Anchoring Symbols on Conceptual Spaces: the Case of Dynamic Scenarios

A. Chella [a,1], M. Frixione [b,2] and S. Gaglio [a,3]

[a] *Dipartimento di Ingegneria Informatica, Università di Palermo, Italy, and ICAR-CNR, Palermo, Italy*

[b] *Dipartimento di Scienze della Comunicazione, Università di Salerno, Italy*

Abstract

The paper shows the link that anchors a conceptual representation of dynamic scenarios to one of the most influential symbolic formalisms used in cognitive robotics, namely the *situation calculus*. A cognitive architecture for robot vision is taken into account as the reference framework. The discussion is based on an experimental setup aimed at obtaining an intelligent visual control of a robotic finger starting from visual data.

Key words: Anchoring; Robot vision; Conceptual spaces; Actions; Situation Calculus

1 Introduction

A cognitive architecture for robot vision has been proposed by the authors [3–5] aimed at the representation of knowledge extracted from visual data related with dynamic scenarios. One of the main assumption of the architecture design is a principled integration of the approaches developed within the artificial vision community, and the propositional systems developed within symbolic knowledge (KR) representation 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 shows in details the link that anchors the conceptual representation for representing dynamic scenes to one of the most influential symbolic formalisms

---

[1] Corresponding author. E-mail: chella@unipa.it

[2] E-mail: frix@dist.unige.it

[3] E-mail: gaglio@unipa.it

Preprint submitted to Elsevier Science *12 August 2002*

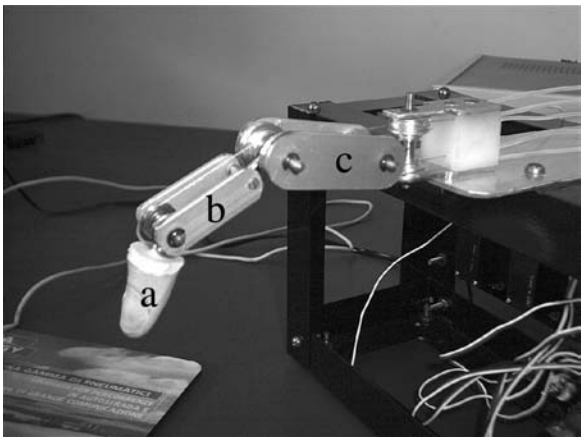

Fig. 1. The robotic finger used in the experimental setup. The terminal phalanx *a*, the middle phalanx *b* and the upper phalanx *c* are shown.

adopted in cognitive robotics, namely the *situation calculus*. It is discussed in particular how *actions*, *situations* and *fluents* may be anchored (in the sense of the computational theory of anchoring proposed by Coradeschi and Saffiotti [7,9]) to the representations in the conceptual level, which are in turns generated from the robot perceptions (for an up to date survey on different perspectives on anchoring see also Coradeschi and Saffiotti [8]).

The main motivation for the adoption of situation calculus is that it is one of the simplest and more deeply analyzed logic formalisms for the representation of knowledge about actions and change and in general about dynamical systems. The situation calculus was primarily developed by McCarthy and Hayes [18], for up to date and exhaustive introductions see Shanahan [24] and Reiter [23]. Today, it is a widely adopted formalism in the *cognitive robotics* literature; efficient Prolog implementations have been proposed, as the GOLOG system [17]; simplified versions of the situation calculus are also at the basis of working mobile robots [12].

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 finger has been entirely developed at the Robotics Laboratory of the Department of Computer Engineering of the 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).

In the described setup, the robotic finger is driven by schematic behaviors [1] to perform 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 finger itself, in order to perform intelligent *visual servoing* operations (see [13]). The acquired visual data are anchored to symbolic descriptions of the operations of the finger.

In a nutshell, the operation of the whole system is summarized as follows: it takes in input a sequence of images corresponding to subsequent phases of the evolution

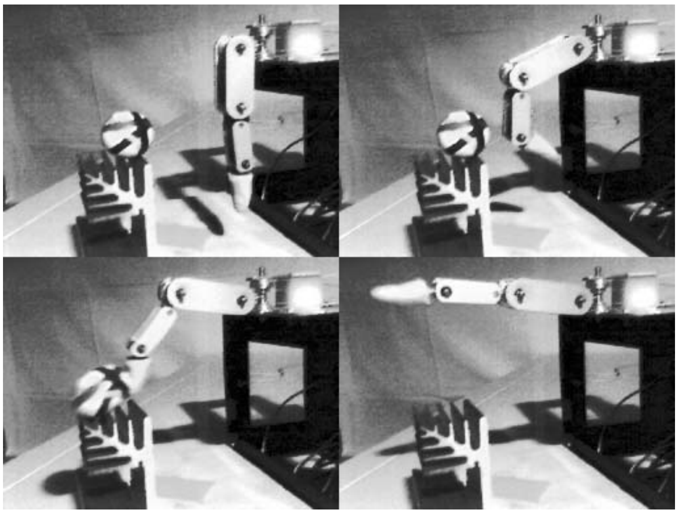

Fig. 2. The robotic finger pushes a ball (from the upper left to the lower right).

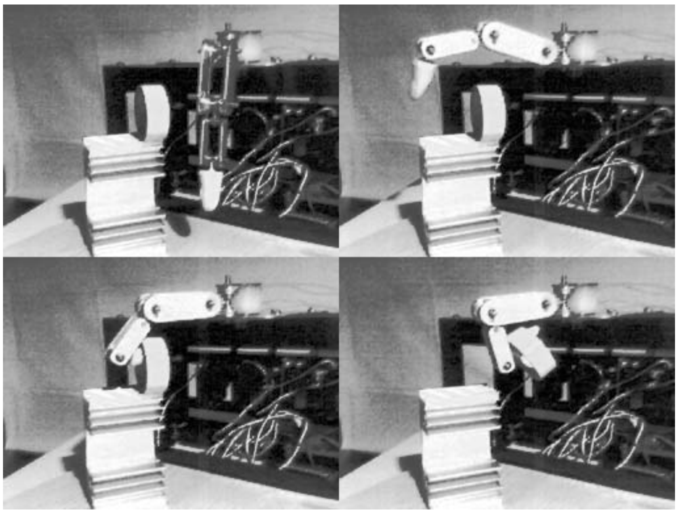

Fig. 3. The robotic finger picks up a simple torus-shaped object (from the upper left to the lower right).

of the scene (the movements of the robotic finger), and it produces in output a declarative description of the scene, formulated as a set of assertions written in the situation calculus formalism.

This symbolic description may be employed to perform high-level inferences, as the ones needed to generate complex long-range plans, or to perform causal and diagnostic reasoning about the system operations. The description may also be used to generate explanations in readable form of the operations of the finger to perform high-level teleautonomy [6].

The paper is organized as follows. In the next Section, the main assumptions underlying the cognitive architecture is summarized. The third Section is devoted to a synthetic description of the conceptual level representation of motion. The fourth

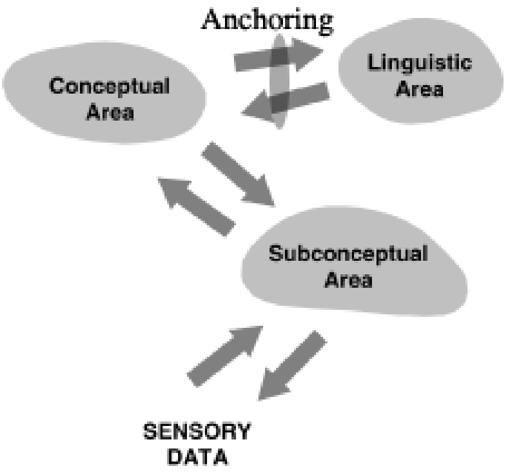

Fig. 4. The three areas of representation, and the relations among them.

Section shows in details how the situation calculus is anchored on the adopted conceptual representation . The last Section discusses the proposed framework with respect to the relevant anchoring frameworks described in the literature. Short conclusions follows.

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 the aspects of man-machine interaction. They face only in a marginal way the general aspects of representation of knowledge (see [5] for a review).

The proposed cognitive architecture hypothesizes that a principled integration of the approaches of artificial vision and of symbolic KR requires the introduction of an intermediate representation between these two kinds of representation. The role of such a representation is played by the notion of *conceptual space* introduced by Gärdenfors [10]. In the described architecture, this intermediate representation is the place where the anchoring occur 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. The term subconceptual suggests that 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 situation calculus formalism. 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 on sensory data by mapping them on the representations in the conceptual area. The purpose of the subsequent discussion is to show how the adopted conceptual representation firmly anchors the propositions in 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* [10] 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. 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 $\mathbf{k}$ is a vector in CS; from the conceptual point of view, it is the epistemologically primitive element at the considered level of analysis. In the case of static scenes [3], a knoxel coincides with a 3D primitive shape, described in terms of a constructive solid geometry (CSG) schema. For example, the robotic finger (Fig. 1) may be described by the knoxels describing the terminal phalanx a, the middle phalanx b and the upper phalanx c.

To account for the perception of dynamic scenes, an intrinsically *dynamic conceptual space* is adopted. The main assumption behind the 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:

$$\mathbf{k} = [x_1(t), x_2(t), \ldots, x_n(t)] \tag{1}$$

where n is the number of degrees of freedom of the moving 3D primitive, e.g., a the 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 $f_j^i(t)$:

$$x_i(t) = \sum_j X_j^i f_j^i(t) \tag{2}$$

It is therefore 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 [21]). 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 as 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. The conceptual space configuration in this case is completely described by the three knoxels participating to the motion of the finger:

$$CS = \{\mathbf{k_a}, \mathbf{k_b}, \mathbf{k_c}\} \tag{3}$$

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 primitive. Such a transition is called a “scattering” from one knoxel to another. This corresponds to a discontinuity in time, and is associated with an instantaneous event.

In the example of the moving finger, a scattering occurs when the finger has reached its upper 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

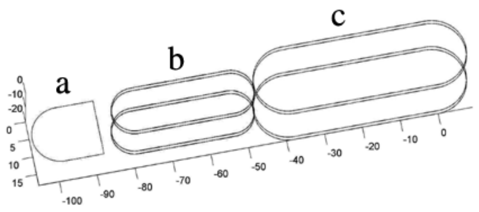

Fig. 5. The CAD model of the robot finger.

terminal phalanx) is replaced by knoxel $\mathbf{k}'_{\mathbf{a}}$, and, similarly, knoxels $\mathbf{k}_{\mathbf{b}}$ and $\mathbf{k}_{\mathbf{c}}$ are replaced by $\mathbf{k}'_{\mathbf{b}}$ and $\mathbf{k}'_{\mathbf{c}}$. The new CS' configuration is:

$$CS' = \{\mathbf{k}'_{\mathbf{a}}, \mathbf{k}'_{\mathbf{b}}, \mathbf{k}'_{\mathbf{c}}\}. \tag{4}$$

The occurred scattering may be described by the set of the two CSs configurations before and after the scattering:

$$\{CS, CS'\} \equiv \{\{\mathbf{k}_{\mathbf{a}}, \mathbf{k}_{\mathbf{b}}, \mathbf{k}_{\mathbf{c}}\}, \{\mathbf{k}'_{\mathbf{a}}, \mathbf{k}'_{\mathbf{b}}, \mathbf{k}'_{\mathbf{c}}\}\}. \tag{5}$$

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 model of the finger built up by a 3D CAD system (Fig. 5).

The images acquired by the camera are processed to extract the finger contours by an algorithm based on *snakes* [2]. A snake is a curve that moves in the image under the influence of forces related to the local distribution of the gray levels. Briefly, when the snake reaches an object contour, it is attracted by the contour and it adapts its shape to assume the contour shape. When the object moves or change its shape, the snake continues to adapt itself in order to track the object.

Formally, a snake is described in a parametric form by:

$$v(s) = (x(s), y(s)) \tag{6}$$

where $x(s)$ and $y(s)$ are the coordinates along the shape contour and s is the normalized arc length:

$$s \in [0, 1] \tag{7}$$

In the case of the finger, the adopted snake model reflects the geometric constraints imposed by the 3D model. The snake defines the energy of a contour E_{snake}, to be:

$$E_{snake}(v(s)) = \int_0^1 (E_{int}(v(s)) + E_{image}(v(s))ds \tag{8}$$

The energy integral is a functional since its variable s is a function (the shape contour). The internal energy E_{int} is formed from a Tikhonov stabilizer and is defined by:

$$E_{int}(vvs)) = a(s) \left| \frac{dv(s)^2}{ds^2} \right| + b(s) \left| \frac{dv(s)^2}{ds^2} \right|^2 \tag{9}$$

where $|.|$ is the Euclidean norm.

The first order continuity term, weighted by $a(s)$, let the contours behave elastically, whilst the second order curvature term, weighted by $b(s)$, let it be resistant to bending. For example, setting $b(s) = 0$ at point s, allows the snake to become second-order discontinuous at point and to generate a corner.

The image functional determines the features which will have a low image energy and hence the features that attract the contours. In general, this functional is made up by three terms:

$$E_{image} = w_{line} T_{line} + w_{edge} E_{edge} + w_{term} E_{term} \tag{10}$$

where w denotes a weighting constant. The w and E corresponds to lines, edges and termination, respectively. The adopted snake model only presents the edge functional which attracts the snake to points with an high edge gradient:

$$E_{image} = E_{edge} = -(G_\sigma * \nabla^2 I(x, y))^2 \tag{11}$$

This is the image functional proposed by Kass, Witkin and Terzoupolos [14]. 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.

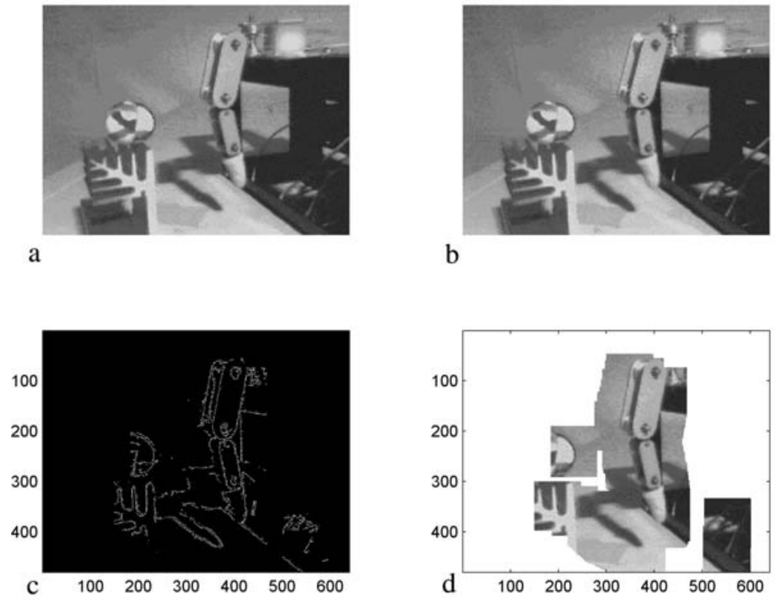

Fig. 6. The filtering operation to individuate the ROI of the finger.

In order to extract the Regions Of Interest (ROI) related to the acquired scenes, some standard filtering operations are performed (see Fig. 6): from the acquired image (a), the noise is reduced by a 5x5 median filter (b); the moving parts are detected by the Canny algorithm (c); image intensities between frames are subtracted in order to individuate the ROI (d).

Fig. 7 shows the attraction operation of the snake on the upper phalanx. After the snake initializes its position and dimensions by individuating the finger contours, it tracks the finger positions during its operations (Fig. 8) The geometric information about the finger position obtained by the tracking snake are sent to the 3D CAD system that generates a VRML animated model of the operation of the finger operations (Fig. 9).

Finally, the data related to the movement of each phalanx are sent to a software module that performs the DFT operation, in order to generate the knoxel configurations of the conceptual space.

4 Anchoring situation calculus on conceptual spaces

The linguistic area is where the dynamics of the conceptual space are represented in terms of logic assertions expressed in the *situation calculus* formalism. Indeed, the representation presupposed by the situation calculus is in many respects homogeneous to the conceptual representation described in the previous Section.

To anchor grounded assertions expressed in situation calculus in terms of structures

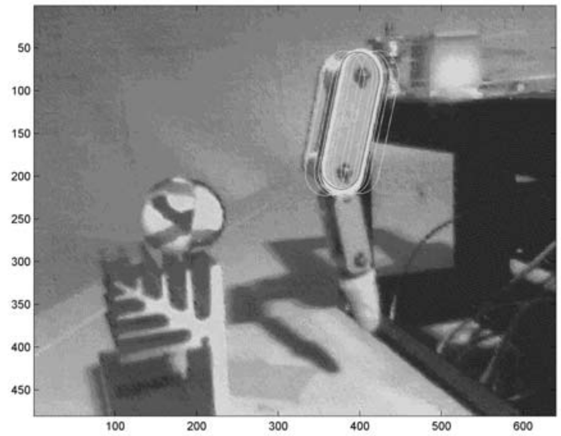

Fig. 7. The attraction operation of the snake.

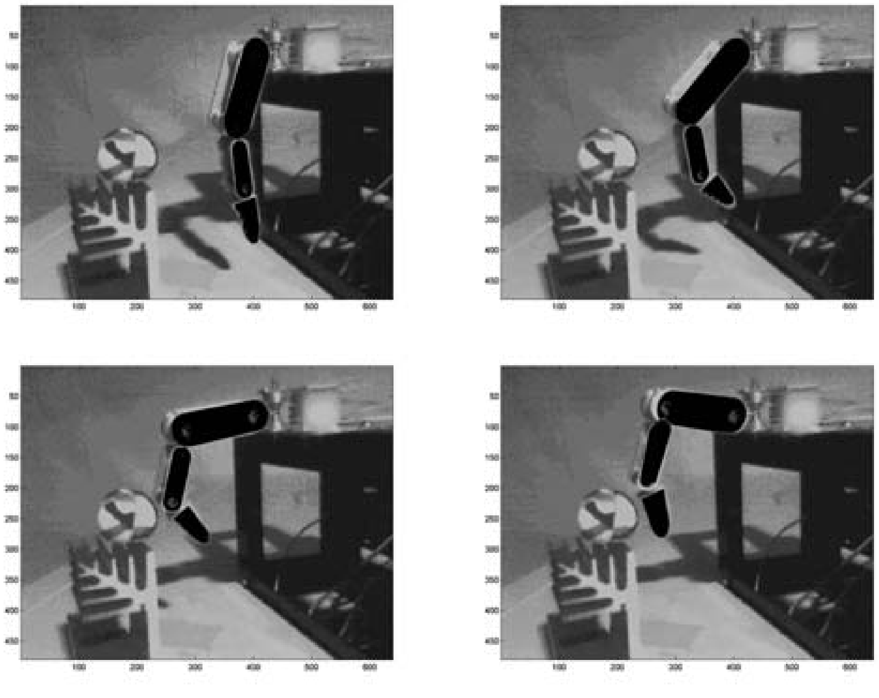

Fig. 8. The tracking operation of the snake.

in the conceptual space, an *anchoring function* Φ is defined from time t to couples $(Assertion, CSstructure)$, where $Assertion$ is a grounded well formed formula in situation calculus, and $CSstructure$ is a suitable structure of knoxels in the conceptual space. In the current experimental setup, the function Φ is implemented as an ad hoc look-up table; other more general implementations, e.g., by means of neural networks or systems of fuzzy rules, are currently under study.

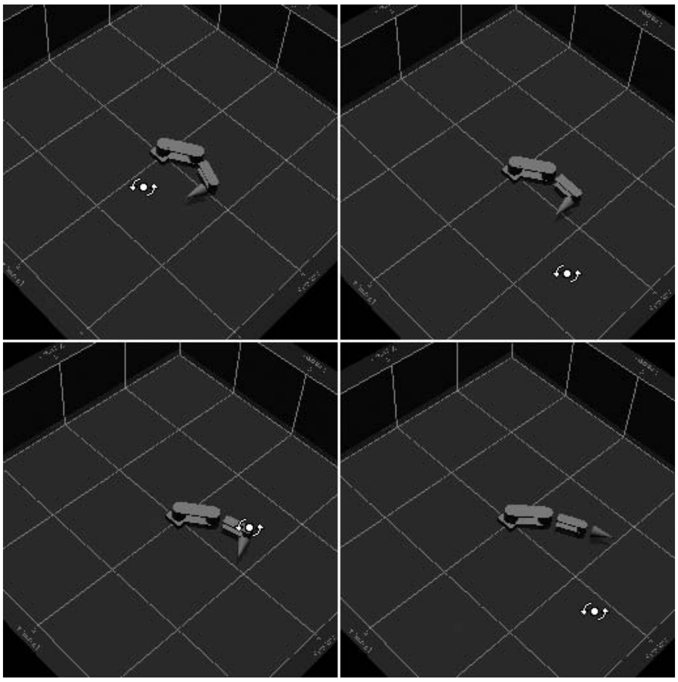

Fig. 9. The VRML animated model of the finger operation.

4.1 Anchoring actions and situations

The basic idea behind the situation calculus is that the evolution of a state of affairs is modelled 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 formalized using 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. Given a first order term o denoting a specific object, the term $pick\_up(o)$ denotes the specific action consisting in picking up o, as in Fig. 3.

In terms of conceptual space, the action a corresponds to a suitable scattering of knoxels in the conceptual space described by the set $\{CS_{i-1}, CS_i\}$ of the two configurations of knoxels before and after the scattering:

$$\Phi_A : time \to (a, \{CS_{i-1}, CS_i\}) \tag{12}$$

The initial situation S_0 corresponds to the initial configurations of knoxels CS_0 in the conceptual space:

$$\Phi_S : time \to (S_0, CS_0) \tag{13}$$

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. Therefore, S_i is anchored to the sequence of configurations of knoxels generated by the sequence of scattering corresponding to the actions:

$$\Phi_S : time \to (S_i, \{CS_0, CS_1, \ldots, CS_{i-1}, CS_i\}) \tag{14}$$

It should be noted that, following from these definitions, the formula $S_i = do(a, S_{i-1})$ means that the action a generates the new situation $\{CS_0, CS_1, \ldots, CS_{i-1}, CS_i\}$ starting from the old one $\{CS_0, CS_1, \ldots, CS_{i-1}\}$.

As an example, consider the finger scenario. Suppose that the terminal phalanx of the finger is initially in the rest state in the position p_1. The initial situation S_0 is anchored to the configuration $CS_0 = \{\mathbf{k_a}, \mathbf{k_b}, \mathbf{k_c}\}$, where $\mathbf{k_a}$ denotes the terminal phalanx in rest state and $\mathbf{k_b}, \mathbf{k_c}$ the other two knoxels also in rest state. Now the phalanx moves from position p_1 to position p_2, and then it rests in p_2. When the motion of the phalanx from p_1 towards p_2 starts, a scattering occurs in the conceptual space CS, and the knoxel $\mathbf{k_a}$ changes its position. Therefore the new configuration of knoxels is $CS_1 = \{\mathbf{k'_a}, \mathbf{k_b}, \mathbf{k_c}\}$ and the new situation S_1 is anchored to the sequence $\{CS_0, CS_1\}$.

Such a scattering corresponds to an (instantaneous) action that is represented by the term $start\_move\_terminal\_phalanx(p_1, p_2)$. The knoxel $\mathbf{k'_a}$ 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 $\mathbf{k'_a}$ continues to be active in it.

When the motion of the phalanx ends, a further scattering occurs, $\mathbf{k'_a}$ disappears, and a new knoxel $\mathbf{k''_a}$ becomes active. Therefore, a new configuration $CS_2 = \{\mathbf{k''_a}, \mathbf{k_b}, \mathbf{k_c}\}$ is generated. This second scattering $\{CS_1, CS_2\}$ corresponds to the instantaneous action $end\_move\_terminal\_phalanx(p_1, p_2)$. The knoxel $\mathbf{k''_a}$ corresponds again to the rest state of the phalanx, but now in the p_2 position. The new situation S_2 is now anchored to the configuration $\{CS_0, CS_1, CS_2\}$.

4.2 Anchoring fluents

According to 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.

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.

In the described framework, a relational fluent f is, in general, anchored to the set of knoxels involved in the description of the fluent itself:

$$\Phi_F : time \rightarrow (f, \{k_a, k_b, \ldots, k_l\}) \tag{15}$$

For example, the fluent $moving\_terminal\_phalanx(p_1, p_2, s)$ is anchored to the knoxel $\mathbf{k_a'}$; the fluent $staying\_terminal\_phalanx(p_1, s)$ is anchored to the knoxel $\mathbf{k_a}$ and the fluent $staying\_terminal\_phalanx(p_2, s)$ is anchored to the knoxel $\mathbf{k_a''}$.

A fluent f holds in situation S_i if the set of knoxels describing the fluents belong to the configuration of knoxels corresponding to the situation:

$$\{k_a, k_b, \ldots, k_l\} \subseteq \{CS_0, CS_1, \ldots, CS_{i-1}, CS_i\} \tag{16}$$

For example, the fluent $moving\_terminal\_phalanx(p_1, p_2, s)$, anchored to $\mathbf{k_a'}$, holds in situation S_1 and it does not hold in S_0 and in S_2. As another example, the fluent $staying\_terminal\_phalanx(p_1, s)$ holds in situation S_0, and the fluent $staying\_terminal\_phalanx(p_2, s)$ holds in situation S_2.

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 Pinto [22] and Chap. 7 of the book of Reiter [23]). Suppose to represent the action of moving the robot finger from point p_1 to point p_2. The process of moving the finger from p_1 to p_2 is initiated by an instantaneous action, say $start\_move\_finger(p_1, p_2)$, and is terminated by another instantaneous action, say $end\_move\_finger(p_1, p_2)$. In the formalism of the situation calculus, processes correspond to relational fluents. For example, the process of moving the finger from p_1 to p_2 corresponds to the fluent $moving\_finger(p_1, p_2, s)$. A formula like $moving\_finger(p_1, p_2, S_1)$ means that in situation S_1 the finger 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

In its traditional version, the 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 time 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, for example, of the motion of the finger of 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 the introduced 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 different knoxels. 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 the arguments of the actions are not taken into account, i.e., the starting position and the final position). 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 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 motion of the phalanxes. Consider for example $start\_move\_finger$. 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 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. Gelfond, Lifschitz and Rabinov [11] and Pinto [22] add to the language of the situation calculus a two argument function $+$, that, 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 $a = a_1 + a_2 + \ldots + a_n$, then it is possible to write $a_i \in a$ for each i such that $1 \leq i \leq n$.

In the described approach, the scattering of a single knoxel in CS is anchored to a primitive action; several knoxels scattering at the same time correspond to a complex action resulting from concurrently performing different primitive actions. The representation of the finger motion example in the formalism of the (concurrent) situation calculus involves two non primitive actions:

$$\begin{aligned} start\_move\_finger = {} & start\_move\_terminal\_phalanx + \\ & + start\_move\_middle\_phalanx + \\ & + start\_move\_upper\_phalanx \\ end\_move\_finger = {} & end\_move\_terminal\_phalanx + \\ & + end\_move\_middle\_phalanx + \\ & + end\_move\_upper\_phalanx \end{aligned}$$

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 now the scattering $\{CS_{i-1}, CS_i\}$ related to a complex action involves the changing of the position of more than one knoxel in the conceptual space.

In the literature, another approach to actions with temporal duration and with concurrency with an explicit treatment of time is based on the *narratives* introduced by Miller and Shanahan [19] (see also the book of Shanahan [24]). Narratives make a clearer distinction between the effects of the actions (as in the do function) and the occurrences of the actions in time by a function $happens(a, t)$, which states that action a occurs at time t. In order to manage narratives, the conceptual space should be rearranged to consider an explicit time parameter for each CS configuration, and the anchoring function Φ should be consequently rearranged.

To describe the system at work, consider the assertions generated from the sequence of images representing the finger that picks up the ball. The initial situation S_0 corresponds to the initial configuration $CS_0 = \{\mathbf{k_a}, \mathbf{k_b}, \mathbf{k_c}, \mathbf{k_d}\}$ in which the first three knoxels correspond to the finger phalanxes at rest, and the last knoxel correspond to the rest ball. In this situation, the fluents $quiet\_finger(S_0)$ and $quiet\_ball(S_0)$ holds.

When the camera perceives the motion of the finger, a scattering occurs in its conceptual space, and a new configuration $CS_1 = \{\mathbf{k'_a}, \mathbf{k'_b}, \mathbf{k'_c}, \mathbf{k_d}\}$ is generated, in which the first three knoxels scattered to describe the composite motion of the phalanxes of the finger. The last knoxel, related to the ball, remains in the previous position. In the linguistic area, this scattering corresponds to the instantaneous composite action $start\_move\_finger$.

The new situation $S_1 = do(start\_move\_finger, S_0)$ (resulting from performing in S_0 the action $start\_move\_finger$) corresponds in CS to the sequence of configurations $\{CS_0, CS_1\}$. In this situation, the fluent $moving\_finger(S_1)$ and $quiet\_ball(S_1)$ holds.

At a certain time point, the camera perceive the finger touching the ball, thus a new scattering occurs, that involves the knoxel related to the ball. The new configuration of knoxels is now: $CS_2 = \{\mathbf{k'_a}, \mathbf{k'_b}, \mathbf{k'_c}, \mathbf{k'_d}\}$ in which the last knoxel scattered to a new position describing the pushed ball. In the linguistic area, this new scattering corresponds to the instantaneous action $push\_ball$.

The current situation is now $S_2 = do(push\_ball, S_1)$ and it corresponds to the sequence of CS configurations $\{CS_0, CS_1, CS_2\}$. In this situation, the fluent $moving\_finger(S_2)$ and $pushed\_ball(S_2)$ holds.

Now, the finger stop its motion, remaining in a new, quiet state. A new scattering occurs involving the knoxels describing the phalanxes of the finger. The CS configuration is now: $CS_3 = \{\mathbf{k''_a}, \mathbf{k''_b}, \mathbf{k''_c}, \mathbf{k'_d}\}$. It should be noted that the last knoxel remain in its current position; this happens because the ball is now out of the camera visual field and, in this case, the system hypothesizes that it indefinitely remains in its motion state.

The current situation is $S_3 = do(stop\_move\_finger, S_2)$ and it corresponds to the sequence of CS configurations $\{CS_0, CS_1, CS_2, CS_3\}$. In this situation, the fluent $quiet\_finger(S_3)$ and $pushed\_ball(S_3)$ holds.

It should be noted from this example, that the anchoring of situation calculus to conceptual space entities, described by the Φ function, allows the logic assertions expressed in situation calculus to be immediately linked on suitable structures in

CS. In this respect, the conceptual area acts as a *simulation structure* in the sense of Weyhrauch [28]. In facts, the assertions of situation calculus are not proved in the logic theoretic sense, but they are "interpreted" as a summary of the dynamics of the knoxels in the conceptual space.

Another interesting perspective of this approach is the disappearing of the *frame problem* (see Shanahan [24] for a review). As described in the example, an action corresponding to a scattering of certain knoxels in CS only alters the positions of the knoxels related to the action itself. In the new generated situation the knoxels which were not affected by the action do not change their positions. Therefore, all the fluents which are related to the unaffected knoxels, simply remain in their truth state. In this sense, the conceptual area has the burden of maintain the information about the part of the environment which is not affected by the execution of an action, without the needs of adopting complex non-monotonic formalisms.

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 frameworks have been developed. In particular, a cited model which presents similarities with the described approach is due to Coradeschi and Saffiotti. Briefly, the linguistic area corresponds to their *Symbol system* and the subsymbolic area corresponds to their *Perceptual system*. The proposed architecture aims to a clearer and theoretically well-founded intermediate level (the conceptual area) which is missing in their model and that allows a sharp distinction between subconceptual and conceptual representations.

The choice of introducing the intermediate conceptual area allow the system to anchor symbols to data which own a well defined structure: according to the conceptual space theory, knoxels are defined in terms of dimensions; a metric function among knoxels is defined that capture the concept of similarity; convex sets of knoxels correspond to basic categories. In this sense the introduced anchoring function Φ is defined in a clear and well founded way.

In the proposed approach, symbols are anchored not only to static objects, ad in the Coradeschi and Saffiotti proposal, but also to complex temporal entities, as fluents, situations and actions. In this sense, the fluents, the actions and the situations are "high-level" symbolic terms, and they summarize the dynamics of the scene, which correspond to the dynamics of knoxels in the conceptual space. Another advantage of the approach is that the use of the *CS* avoids us the needs to define "low-level" sensor fluents as Witkowski, Randell and Shanahan [29].

The proposed model primarily faces one aspect of anchoring: the representation issue. According to Coradeschi and Saffiotti, the other aspect of anchoring is the procedural one. They introduce the *tracking* and the *reacquiring* functionalities that allows them to follow and update the link between symbols and percepts. Currently, in the proposed architecture, the procedural aspects of anchoring are delegated to the subconceptual area: e.g., the snakes algorithms have the burden of track and eventually reacquire the primitives corresponding to the knoxels. An interesting line of research may be aimed at the formalization the prediction and update capabilities typical of the Kalman filters in terms of the conceptual space representation.

Presently, the proposed architecture presupposes a *top-down* design, in the sense that the system designer 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 to the system. 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, for example by means of a system similar of the *SSH* architecture proposed by Kuipers [15,16].

A related improvement would consist in adding the capability of learning sequences of actions by experience and imitation, as proposed by Nicolescu and Mataric [20]. In the proposed architecture, a sequence of actions corresponds to a sequence of scatterings in the CS. Such sequences could be learned for example by suitable recurrent neural networks.

The proposed architecture 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. [25].

Multiagent architectures may also play *language games* of the kind described by Steels [27] and Sierra-Santibáñez [26]. 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 is determined by the interaction of the agents.

6 Conclusions

In the above Sections, a possible interpretation of the language of the situation calculus is suggested in terms of conceptual spaces. In this way the situation calculus can be adopted as the linguistic area formalism for the model, with the advantage of using a powerful, well understood and widespread formal tool. Besides this, a conceptual interpretation of the situation calculus is interesting in itself. Indeed, it could be considered complementary with respect to traditional, model theoretic interpretations for logic oriented representation languages.

Model theoretic semantics (in its different versions: purely Tarskian for extensional languages, possible worlds semantics for modal logic, preferential semantics for non monotonic formalisms, and so on) has been developed with the aim of accounting for certain metatheoretical properties of logical formalisms (such as logical consequence, validity, correctness, completeness, and son on). However, it is of no help in establishing how symbolic representations are anchored to their referents.

In addition, the model theoretic approach to semantics is "ontologically uniform", in the sense that it hides the ontological differences between entities denoted by expressions belonging to the same syntactic type. For example, all the individual terms of a logical language are mapped onto elements of the domain, no matter of the deep ontological variety that may exist between the objects that constitute their intended interpretation. Consider the situation calculus. According to its usual syntax, situations, actions and objects are all represented as first order individual terms; therefore, in a model theoretic interpretation, they are all mapped on elements of the domain. This does not constitute a problem given the above mentioned purposes of model theoretic semantics. However, it becomes a serious drawback if the aim is that of anchoring symbols to their referents through the sensory activities of an agent.

In this perspective, an interpretation of symbols in terms of conceptual spaces of the form sketched in the above pages offers:

- a kind of interpretation that does not constitute only a metatheoretic device allowing to single out certain properties of the symbolic formalism; rather, it is assumed to offer a further level of representation that is, in some sense, closer to the data coming from sensors, and that, for this reason, can help in anchoring the symbols to the external world.
- A kind of interpretation that accounts for the ontological differences between the entities denoted by symbols belonging to the same syntactic category. This would result in a richer and finer grained model, that stores information that is not explicitly represented at the symbolic level, and that therefore can offer a further source of "analog" inferences, offering at the same time a link between deliberative inferential processes, and forms of inference that are closer to the lower levels of the cognitive architecture (reactive behaviors, and so on).

Acknowledgements

Authors would like to thank Gigina Aiello, Christian Balkenius, Shimon Edelman, Peter Gärdenfors, Lars Kopp, Fiora Pirri, Ray Reiter, Alessandro Saffiotti and Luc

Steels for having discussed over the years the topics of this paper. Thanks to Ignazio Infantino for the implementation of the vision system and to Francesco Raimondi for the development of the robotic finger.

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, 1998.

[2] A. Blake and M. Isard. *Active Contours*. Springer, Berlin, 1998.

[3] A. Chella, M. Frixione, and S. Gaglio. A cognitive architecture for artificial vision. *Artif. Intell.*, 89:73–111, 1997.

[4] A. Chella, M. Frixione, and S. Gaglio. An architecture for autonomous agents exploiting conceptual representations. *Robotics and Autonomous Systems*, 25(3-4):231–240, 1998.

[5] A. Chella, M. Frixione, and S. Gaglio. Understanding dynamic scenes. *Artif. Intell.*, 123:89–132, 2000.

[6] L. Conway, R.A. Volz, and M.W. Walker. Teleautonomous systems: Projecting and coordinating intelligent actions at a distance. *IEEE Trans. on Robotics and Automation*, 6(2):146–158, 1990.

[7] S. Coradeschi and A. Saffiotti. Anchoring symbols to sensor data: preliminary report. In *Proc. of the 17th AAAI Conf.*, pages 129–135, Menlo Park, CA, 2000. AAAI Press.

[8] S. Coradeschi and A. Saffiotti, editors. *Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems, 2001 AAAI Fall Symposium*. AAAI Press, Menlo Park, CA, 2001.

[9] S. Coradeschi and A. Saffiotti. Perceptual anchoring of symbols for action. In *Proc. of the 17th Intl. Conf. on Artificial Intelligence IJCAI-01*, pages 407–412, San Mateo, CA, 2001. Morgan Kaufmann.

[10] P. Gärdenfors. *Conceptual Spaces*. MIT Press, Bradford Books, Cambridge, MA, 2000.

[11] M. Gelfond, V. Lifschitz, and A. Rabinov. What are the limitations of the situation calculus? In R. Boyer, editor, *Essays in Honor of Woody Bledsoe*, pages 167–179. Kluwer Academic, 1991.

[12] D. Haehnel, W. Burgard, and G. Lakemeyer. Golex - bridging the gap between logic (GOLOG) and a real robot. In *Proc. of the 22nd German Conference on Artificial Intelligence (KI 98)*, Bremen, Germany, 1998, 1998.

[13] S. Hutchinson, G.D. Hager, and P.I. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651–6700, 1996.

[14] M. Kass, A. Witkin, and D. Terzoupolos. Snakes: active contour models. In *Proc. of First Intern. Conf. on Computer Vision*, pages 259–268. Springer-Verlag, 1987.

[15] B. Kuipers and Y.-T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and Autonomous Systems,*, 8:47–63, 1991.

[16] B. Kuipers. The spatial semantic hierarchy. *Artif. Intell.*, 119:191–233, 2000.

[17] H.J. Levesque, R. Reiter, Y. Lesperance, F. Lin, and R. Scherl. GOLOG: a logic programming language for dynamic domains. *Journal of Logic Programming*, 31:59–84, 1997.

[18] J. McCarthy and P.J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

[19] R. Miller and M. Shanahan. Narratives in the situation calculus. *Journal of Logic and Computation*, 4(5):513–530, 1994.

[20] M.N. Nicolescu and M. Mataric. Learning task representations from experienced demonstrations. In S. Coradeschi and A. Saffiotti, editors, *Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems, 2001 AAAI Fall Symposium*, pages 17–24, Menlo Park, CA, 2001. AAAI Press.

[21] A.V. Oppenheim and R.W. Shafer. *Discrete-Time Signal Processing*. Prentice Hall, Inc., Englewood Cliffs, N.J., 1989.

[22] J. Pinto. Temporal reasoning in the situation calculus. Technical report, Dept. of Computer Science, Univ. of Toronto, Toronto, CA, 1994.

[23] R. Reiter. *Knowledge in Action: Logical Foundations for Describing and Implementing Dynamical Systems*. MIT Press, Bradford Books, Cambridge, MA, 2001.

[24] M. Shanahan. *Solving the frame problem*. MIT Press, Cambridge, MA, 1997.

[25] S. Shapiro, Y. Lesperance, and H. Levesque. Specifying communicative multiagent systems. In W. Wobcke, M. Pagnucco, and C. Zhang, editors, *Agents and Multi-Agent Systems — Formalisms, Methodologies, and Applications*, Lecture Notes in Artificial Intelligence, pages 1–14, Berlin, 1998. Springer-Verlag.

[26] J. Sierra-Santibáñez. Grounded models as a basis for intuitive reasoning: The origins of logical categories. In S. Coradeschi and A. Saffiotti, editors, *Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems, 2001 AAAI Fall Symposium*, pages 101–108, Menlo Park, CA, 2001. AAAI Press.

[27] L. Steels. *The Talking Heads Experiment Volume I. Words and Meaning*. Antwerpen, 1999. Special pre-edition for LABORATORIUM.

[28] R.W. Weyhrauch. Prolegomena to a theory of mechanized formal reasoning. *Artif. Intell.*, 13(1-2):133–170, 1980.

[29] M. Witkowski, D. Randell, and M. Shanahan. Deriving fluents from sensor data for mobile robots. In S. Coradeschi and A. Saffiotti, editors, *Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems, 2001 AAAI Fall Symposium*, pages 44–51, Menlo Park, CA, 2001. AAAI Press.