, this is a process that is initiated by an instantaneous action, say *start_move_finger*(f_1, p_1, p_2), and is terminated by another instantaneous action, say *end_move_finger*(f_1, p_1, p_2). Processes correspond to relational fluents. For example, the process of moving the finger from p_1 to p_2 corresponds to the fluent *moving_finger*(f_1, p_1, p_2, s). A formula like *moving_finger*(f_1, p_1, p_2, S_1) is true if in situation S_1 the finger f_1 is moving from position p_1 to position p_2. The anchoring of processes immediately follows from the anchoring of actions and fluents without particular modifications of the Φ function.

4.4. Anchoring concurrent actions

Traditional situation calculus does not allow to account for concurrency. Actions are assumed to occur sequentially, and it is not possible to represent several instantaneous actions occurring at the same instant. In the considered setup, this limitations is too severe. When a scattering occurs in a CS it may happen that more knoxels are involved. This is tantamount to say that several instantaneous actions occur concurrently. This is the case, e.g. of the motion of the finger described in the previous paragraph. The trajectory of the whole finger can be represented as a composite motion made up by three knoxels: $\mathbf{k_a}$ (the motion of the terminal phalanx), $\mathbf{k_b}$ (the motion of the middle phalanx) and $\mathbf{k_c}$ (the motion of the upper phalanx). More in general, according to this terminology, *composite simple motions* are motions of composite objects. A composite simple motion corresponds in a CS to the set of the knoxels corresponding to the motions of its components. The beginning and the end of a composite simple motion always involve the scattering of more than one knoxel. Therefore, composite simple motions always entail some form of concurrency.

Suppose to represent within the situation calculus the whole motion of the finger. According to what stated before, moving the finger is represented as a process, that is started by a certain action, say *start_move_finger*, and that is terminated by another action, say *end_move_finger*. (For sake of brevity, here we omit the arguments of the actions.) The process of moving the finger is represented as a fluent *moving_finger*(s), that is true if in the situation s the finger is moving. The scattering in the CS corresponding to both *start_move_finger* and *end_move_finger* involve three knoxels, namely $\mathbf{k_a}$, $\mathbf{k_b}$ and $\mathbf{k_c}$, that correspond, respectively, to the motions of the phalanxes. Consider e.g. *start_move_finger*. It is composed by three concurrent actions, say *start_move_terminal_phalanx*, *start_move_middle_phalanx* and *start_move_upper_phalanx*, each of them corresponding to the scattering of one knoxel in the CS (resp. $\mathbf{k_a}$, $\mathbf{k_b}$ and $\mathbf{k_c}$).

Extensions of the situation calculus that allow for a treatment of concurrency have been proposed in the literature [16,23,26,28]. We adopt the approach developed in [16] and [26], according to which a two argument function $+$ is added to the language. Given two actions as its arguments, $+$, produces an action as its result. In particular, if a_1 and a_2 are two actions, $a_1 + a_2$ denotes the action of performing a_1 and a_2 concurrently. According to this approach, an action is *primitive* if it is not the result of performing other actions concurrently. If a is a complex action such that $a = a_1 + a_2 + \cdots + a_n$, then we write that $a_i \in a$ for each i such that $1 \leq i \leq n$.

In our approach, primitive actions correspond to the scattering of a single knoxel in the CS; the contemporary scattering of several knoxels corresponds to a complex action resulting from concurrently performing different primitive actions. For example, the motion of the whole finger can be represented by defining two non-primitive actions in the following way:

$$\begin{aligned} \mathit{start\_move\_finger} = {} & \mathit{start\_move\_terminal\_phalanx} \\ & + \mathit{start\_move\_middle\_phalanx} \\ & + \mathit{start\_move\_upper\_phalanx} \end{aligned}$$

$$\begin{aligned} \mathit{end\_move\_finger} = {} & \mathit{end\_move\_terminal\_phalanx} \\ & + \mathit{end\_move\_middle\_phalanx} \\ & + \mathit{end\_move\_upper\_phalanx} \end{aligned}$$

generated by the listed six primitive actions.

The anchoring function Φ does not need any modification; the main difference from the previous cases is that the scattering $(\mathrm{CS}_{i-1}, \mathrm{CS}_i)$, corresponding to a complex action, involves a change in the position of more than one knoxel in the conceptual space.

4.5. The anchoring system at work

To describe the system at work, consider the assertions generated from the sequence of the finger pushing a ball. The initial situation S_0 corresponds to an initial configuration $\mathrm{CS}_0 = \{\mathbf{k_a}, \mathbf{k_b}, \mathbf{k_c}, \mathbf{k_d}\}$ in which the first three knoxels correspond to the finger phalanxes at rest, and the last knoxel corresponds to the resting ball. In this situation, the fluents *quiet_finger*(S_0) and *quiet_ball*(S_0) hold.

When the camera perceives the motion of the finger, a scattering occurs in the conceptual space, and a new configuration $\mathrm{CS}_1 = \{\mathbf{k}'_\mathbf{a}, \mathbf{k}'_\mathbf{b}, \mathbf{k}'_\mathbf{c}, \mathbf{k_d}\}$ is generated, in which the scattering of first three knoxels represents the beginning of the composite motion of the whole finger. The last knoxel, corresponding to the resting ball, remains unchanged.

In the linguistic area, this scattering corresponds to an occurrence of the instantaneous composite action *start_move_finger*. The new situation $S_1 = do(\mathit{start\_move\_finger}, S_0)$ (i.e. the situation resulting from performing in S_0 the action *start_move_finger*) corresponds in the CS to the sequence of configurations $\{\mathrm{CS}_0, \mathrm{CS}_1\}$. In the new situation, the fluents *moving_finger*(S_1) and *quiet_ball*(S_1) hold.

At a certain time point, the camera perceives the finger touching the ball. A new scattering occurs, that affects the knoxel corresponding to the ball. The configuration of the conceptual space becomes: $\mathrm{CS}_2 = \{\mathbf{k}'_\mathbf{a}, \mathbf{k}'_\mathbf{b}, \mathbf{k}'_\mathbf{c}, \mathbf{k}'_\mathbf{d}\}$ in which the last knoxel scattered to a new position corresponding to the motion of the ball.

In the linguistic area, the new scattering corresponds to an occurrence of the instantaneous action *push_ball*. The current situation is now $S_2 = do(\mathit{push\_ball}, S_1)$, which corresponds to the sequence of CS configurations $\{\mathrm{CS}_0, \mathrm{CS}_1, \mathrm{CS}_2\}$. In this situation, the fluents *moving_finger*(S_2) and *pushed_ball*(S_2) hold.

Then, the finger stops its motion and a new rest state begins. A further scattering occurs, involving the knoxels that corrspond to the phalanxes of the finger. The CS configuration becomes: $\mathrm{CS}_3 = \{\mathbf{k}''_\mathbf{a}, \mathbf{k}''_\mathbf{b}, \mathbf{k}''_\mathbf{c}, \mathbf{k}'_\mathbf{d}\}$. Note that the last knoxel remains in its previous position; this because the ball went out the visual field of the camera, and, in such cases, the system assumes that objects indefinitely remain in the motion state they were observed the last time.

Now the current situation is $S_3 = do(\mathit{stop\_move\_finger}, S_2)$ and it corresponds to the sequence of CS configurations $\{\mathrm{CS}_0, \mathrm{CS}_1, \mathrm{CS}_2, \mathrm{CS}_3\}$. In this situation, the fluents *quiet_finger*(S_3) and *pushed_ball*(S_3) hold.

In the above example, the conceptual space is directly linked to the environment through perception: all the entities represented in the CS have a precise counterpart in the external world as perceived by the agent. The symbols generated at the linguistic area summarize the operations of the finger according to the situation calculus formalism. In this way, the conceptual and the linguistic area describe the finger operations at two different levels of representation: the *symbolic* one (at the linguistic area) and the *analogue* one at the conceptual area.

5. Discussion

In the last few years, the problem of anchoring symbols to data coming out of sensors became a relevant topic in autonomous robotics, and several proposals have been developed. In particular, a model which presents similarities with our approach is due to Coradeschi and Saffiotti. Briefly, our linguistic area corresponds to their *Symbol system*, and our subsymbolic area corresponds to their *Perceptual system*. In addition, our architecture includes a further level (the conceptual area), which is missing in their model. This choice allows the system to anchor symbols to representations with a rich geometric structure that can support various forms of reasoning, thus relieving the linguistic formalism of many tasks.

In our model, symbols are not anchored only to static objects (as in the approach by Coradeschi and Saffiotti), but also to temporal entities, such as fluents, situations and actions. Fluents, actions and situations are “high-level” symbolic terms; they summarize the dynamics of the scene, as represented by the dynamics of the knoxels in the conceptual space. Another advantage of our approach is that the use of the CS dispense us from defining “low-level” sensor fluents [33].

According to Coradeschi and Saffiotti, anchoring involves two issues: the *representational* and the *procedural* issues. In this paper we primarily face the former. As far as procedural issues are concerned, Coradeschi and Saffiotti introduce a *tracking* and a *reacquiring* functionality, in order to follow and update the link between symbols and percepts. Currently, in our architecture, the procedural aspects are delegated to the subconceptual area. For example, the snake algorithms have the burden of tracking and eventually reacquiring the primitive shapes corresponding to the knoxels. An interesting line of research may be the formalization of the prediction and updating capabilities typical of the Kalman filters in the terms of conceptual space representations.

Presently, we presuppose a *top-down* design, in the sense that the designer of the system is responsible for several tasks: choosing the dimensions of the conceptual space, defining the predicates that describe at the symbolic level the actions and the fluents, and so on. An important improvement would consist in adding some *self-organization* capabilities. For example, the system should be able to *explore* the CS and discover interesting structures of knoxels that can be linked to new symbols in the linguistic area, e.g. by means of a system similar of the *SSH* architecture proposed by Kuipers [20].

A related improvement would consist in adding the capability of learning sequences of actions by experience and imitation, as proposed by Nicolescu and Mataric [24]. In our model, a sequence of actions corresponds to a sequence of scatterings in the CS. Such sequences could be learned by the system, e.g. by means of suitable recurrent neural networks.

Presently, or model has been developed having in mind a single robotic agent. An interesting research topic concerns a generalization towards a multiagent architecture. Each agent would be endowed with its own conceptual and linguistic areas, and the passing of messages among agents may be aimed to a *convergence* of conceptual spaces. This generalization would be useful for the anchoring of the multiagent extensions of the situation calculus proposed by Shapiro et al. [29].

Multiagent architectures may also play *language games* of the kind described by Steels [31] and Sierra-Santibáñez [30]. The cooperation and competition among the agents may allow them to suitably build their conceptual space by taking into account only the dimensions of the CS relevant to the competitions. In this way, the CS evolution would be determined by the interaction of the agents.

6. Conclusions

In the above sections, a possible interpretation of the language of the situation calculus in terms of conceptual spaces is suggested. In this way, the situation calculus can be adopted as the formalism for the linguistic area of the model, with the advantage of using a powerful, well understood and widespread formal tool.

In recent years, in the field of *cognitive robotics*, various formalisms based on situation calculus have been proposed, such as [12,13,21]. They allow the programmer to describe the robot operations at an high level of abstraction, and account for concurrent processes with priorities, interrupts, reactivity to exogenous events and so on. Some of such formalisms, although in simplified versions, have been tested on working robots for various tasks, such as, among others, mail delivery applications [18], interactive museum guides [3] and control of mobile manipulators [14].

In many of these systems, perception is taken into account only in a marginal way: it is modelled by simple *sense actions* that test if some trivial condition holds in the environment (e.g. the state of a door or the position of a block on the table), or it is fully delegated to complex low-level execution modules [17].

Our approach is aimed to allow a robotic system to perform complex perceptual operation, in such a way that they can be integrated in its high-level activities. Therefore, actions, situations and fluents of the situation calculus are anchored in a theoretically well founded way to the perceptual activities of the robot, thus allowing for far richer descriptions of the evolution of the environment. In this way, an artificial vision system may generate situation calculus formulas describing which courses of actions are occurring in the external world, which are the current situations, which complex fluents hold, and so on. This rich representation is homogeneous with the high-level robot programming formalisms mentioned above. In particular, the integration of cognitive representations and rich perceptual information makes it possible to design forms of top-down, knowledge driven perceptual activities (e.g. perceptual explorations, attentive processes).

In addition, conceptual spaces can act as a *simulation structure* in the sense of Weyhrauch [32]. In other words, many forms of inference (particularly of spatial and causal nature) are more likely to be performed taking advantage from the geometric structure of the CS, rather than as logical deductions in the linguistic area. Immediate examples could be the tracking and reacquiring functionalities of anchoring [7] proposed by Coradeschi and Saffiotti which can be described in terms of knoxels expectations [6] in conceptual space.

Acknowledgements

Thanks to Gigina Aiello, Edoardo Ardizzone, Ron Arkin, Christian Balkenius, Shimon Edelman, Peter Gärdenfors, Donatella Guarino, Lars Kopp, Yves Lesperance, Fiora Pirri, Ray Reiter, Alessandro Saffiotti, Pino Spinelli and Luc Steels for having discussed with us over the years, the topics of this paper. Ignazio Infantino and Francesco Raimondi contributed to the hardware and software implementation of the system. This work has been partially supported by the Project “Efficient robot vision algorithms for the automatic recognition of objects in the International Space Station” sponsored by the Italian Space Agency (ASI).

References

[1] R. Arkin, Behavior-based Robotics, MIT Press, Cambridge, MA, 1998.

[2] A. Blake, M. Isard, Active Contours, Springer, Berlin, 1998.

[3] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, S. Thrun, Experiences with an interactive museum tour-guide robot, Artificial Intelligence 114 (1999) 3–55.

[4] A. Chella, M. Frixione, S. Gaglio, A cognitive architecture for artificial vision, Artificial Intelligence 89 (1997) 73–111.

[5] A. Chella, M. Frixione, S. Gaglio, An architecture for autonomous agents exploiting conceptual representations, Robotics and Autonomous Systems 25 (3–4) (1998) 231–240.

[6] A. Chella, M. Frixione, S. Gaglio, Understanding dynamic scenes, Artificial Intelligence 123 (2000) 89–132.

[7] A. Chella, M. Frixione, S. Gaglio, Conceptual spaces for symbol anchoring—review of the book: Conceptual Spaces. The geometry of Thought, by Peter Gärdenfors, Robotics and Autonomous Systems 43 (2003) 193–195.

[8] L. Conway, R.A. Volz, M.W. Walker, Teleautonomous systems: projecting and coordinating intelligent actions at a distance, IEEE Transactions on Robotics and Automation 6 (2) (1990) 146–158.

[9] S. Coradeschi, A. Saffiotti, Anchoring symbols to sensor data: preliminary report, in: Proceedings of the 17th AAAI Conference, AAAI Press, Menlo Park, CA, 2000, pp. 129–135.

[10] S. Coradeschi, A. Saffiotti (Eds.), Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems, Proceedings of the 2001 AAAI Fall Symposium, AAAI Technical Report FS-01-01, AAAI Press, Menlo Park, CA, 2001.

[11] S. Coradeschi, A. Saffiotti, Perceptual anchoring of symbols for action, in: Proceedings of the 17th International Conference on Artificial Intelligence IJCAI-01, Morgan Kaufmann, San Mateo, CA, 2001, pp. 407–412.

[12] G. De Giacomo, Y. Lesperance, H.J. Levesque, ConGolog, a concurrent programming language based on the situation calculus, Artificial Intelligence 121 (1–2) (2000) 109–169.

[13] G. De Giacomo, H.J. Levesque, An incremental interpreter for high-level programs with sensing, in: H.J. Levesque, F. Pirri (Eds.), Logical Foundations for Cognitive Agents, Springer, Berlin, 1999, pp. 86–102.

[14] A. Finzi, F. Pirri, M. Pirrone, M. Romano, M. Vaccaro, Autonomous mobile manipulators managing perception and failures, in: Proceedings of the AGENTS'01, ACM Press, 2001, pp. 196–203.

[15] P. Gärdenfors, Conceptual Spaces, MIT Press/Bradford Books, Cambridge, MA, 2000.

[16] M. Gelfond, V. Lifschitz, A. Rabinov, What are the limitations of the situation calculus?, in: R. Boyer (Ed.), Essays in Honor of Woody Bledsoe, Kluwer Academic Publishers, Dordrecht, 1991, pp. 167–179.

[17] D. Haehnel, W. Burgard, G. Lakemeyer, GOLEX—bridging the gap between logic (GOLOG) and a real robot, in: Proceedings of the 22nd German Conference on Artificial Intelligence (KI 98), Bremen, Germany, 1998.

[18] M. Jenkin, Y. Lesperance, H.J. Levesque, F. Lin, J. Lloyd, D. Marcu, R. Reiter, R.B. Scherl, K. Tam, A logical approach to portable high-level robot programming, in: Proceedings of the Tenth Australian Joint Conference on Artificial Intelligence, 1997, pp. 1–12.

[19] M. Kass, A. Witkin, D. Terzoupolos, Snakes: active contour models, in: Proceedings of First International Conference on Computer Vision, Springer, Berlin, 1987, pp. 259–268.

[20] B. Kuipers, The spatial semantic hierarchy, Artificial Intelligence 119 (2000) 191–233.

[21] H.J. Levesque, R. Reiter, Y. Lesperance, F. Lin, R. Scherl, GOLOG: a logic programming language for dynamic domains, Journal of Logic Programming 31 (1997) 59–84.

[22] J. McCarthy, P.J. Hayes, Some philosophical problems from the standpoint of artificial intelligence, in: B. Meltzer, D. Michie (Eds.), Machine Intelligence 4, Edinburgh University Press, 1969, pp. 463–502.

[23] R. Miller, M. Shanahan, Narratives in the situation calculus, Journal of Logic and Computation 4 (5) (1994) 513–530.

[24] M.N. Nicolescu, M. Mataric, Learning task representations from experienced demonstrations, in: S. Coradeschi, A. Saffiotti (Eds.), Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems, Proceedings of the 2001 AAAI Fall Symposium, AAAI Press, Menlo Park, CA, 2001, pp. 17–24.

[25] A.V. Oppenheim, R.W. Shafer, Discrete-time Signal Processing, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[26] J. Pinto, Temporal reasoning in the situation calculus, Technical Report, Department of Computer Science, University of Toronto, 1994, Toronto, CA.

[27] R. Reiter, Knowledge in Action: Logical Foundations for Describing and Implementing Dynamical Systems, MIT Press/Bradford Books, Cambridge, MA, 2001.

[28] M. Shanahan, Solving the Frame Problem, MIT Press, Cambridge, MA, 1997.

[29] S. Shapiro, Y. Lesperance, H. Levesque, Specifying communicative multiagent systems, in: W. Wobcke, M. Pagnucco, C. Zhang (Eds.), Agents and Multi-agent Systems Formalisms, Methodologies, and Applications, Lecture Notes in Artificial Intelligence, Springer, Berlin, 1998, pp. 1–14.

[30] J. Sierra-Santibáñez, Grounded models as a basis for intuitive reasoning: the origins of logical categories, in: S. Coradeschi, A. Saffiotti (Eds.), Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems, Proceedings of the 2001 AAAI Fall Symposium, AAAI Press, Menlo Park, CA, 2001, pp. 101–108.

[31] L. Steels, The Talking Heads Experiment, vol. I, Words and Meaning, 1999, Antwerpen, LABORATORIUM, Special pre-edition.

[32] R.W. Weyhrauch, Prolegomena to a theory of mechanized formal reasoning, Artificial Intelligence 13 (1–2) (1980) 133–170.

[33] M. Witkowski, D. Randell, M. Shanahan, Deriving fluents from sensor data for mobile robots, in: S. Coradeschi, A. Saffiotti (Eds.), Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems, Proceedings of the 2001 AAAI Fall Symposium, AAAI Press, Menlo Park, CA, 2001, pp. 44-51.

A. Chella was born in Florence, Italy, on 4 March 1961. He received his laurea degree in Electronic Engineering in 1988 and his Ph.D. in Computer Engineering in 1993 from the University of Palermo, Italy. Since 2001 he is a professor of robotics at the University of Palermo and a scientific advisor of the CERE (Center of Study of Computer Networks) of the Italian Research, Council (CNR). His research interests are in the field of autonomous robotics, artificial vision, hybrid (symbolic–subsymbolic) systems and knowledge representation. He is a member of IEEE, ACM and AAAI.

M. Frixione was born in Genoa, Italy, in 1960. He received his Laurea degree and his Ph.D. in Philosophy from the University of Genoa, respectively, in 1986 and 1993. Currently, he is Assistant Professor in Philosophy of Language at the Department of Communication Sciences of the University of Salerno, Italy. His research interests are in the field of Cognitive Sciences and Artificial Intelligence, and include Knowledge Representation, Hybrid Systems and the philosophical aspects of Cognitive Sciences.

S. Gaglio was born in Agrigento, Italy, on 11 April 1954. He graduated in electronic engineering at the University of Genoa, Genoa, Italy in 1977. In 1977 he was awarded a Fulbright scholarship to attend graduate courses in USA, and in 1978 he received the M.S.E.E. degree from the Georgia Institute of Technology, Atlanta, USA. Since 1986 he is a professor of artificial intelligence at the

University of Palermo, Italy. Since 1999 he is the Director of CERE (Center of Study of Computer Networks) of the Italian Research Council (CNR). He has been member of various committees for projects of national interest in Italy and he is referee of various international scientific congresses and journals. His present research activities are in the area of artificial intelligence and robotics. He is a member of IEEE, ACM, and AAAI.

Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Robotics and Autonomous Systems 43 (2003) 189–192

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Book review

Evans' *Varieties of Reference* and the anchoring problem

Michael L. Anderson

Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, USA

To think about how to anchor abstract symbols to objects in the world is to become part of a tradition in philosophy with a long history, and an especially rich recent past. It is to ask, with Wittgenstein, "What makes my thought about him, a thought about *him*?" and thus it is to wonder not just about the nature of referring expressions or singular terms, but about the nature of referring beings. With this in mind I hereby endeavor—briefly, incompletely, but hopefully still usefully—to introduce what in my judgment is the single best philosophical starting-point for those interested in understanding the referential connections between symbols and the world, and the cognitive, epistemic, and linguistic capacities which support them: *The Varieties of Reference* by Gareth Evans.[1]

It is worthwhile first of all to note, as the title indicates, that it is the *varieties* of reference that are of interest. It is Evans' contention that no single theory can account for our various use of singular terms; although the different kinds of reference share certain features, and rely on related cognitive, linguistic and epistemic capacities, it appears that, rather than being a class defined by necessary and sufficient criteria for membership, they form a family of abilities, united, like a thread, by its overlapping fibers.

Evans does not defend this claim so much as display it in his account. Much of the underlying variety in reference can be brought out by considering the guiding principle of the work as a whole, which Evans calls Russell's Principle: "The principle is that a subject cannot make a judgment about something, unless he knows which object the judgment is about" (p. 89). A judgment is here construed as something very general, of the form: ⟨a is F⟩. Given the generality of the account, it seems fairly clear that the ability to make—to determine the truth of—some such judgments is necessary for autonomous systems (even when this ability is not implemented in the form of a per se symbolic reasoner). Insofar as this is true—and given that Russell's principle is correct (I will not delve into Evans' interesting and convincing defense)—any autonomous system must know (or have the ability to discover) which thing in the world ⟨a⟩ is.

This hardly seems objectionable. The trouble, as Evans himself admits, is in spelling out what such knowledge amounts to. He suggests that the condition for knowing which thing ⟨a⟩ is might be met by an agent who: (1) possesses the knowledge of some discriminating feature of ⟨a⟩, or (2) has the ability to locate ⟨a⟩ in her vicinity, or (3) has the capacity to recognize ⟨a⟩, that is, the disposition to identify one (and only one) object as ⟨a⟩. Of course, even this specification of conditions leaves ample room for alternate interpretations (Evans spends some time on an effective critique of the photograph and causal theories of reference, demonstrating the inadequacy of their versions of the above criteria) but it does neatly and naturally suggest three varieties of reference deserving of further investigation: (1) information-based reference, (2) demonstrative reference, and (3) recognition-based reference, to which list Evans adds some other items, of which self-reference is the most important.

E-mail address: mikeoda@cs.umd.edu (M.L. Anderson).

[1] Gareth Evans, The Varieties of Reference, Clarendon Press, Oxford, 1984, xiii + 418 pages, ISBN 0-19-824686-2.

doi:10.1016/S0921-8890(03)00022-8

Taking each in turn, and roughly, an information-based thought about ⟨a⟩ "is the result of a belief about how the world is which the subject has because he has received information (or misinformation) from the object" (p. 121). In this case, the reference is to the object from which the information derives, even in the case where that information is mistaken, as in the famous case of referring to 'the man holding the champagne', whose glass is in fact full of sparkling cider. The paradigm case of demonstrative reference is the simple 'this', but also includes 'that', 'here', 'there', and all like descriptionless, indexical identifications. Finally (I shall here ignore self-reference, although Evans' account of it is interesting, and the relation he describes between 'here'-thoughts and 'I'-thoughts is central to his overall account) recognition-based reference deals with the case where an agent refers to an object previously encountered and remembered. Evans writes: "[I]f a subject is disposed to identify a particular object as the object of his thought, and in so doing is exercising a genuine recognitional capacity stemming from the encounter or encounters from which the memory information that saturates the thought derives, then, it seems to me, that object is the object of his thought, irrespective of whether or not it can be identified by means of any descriptions which the subject might otherwise have" (p. 269).

It is likely that Evans' discussions of demonstrative and recognition-based reference will have the most immediate relevance to those involved in understanding anchoring. And in this regard it is worth mentioning what I take to be Evans' greatest strength, considered from the standpoint of one interested in the behavior of autonomous, embodied agents: his insistence on situating reference in the larger context of being and acting in the world. I am impressed in particular with his argument that demonstrative reference requires of the agent awareness of an ego-centered space within which (and in terms of which) experience is instantiated and actions effected. Consider, in this regard, the difference between the judgments ⟨There's a fire here⟩ and ⟨There's a fire there⟩, or ⟨There's a dollar here⟩. Surely successful anchoring has not been displayed by a system that does not react differently in each case. That is, it is not enough to tag an object with an arbitrary symbol, and maintain this connection (although doing even this is not without its challenges!); one must connect with the right symbol, in the right way, so as to support appropriate reasoning about, and reaction to, the objects of the world.

In addition to recommending *The Varieties of Reference* as the single best philosophical resource for those interested in this immense project, I have also compiled a brief bibliography of core readings [1–12], and a longer list of other useful and important work [13–64]. It is my hope that the collective encounter with these works can help build a Lingua Franca of anchoring, without which the collaborative effort required to advance understanding in this difficult area will be much hindered.

Core readings

[1] T. Burge, Individualism and the Mental, Midwest Studies in Philosophy IV, University of Minnesota Press, Minneapolis, MN, 1979.

[2] K. Donellan, Reference and definite descriptions, in: A.P. Martinich (Ed.), The Philosophy of Language, Oxford University Press, Oxford, 1985.

[3] G. Evans, The Varieties of Reference, Oxford University Press, Oxford, 1982.

[4] G. Frege, On sense and meaning, in: A.P. Martinich (Ed.), The Philosophy of Language, Oxford University Press, Oxford, 1985.

[5] D. Kaplan, Demonstratives, in: A.P. Martinich (Ed.), The Philosophy of Language, Oxford University Press, Oxford, 1985.

[6] S. Kripke, Naming and Necessity, Harvard University Press, Cambridge, MA, 1980.

[7] A.P. Martinich (Ed.), The Philosophy of Language, Oxford University Press, Oxford, 1985.

[8] J. McDowell, Singular thought and the extent of inner space, in: P. Pettit, J. McDowell (Eds.), Subject, Thought, and Context, Oxford University Press, Oxford, 1986.

[9] J. McDowell, On the sense and reference of a proper name, Mind 86 (1977) 159–185.

[10] J. Perry, The Problem of the Essential Indexical, Oxford University Press, Oxford, 1993.

[11] H. Putnam, The meaning of meaning, in: Mind, Language and Reality, Cambridge University Press, Cambridge, 1975.

[12] P.F. Strawson, On referring, in: A.P. Martinich (Ed.), The Philosophy of Language, Oxford University Press, Oxford, 1985.

Extended bibliography

[13] F. Ackerman, Content, character, and nondescriptive meaning, in: J. Almog, J. Perry, H. Wettstein (Eds.), Themes from Kaplan, Oxford University Press, Oxford, 1989.

[14] F. Ackerman, Proper names, propositional attitudes, and nondescriptive connotations, Philosophical Studies 35 (1979) 55–69.

[15] L. Addis, Intrinsic reference and the new theory, in: P.A. French, T.E. Uehling, H.K. Wettstein (Eds.), Midwest Studies in Philosophy XIV: Contemporary Perspectives in the Philosophy of Language, University of Minnesota Press, Minneapolis, MN, 1990.

[16] K. Akins, Of sensory systems and the aboutness of mental states, Journal of Philosophy 93 (7) (1996) 337–372.

[17] K. Bach, Indexical content, in: Routledge Encyclopedia of Philosophy, Routledge, New York, 1998.

[18] K. Bach, Thought and Reference, Oxford University Press, Oxford, 1987.

[19] K. Bach, Thought and object: *de re* representations and relations, in: M. Brand, R.M. Harnish (Eds.), The Representation of Knowledge and Belief, University of Arizona Press, Tucson, AZ, 1986.

[20] T. Blackburn, The elusiveness of reference, in: P.A. French, T.E. Uehling, H.K. Wettstein (Eds.), Midwest Studies in Philosophy XII: Realism and Antirealism, University of Minnesota Press, Minneapolis, MN, 1988.

[21] D. Braun, Empty names, Noûs 32 (4) (1993).

[22] T. Burge, Reference and proper names, Journal of Philosophy 70 (1973) 425–439.

[23] H.N. Castañeda, Direct reference, the semantics of thinking, and guise theory, in: J. Almog, J. Perry, H. Wettstein (Eds.), Themes from Kaplan, Oxford University Press, Oxford, 1989.

[24] M. Devitt, Against direct reference, in: P.A. French, T.E. Uehling, H.K. Wettstein (Eds.), Midwest Studies in Philosophy XIV: Contemporary Perspectives in the Philosophy of Language, University of Minnesota Press, Minneapolis, MN, 1990.

[25] M. Devitt, Singular terms, Journal of Philosophy 71 (1974) 183–205.

[26] K. Donellan, Proper names and identifying descriptions, in: S.P. Schwartz (Ed.), Naming, Necessity and Natural Kinds, Cornell University Press, Ithaca, NY, 1977.

[27] F. Dretske, The intentionality of cognitive states, in: W. Lycan (Ed.), Mind and Cognition, 2nd ed., Basil Blackwell, Oxford, 1999.

[28] F. Dretske, Aspects of cognitive representation, in: M. Brand, R. Harnish (Eds.), The Representation of Knowledge and Belief, University of Arizona Press, Tucson, AZ, 1986.

[29] M. Dummett, Frege: Philosophy of Language, 2nd ed., Harvard University Press, Cambridge, MA, 1981.

[30] M. Dummett, The Interpretation of Frege's Philosophy, Harvard University Press, Cambridge, MA, 1981.

[31] G. Evans, The causal theory of names, in: A.P. Martinich (Ed.), The Philosophy of Language, Oxford University Press, Oxford, 1985.

[32] G. Evans, Understanding Demonstratives, Collected Papers, Oxford University Press, Oxford, 1985.

[33] J. Fodor, A theory of content, in: W. Lycan (Ed.), Mind and Cognition, 2nd ed., Basil Blackwell, Oxford, 1999.

[34] G. Forbes, Cognitive architecture and the semantics of belief, in: P.A. French, T.E. Uehling, H.K. Wettstein (Eds.), Midwest Studies in Philosophy XIV: Contemporary Perspectives in the Philosophy of Language, University of Minnesota Press, Minneapolis, MN, 1990.

[35] I. Hacking, Representing and Intervening, Cambridge University Press, Cambridge, 1983.

[36] I. Hacking, Why Does Language Matter to Philosophy? Cambridge University Press, Cambridge, 1975.

[37] T. Horgan, J. Tienson, The intentionality of phenomenology and the phenomenology of intentionality, in: D. Chalmers (Ed.), Philosophy of Mind: Classical and Contemporary Readings, Oxford University Press, Oxford, 2002.

[38] S. Kripke, On rules and private language, in: A.P. Martinich (Ed.), The Philosophy of Language, Oxford University Press, Oxford, 1985.

[39] S. Kripke, Speaker's reference and semantic reference, in: A.P. Martinich (Ed.), The Philosophy of Language, Oxford University Press, Oxford, 1985.

[40] C. Li, Natural kinds: direct reference, realism and the impossibility of necessary, a posteriori truth, Review of Metaphysics 47 (1993) 261–276.

[41] K. Ludwig, Singular thought and the Cartesian theory of mind, Noûs 30 (4) (1996).

[42] J. McDowell, Mind and World, Harvard University Press, Cambridge, MA, 1994.

[43] C. McGinn, The mechanism of reference, Synthese 49 (1981) 157–186.

[44] M. McKinsey, Names and intentionality, Philosophical Review 87 (1978) 171–200.

[45] R. Millikan, Biosemantics, in: W. Lycan (Ed.), Mind and Cognition, 2nd ed., Basil Blackwell, Oxford, 1999.

[46] R. Millikan, White Queen Psychology and Other Essays for Alice, MIT Press, Cambridge, MA, 1995.

[47] R. Millikan, Language, Thought and Other Biological Categories, MIT Press, Cambridge, MA, 1984.

[48] M. O'Donovan-Anderson, Content and Comportment: On Embodiment and the Epistemic Availability of the World, Rowman and Littlefield, Lanham, 1997.

[49] D. Perlis, Consciousness as self function, Journal of Consciousness Studies 4 (5–6) (1997) 509–525.

[50] D. Perlis, Putting one's foot in one's head—Part II. How?, in: E. Dietrich (Ed.), From Thinking Machines to Virtual Persons: Essays on the Intentionality of Computers, Academic Press, New York, 1994.

[51] D. Perlis, Putting one's foot in one's head—Part I. Why? Noûs 25 (1991) 425–455.

[52] J. Perry, Indexicals and demonstratives, in: B. Hale, C. Wright (Eds.), A Companion to Philosophy of Language, Blackwell, Oxford, 1997.

[53] H. Putnam, Explanation and reference, in: Mind, Language, and Reality, Cambridge University Press, Cambridge, 1975.

[54] M. Richard, Indexicals, in: W. Bright (Ed.), International Encyclopedia of Linguistics, Oxford University Press, Oxford, 1992.

[55] F. Recanati, Direct Reference: From Language to Thought, Blackwell Press, Oxford, 1993.

[56] B. Russell, Descriptions, in: A.P. Martinich (Ed.), The Philosophy of Language, Oxford University Press, Oxford, 1985.

[57] S. Schiffer, Indexicals and the theory of reference, Synthese 49 (1981) 43–100.

[58] S. Schiffer, The basis of reference, Erkenntnis 13 (1978) 171–206.

[59] J. Searle, Proper names, in: A.P. Martinich (Ed.), The Philosophy of Language, Oxford University Press, Oxford, 1985.

[60] J. Searle, Intentionality, Cambridge University Press, Cambridge, 1983.

[61] R. Stalnaker, Indexical belief, Synthese 49 (1981) 129–151.

[62] P.F. Strawson, Meaning and truth, in: A.P. Martinich (Eds.), The Philosophy of Language, Oxford University Press, Oxford, 1985.

[63] P.F. Strawson, Individuals, Routledge, London, 1959.

[64] H. Wettstein, How to bridge the gap between meaning and reference, Synthese 84 (1984) 63–84.

Michael L. Anderson holds a Ph.D. in Philosophy from Yale University. He is currently a post-doctoral research associate at the University of Maryland, Institute for Advanced Computer Studies, where he studies the principles of self-correcting agency in real-world settings. His primary research goal is to show how reasoning and mental representation are grounded in physical structures and environmental interactions. Thus, for instance, he has argued that symbol anchoring requires the ability to interact with and manipulate the objects in one's environment. Current research includes an account of proprioception, detailing both how the particular structure of this aspect of our bodily experience is shaped by the physical and neural pathways which support it, as well as the role this bodily sense plays in structuring and supporting perception more generally.

Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Robotics and Autonomous Systems 43 (2003) 193–195

Robotics and Autonomous Systems

www.elsevier.com/locate/robot

Book review

Conceptual spaces for anchoring

A. Chella[a,b], M. Frixione[b,c,*], S. Gaglio[a,b]

[a] Dipartimento di Ingegneria Informatica, Università di Palermo, Palermo, Italy
[b] ICAR-CNR, Palermo, Italy
[c] Dipartimento di Scienze della Comunicazione, Università di Salerno, Salerno, Italy

In his book, *Conceptual Spaces. The Geometry of Thought*[1], Peter Gärdenfors proposes the notion of *conceptual space* as a way to overcome the opposition between the traditional, symbolic representations of "good old-fashioned" artificial intelligence on the one side, and the connectionist, subsymbolic representations on the other side. Conceptual spaces would offer a third kind of approach to knowledge representation in the cognitive sciences.

The conceptual space approach is based on a geometric treatment of concepts and knowledge representation. According to Gärdenfors, concepts are not independent of each other, and are structured into *domains*. Examples of possible domains are shape, color, taste, sound, the domain of kinematic properties, the domain of dynamic properties, and so on. A *conceptual space* is structured as a set of domains. Each domain is in its turn defined in terms of a set of *quality dimensions*. For example, the domain of *color* could be made up by such dimension as *hue*, *saturation* and *brightness*, the dimensions of the domain of *taste* could be *sweet*, *bitter*, *saline* and *sour*. Other examples of quality dimensions for some possible domain could be *temperature*, *weight*, *time*, *pitch*. Quality dimensions can be either more or less tightly connected to observable properties, or more abstract in nature (e.g., concerned with functional or social aspects, or deriving from scientific categorizations, as is the case, for example, of *mass*).

Each quality dimension has a particular geometric structure (typically, a topological or a metric structure). For example, the *weight* dimension is presumably isomorphic to the half-line of real non-negative numbers. Other quality dimensions have different structures: the *hue* dimension of the *color* domain is likely to be circular; there can be discrete dimensions, and so on.

In any case, the form of a conceptual space strictly depends on the cognitive structure and abilities of a given class of agents. For example, the color domain in non-human animals, or in artificial agents with different kind of sensors can deeply differ from the human color domain. Moreover, in human beings certain dimensions can be assumed to be innate, other to depend from cultural factors.

According to Gärdenfors, given a certain conceptual space individual objects are represented as points: every object is characterized by a set of values, one value for each dimension of the domains constituting the conceptual space. The values of the dimensions are the coordinates of the point representing the object. As a consequence of the geometric structure of the dimensions, a notion of *distance* can be defined between the points of a space. The distance between two points can be interpreted as a measure of the similarity of the corresponding individuals.

[*] Corresponding author.
E-mail addresses: chella@unipa.it (A. Chella), frix@dist.unige.it (M. Frixione), gaglio@unipa.it (S. Gaglio).

[1] MIT Press, Cambridge, MA, 2000, 317 pages, ISBN 0262071991.

0921-8890/03/$ – see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0921-8890(03)00023-X

Concepts are represented as *regions* in a conceptual space: a concept corresponds to the region of the space in which are located the points that share certain features at some degree. (Gärdenfors distinguishes between *properties*, which are based on a single domain, and *concepts*, which involve different domains. For sake of simplicity, we shall not consider this distinction in the following.) The geometric structure of conceptual spaces and the interpretation of distance in terms of similarity allows for a geometric treatment of concepts. Different geometric properties of regions correspond to different kinds of concepts. A special role in Gärdenfors' theory is played by so called *natural* concepts, which correspond to convex regions in a conceptual space. In the case of natural concepts, conceptual spaces allow to account for prototypical effects: given a certain convex region representing a natural concept, the central points of the region correspond to "better", or "more typical" instances of the category, the peripheral points correspond "less typical" instances.

Such a geometric treatment of concepts is employed by Gärdenfors in order to face many problems within the field of the cognitive sciences. Examples taken from the book include categorization, concept formation, concept learning, induction, metaphors, lexical semantics for natural languages.

Symbolic, subsymbolic and conceptual approaches must not be intended as competing paradigms. Rather, according to Gärdenfors, they correspond to different, coexisting levels of representation within a cognitive system. Subsymbolic (or subconceptual) level is the lowest level of representations, directly connected to perception; in it information is represented in terms of neural patterns of activation. The linguistic level is the most abstract one. The conceptual level (i.e., the level of conceptual spaces) is situated between symbols and subsymbolic patterns. In particular, the conceptual level can be seen as an internal semantic level for the symbolic representations: symbolic expressions are given a meaning in terms of geometric structures in conceptual spaces. In this perspective, symbol grounding can be achieved through the geometric representations of the conceptual level, which, in their turn, are connected to action and perception through the subsymbolic computations at the subconceptual level.

Many forms of inference can be accounted for at the conceptual level. In particular, it is our opinion that most forms of spatial and causal reasoning are likely to be performed at the conceptual level, taking advantage from the geometric structure of conceptual spaces. In this perspective, the role of the symbolic level could be primarily concerned with communication, and with some special forms of abstract reasoning.

Let us consider now the relevance of conceptual spaces for anchoring. Anchoring has to do with connecting symbols and sensor data that refer to the same physical object, and with preserving such a correspondence as the environment or the state of the agent change. Sensor data pertain to the subconceptual level. Conceptual spaces act as an intermediary between symbols and subconceptual processing. Therefore, conceptual spaces are a good candidate for the study and formalization of anchoring.

We said before that individual objects correspond to points in a conceptual space. This is certainly true from a synchronic point of view. However, in a diachronic perspective, objects can be more profitably seen as *trajectories* in conceptual spaces. As a given state of affairs evolves, the properties of the involved objects change: if an object moves, its spatial coordinates are modified; it can happens that objects alter their shape or color during the course of time, and so on. As the properties of objects are modified, the points representing them in a conceptual space move, and describe a certain trajectory. Such trajectories usually have relevant geometrical properties, they show important regularities. For example, as far as the changes of some object are gradual the corresponding trajectory is smooth, physical laws constrain the change of the values of the various dimensions, and so on.

In our opinion, anchoring could take great advantage from a geometric formulation in terms of conceptual spaces. Let us consider two typical anchoring functionalities, namely *tracking* and *reacquiring*. *Tracking* consists in keeping a symbol aligned to the corresponding perceptual data as such data change in time. Rather than some form matching of (possibly complex) symbolic descriptions to perceptual data, tracking could be more fruitfully seen as a form of inference at the conceptual level, that takes advantages from the geometric structure of conceptual spaces. Starting from the geometric properties of the corresponding trajectory, hypotheses can be made concerning the future evolution of a given object. Figuring out the evolution of an object (its future position, or the way in which his features are going to change) could allow to keep its symbolic representation aligned to the corresponding perceptual data. In this spirit, tracking can be considered as a matter of extrapolating trajectories in a conceptual space.

Reacquiring has to do with recognizing an object that has been re-observed after some time (for example, after that it has been occluded for a while behind some obstacle). In most cases, also reacquiring can be more profitably seen as a form of conceptual level reasoning, rather than as a process driven by some form of symbolic inference. Typically, it can be traced back to some form of interpolation in a conceptual space. To reacquire an object amounts to realize that two separate segments of trajectories in a conceptual space can be seen as parts of the same overall trajectory (the former segment corresponding to the disappeared object, the latter to the reappeared one). In other terms, to identify again an object that disappeared amounts to interpolate trajectories in conceptual spaces.

In this perspective, symbolic expressions are nothing more than mere labels for the entities represented at the conceptual level. However, this does not exclude that anchoring in conceptual spaces could take advantage from top down information: high level, symbolic knowledge can constrain the possible shape of the (interpolated or extrapolated) trajectories.

Summing up, according to the spirit of Gärdenfors' proposal, various forms of reasoning can be profitably seen as forms of geometric reasoning performed at the conceptual level. Such a *geometric* approach to knowledge representation and reasoning would be fruitful for the aims of autonomous robots design. In particular, it is our opinion that it could offer a general framework for the study, the formalization and the implementation of anchoring in cognitive systems.

Antonio Chella was born in Florence, Italy, on 4 March 1961. He received his Laurea degree in Electronic Engineering in 1988 and his Ph.D. in Computer Engineering in 1993 from the Università di Palermo, Italy. Since 2001 he is professor of robotics at the Università di Palermo and scientific advisor of the Center of Study of Computer Networks (CERE) of the Italian Research Council (CNR). His research interests are in the field of autonomous robotics, artificial vision, hybrid (symbolic–subsymbolic) systems and knowledge representation. He is a member of IEEE, ACM and AAAI.

Marcello Frixione was born in Genoa, Italy, in 1960. He received his Laurea degree and his Ph.D. in Philosophy from the University of Genoa, Genoa, Italy, respectively in 1986 and 1993. Currently, he is professor of Philosophy of Language at the Department of Communication Sciences of the Università di Salerno, Italy. His research interests are in the field of cognitive sciences and artificial intelligence, and include knowledge representation, hybrid systems and the philosophical aspects of cognitive sciences.

Salvatore Gaglio was born in Agrigento, Italy, on 11 April 1954. He graduated in Electronic Engineering at the University of Genoa, Genoa, Italy in 1977. In 1977 he was awarded a Fulbright scholarship to attend graduate courses in USA, and in 1978 he received the M.S.E.E. degree from the Georgia Institute of Technology, Atlanta, USA. From 1986 he is a professor of artificial intelligence at the Università di Palermo, Italy. From 1999 he is the Director of Center of Study of Computer Networks (CERE) of the Italian Research Council (CNR). He has been member of various committees for projects of national interest in Italy and he is referee of various international scientific congresses and journals. His present research activities are in the area of artificial intelligence and robotics. He is a member of IEEE, ACM, and AAAI.