In *Proceedings KR 2004*, pp. 64–72

A Logic-Based Formulation of Active Visual Perception

Murray Shanahan and David Randell

Department of Electrical and Electronic Engineering,
Imperial College London,
Exhibition Road,
London SW7 2BT,
England.
m.shanahan@imperial.ac.uk, d.randell@imperial.ac.uk

Abstract

Building on earlier attempts to characterise robot perception as a form of abduction, this paper presents a logical account of active visual perception in the context of an upper-torso humanoid robot. Using the event calculus to represent actions and change, and aspect graphs to represent shape, the formalisation captures the way the robot's knowledge of the objects in its workspace can increase through interaction with them.

1 Introduction

The goal of contemporary research in cognitive robotics is to endow robots with high-level cognitive skills by deploying the traditional AI concepts of representation and reasoning [Lespérance, *et al.*, 1994]. With a few exceptions [Pirri & Finzi, 1999; Pirri & Romano, 2002], most cognitive robotics work to date has concentrated on the issue of determining a robot's course of actions, and has assumed that perception is a black box that can supply, on demand, facts about the world to a higher-level cognitive component [Levesque, 1996; Scherl & Levesque, 2003]. Accordingly, the critical issue of how the transition is made from raw sensor data to meaningful symbolic representation is cast outside the theoretical framework used to tackle cognition.

By contrast, in [Shanahan, 1996a] and [Shanahan, 1996b], a logical account of robot perception was given, based on abduction, that draws a less sharp distinction between cognition and perception. This formal abductive treatment of perception has been influential in cognitive vision [Cohn, *et. al.*, 2003] and spatial reasoning [Hazarika & Cohn, 2002; Remolina & Kuipers, 2004], and has found application in several robotics projects [Shanahan, 2000; Shanahan & Witkowski, 2001]. In [Shanahan, 2002], the basic abductive account was extended to permit a bidirectional flow of information between cognition and low-level sensing, through the mechanism of expectation. The result is a theoretical framework which, being expressed in the *lingua franca* of formal logic, has the potential to embrace the full conceptual trinity of cognition, action, and perception.

The chief contribution of the present paper is to overcome a significant shortcoming in previous logical characterisations of robot perception using abduction. Specifically, while data from simple sensors has been given an abductive gloss in a dynamic setting [Shanahan, 1997b; Santos & Shanahan, 2002], and richer (visual) data has been given the abductive treatment in a static setting (ie: single frames) [Shanahan, 2002], the abductive account has yet to be extended to a *rich* sensory modality, such as vision, in a *dynamic* setting. This extension is important because, as researchers in active vision have demonstrated, the extra data resulting from a robot's interactions with its environment, far from increasing the burden of interpretation, can actually facilitate it through the provision of vital new cues about the scene [Aloimonos, *et al.*, 1987; Ballard, 1991]. Accordingly, the present paper offers a logical characterisation of *active* visual perception, based on abduction.

The experimental setup for this work is an upper-torso humanoid robot with a stereo camera, mounted on a pan-and-tilt head, and two arms, each having three degrees-of-freedom (Figure□). This robot is mechanically simpler than other humanoids that have recently been built in Japan and the U.S.A., but is nonetheless sufficient for our present purposes. The robot's task is to identify "interesting" objects on a cluttered workbench, and then to nudge them using visual servoing.

Typically, the changes in position and orientation of a nudged object will permit its shape to be pinned down more precisely. It is this capacity to improve the representation of a perceived object through interaction with it that the new theoretical account must encompass. In addition, the theoretical story has to some extent been validated by implementation. This takes the form of an abductive meta-interpreter, written in Prolog, which carries out a dialogue with a C++ program that uses standard off-the-shelf, low-level vision algorithms to pre-process incoming data from the robot's camera.

The paper is organised as follows. Section 2 recapitulates the basic idea of treating visual perception as abduction, and introduces the notion of explanatory value. Section 3 sketches an implementation of this abductive framework. Section 4 shows how to use the event calculus to give a dynamic account of perception, setting the scene for Section 5, which presents a version of abduction suitable for active vision. Finally, Section 6 applies this to sequences of visual frames, using the idea of an aspect graph.

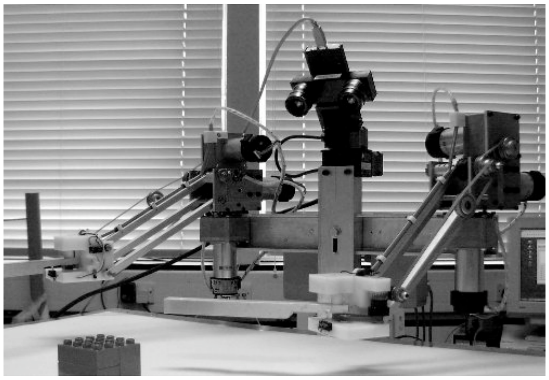

Figure 1: LUDWIG the Humanoid Robot

2 Visual Perception as Abduction

Logically speaking, visual perception can be characterised as follows [Shanahan, 2002]. Let Σ be a background theory that captures the causal relationship between objects in a scene and the low-level image data they give rise to. Then, given a conjunction Γ of formulae representing a collection of low-level image data, the job of perception is to find a conjunction Δ of formulae such that,

$$\Sigma \wedge \Delta \models \Gamma.$$

In other words, the idea is to find hypotheses about the external world that would explain the visual sensor data the robot has received. To rule out trivial or uninformative explanations, certain restrictions must be imposed on the nature of Δ. First, Δ must be consistent with Σ. Second, a set of predicates is designated as *abducible*, and Δ must mention only predicates from that set.

It's clear from this definition that, in general, there will be many Δs that can explain a given Γ. To distinguish them, we can introduce a measure of the *explanatory value* of a hypothesis. The main criterion to be met by this measure is that it should assign higher explanatory value to hypotheses that explain more data. Since the definition above insists that Δ *fully* explains Γ, this criterion is achieved through the deployment of *noise terms* [Poole, 1995; Shanahan, 1997b]. These are formulae that "explain away" items of sensor data as noise. The more noise terms a hypothesis has, the less its explanatory value should be.

Here are the formal details. Let $\Delta_1 \ldots \Delta_n$ be the set of all hypotheses that explain a given Γ according to the above definition. Given that one and only one of these hypotheses can be the true explanation, and if none of the hypotheses is subsumed by any other, we can assume $\Delta_1 \oplus \ldots \oplus \Delta_n$. Now, consider any hypothesis Δ_k, and let R be,

$$\Delta_1 \oplus \ldots \oplus \Delta_{k-1} \oplus \Delta_{k+1} \oplus \ldots \oplus \Delta_n.$$

Then, from the laws of probability, we have,

$$P(\Delta_k \mid \Delta_k \oplus R) = \left(1 + \frac{P(R)}{P(\Delta_k)}\right)^{-1} \tag{1}$$

where $P(\Delta_k)$ is the prior probability of Δ and $P(R)$ is the prior probability of R. From the form of R, we have,

$$P(R) = \left(\sum_{i=1}^{n} P(\Delta_i)\right) - P(\Delta_k). \tag{2}$$

For any hypothesis Δ of the form $\psi_1 \wedge \cdots \wedge \psi_m$, we have,

$$P(\Delta) = \prod_{j=1}^{m} P(\psi_j) \tag{3}$$

From equations (1) to (3), the posterior probability of any hypothesis can be calculated, given the set of all hypotheses, and this is taken to be its explanatory value [Poole, 1993; Shanahan, 2002]. All that remains is to assign suitable prior probabilities to the individual conjuncts in a hypothesis, that is to say to each type of formula ψ that can appear in a Δ. In practise, a hypothesis comprises two types of formula — formulae that postulate objects in the external world and describe their properties, and noise terms. It should be clear from equation (3) that, in general, a hypothesis explaining a portion of Γ with multiple noise terms will have a lower explanatory value than one explaining the same portion of Γ with a single formula that postulates an object in the external world. In this way, the definition of explanatory value meets its main design criterion.

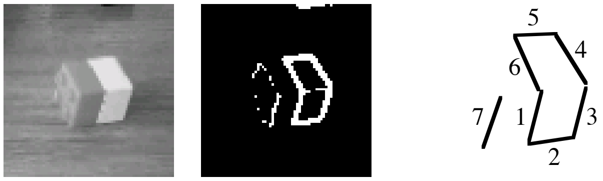

Figure 2: A Block and its Edges

To clarify all this, let's take a look at an example. Suppose the robot is interested in cuboidal objects, such as Lego bricks. In the implemented system, the recognition of such objects proceeds in two stages — first, the abduction of trapezoidal regions from straight line edges, then the abduction of cuboids from trapezoidal regions. Only the first stage is considered in what follows. The left-hand side of Figure 2 shows part of a sample image from one of the robot's cameras. The middle of the figure shows the result of applying a simple edge detection algorithm based on the Sobel operator. An edge following algorithm can then be used to extract seven prominent straight edges, as shown on the right of the figure.

Now, let the formula *Line*(w,p_1,p_2) mean that there is a visible line w from point p_1 to point p_2. Let □ be a conjunction of formulae representing the seven edges shown, including, for example, the following two formulae.

Line(1,[238,157],[241,147])

Line(2,[240,159],[247,157])

Next, let the formula *Region*(r) mean that there is a visible trapezoidal region r. Let the background theory □ include the following axioms.

$$\square p_1,p_2\ [\mathit{Line}(w,p_1,p_2)]\ \square \qquad \text{(L1)}$$
$$\square r\ [\mathit{Region}(r)\ \square\ \mathit{SideOf}(w,r)]$$

$$\square p_1,p_2\ [\mathit{Line}(w,p_1,p_2)]\ \square\ \ \mathit{Noise}(w) \qquad \text{(L2)}$$

The first axiom explains the presence of a visible line by the presence of a visible region, while the second axiom explains away a line as mere noise. In addition, let □ include a set of constraints about regions. These will insist, for example, that a (trapezoidal) region has only four sides, and that certain spatial relationships hold between them. Here’s an example.

$$[\mathit{SideOf}(w_1,r)\ \square\ \mathit{SideOf}(w_2,r)\ \square\ w_1 \neq w_2]\ \square$$
$$[\mathit{Parallel}(w_1,w_2)\quad \mathit{Joins}(w_1,w_2)]$$

Next, suppose the *Region* and *SideOf* predicates are abducible. Even with such a simple background theory, there are several competing hypotheses for explaining □ according to the abductive definition. For example, let $\square_1$ be a hypothesis that posits two regions, one bounded on three sides by lines 1, 2, and 3, and the other bounded on three sides by lines 4, 5, and 6, with line 7 being explained away as noise. The formulae comprising $\square_1$ will include the following.

Region(R_0)

SideOf(1,R_0)

SideOf(2,R_0)

Noise(7)

Let $\square_2$ be an alternative hypothesis that also posits a region bounded by lines 4, 5, and 6, but whose second posited region is bounded on two sides by lines 3 and 7. In $\square_2$, lines 1 and 2 are dismissed as noise. (A less trivial □ would take account of partial sides.) Now suppose the prior probability assigned to both *Region* and *Noise* formulae is 0.5. Then P($\square_1$) = 0.125 and P($\square_2$) = 0.0625. (*SideOf* formulae are assigned a probability of 1, because they are conditional on *Region* formulae, and can be ignored in the calculation of explanatory value.) In other words, $\square_1$ has greater explanatory value than $\square_2$ because it has fewer noise terms, and properly explains more of the data.

Of course, as we can see from Figure 2, hypothesis $\square_2$ is actually nearer the mark. There is indeed a surface bounded by lines 3 and 7, and this surface constitutes one face of a cuboidal object. The correct hypothesis comes out on top, in this example, after stage two of the interpretive process, which abduces cuboidal objects from trapezoidal regions. However, for this to work, the abductive mechanism requires the ability to test the expectations of a hypothesis.

3 Logic Programming Implementation

When the theoretical framework of [Shanahan, 2002] was published, the accompanying implementation was still under construction, so no details of it were given. A full implementation now exists, and this forms the basis of an ongoing project to implement the theory of active visual perception presented below.

The core of the implementation is an abductive logic programming meta-interpreter [Kakas, *et al*., 1992]. The abductive meta-interpreter employs largely standard techniques, with the following exception. Given a list of data items G_1 to G_n, the procedure does not find an explanation for G_1, then an explanation for G_2, and so on, as a standard abductive procedure would. Rather, it finds an explanation E for G_1, and before moving on to consider G_2 fully, it finds out how much of G_2 to G_n is already explained by E. In this way, unpromising hypotheses that explain very little of the sensor data can be weeded out early. (The same issue is tackled in a somewhat different way by Poole [1993].)

The core abductive meta-interpreter is embedded in a procedure that carries out low-level tests on the raw image to confirm or disconfirm the expectations of the hypotheses under construction. The procedure uses the following algorithm. First, a list of promising hypotheses is formed. These are then ranked according to explanatory value. Then, each of the best M hypotheses has its expectations confirmed or discomfirmed. The hypotheses are then ranked again, according to their updated explanatory values, and the best N are picked. The parameters M and N must be adjusted to ensure efficiency without loss of useful hypotheses.

The present implementation has two components. The abductive meta-interpreter is written in Prolog. But all the low-level image processing is carried out by a C++ program, which also has executive control. As the implementation currently stands, the user interacts directly with the C++ program via a GUI, and selects a region of an image for attention. A Sobel edge detector then finds all straight line edges over a certain length within the selected region. These are passed to the Prolog-based abductive meta-interpreter, which finds the best explanations according to the method outlined above.

In the current implementation, the expectations of a hypothesis are straight line edges, represented by their end-points. In order to confirm/disconfirm an expectation, the abductive meta-interpreter forwards these end-points to a routine in the C++ program, which runs a highly sensitive edge-checking algorithm to gauge whether there might be an edge between them. This algorithm picks up far fainter edges than the original edge finder, but is only ever run on a pre-selected pair of points where there is already the expectation of an edge.

The existing implementation works well for simple examples, such as the one in Figure 2. But it is only a proof-of-concept, and there is a great deal about it that might be varied. For example, the basic techniques can easily be adapted to take their input from other kinds of low-level image processing technique, such as colour-based segmentation, stereo matching, or optical flow. The use of each of these is under investigation. In addition, since sensor fusion can be accommodated within the theoretical framework of abduction, work is also being carried out to see how multiple sources of low-level data might be exploited.

4 Abduction and Active Perception

One of the motivations for active perception is the prodigious quantity of raw data available to a robot's sensors, especially to vision. Only a small portion of that data can be processed in real-time. Therefore, techniques are required that will focus attention on potentially the most useful data. For example, most mammals instinctively move their eyes and head so as to rapidly centre the visual field on a potential source of danger. The selection of data can also be made in such a way as to facilitate the piece-by-piece construction of an increasingly complete and accurate model of the world, and this can be done through *expectation.*

In essence, perception, cognition, and action must act in concert to carry out what philosophers of science call *hypothetico-deductive* reasoning. This comprises three steps. First, the most promising hypotheses are formed that might explain the sensor data. This process was formalised in Section 2. Second, the expectations of competing hypotheses are deduced. Third, those expectations are tested by carrying out experiments. The value of a hypothesis is reduced if its expectations are unfulfilled, while the value of a hypothesis increases if its expectations are met.

In the example of Section 2, the hypothesis that there is a long two-colour block that includes a face bounded by lines 3 and 7 is in competition with the hypothesis that there is a short white block that includes a face bounded by lines 1 and 3. These hypotheses have differing expectations. According to the first (correct) hypothesis, for example, line 2 should extend further to the left to meet line 7. According to the second (incorrect) hypothesis, there should be a nearly vertical line running down from the point where lines 5 and 6 meet. Various actions could be performed to test these expectations. The robot could nudge the object to see it from a new angle, it could move its head to get a better view, or (as discussed in [Shanahan, 2002]) it could simply locally adjust the threshold of the edge detection procedure to make it more sensitive.

The emphasis of the present paper is on robot actions that bring about *informative changes* to its environment — the act of nudging the block is an example — and the aim is to devise a logic-based theory of perception that takes such actions into account. But to accommodate informative change, the basic abductive prescription of Section 2 must be extended fairly dramatically. Three distinct modifications are required.

1. In the preceding abductive definition of visual perception, the background theory Σ describes a static world. In the extended version, it must allow for change, actions, and events.
2. The preceding definition produces a single hypothesis from a single frame. The extended version must generate an ever-improving series of hypotheses in response to a continuous sequence of frames.
3. For the preceding definition, the issue of tracking doesn't arise. By contrast, in the extended version, an explanation of the sensor data has to incorporate hypotheses about likely correspondences between features in successive frames.

To begin with, let's see how the basic abductive definition can be made to take account of actions and events. This requires the adoption of a logic-based formalism for reasoning about action [Shanahan, 1997a]. This paper uses the event calculus, which has already been successfully applied to robot perception [Shanahan, 1996a], though not to vision. The ontology of the event calculus includes actions, fluents, and time points. The basic axioms, adapted from [Shanahan, 1999], are as follows.

$$HoldsAt(f,t) \leftarrow Initially(f) \wedge \neg Clipped(0,f,t) \qquad \text{(EC1)}$$

$$\begin{aligned} & HoldsAt(f,t_2) \leftarrow \qquad \text{(EC2)} \\ & \quad Happens(e,t_1) \wedge Initiates(e,f,t_1) \wedge t_1 < t_2 \wedge \\ & \qquad \neg Clipped(t_1,f,t_2) \end{aligned}$$

$$\begin{aligned} & Clipped(t_1,f,t_3) \leftrightarrow \qquad \text{(EC3)} \\ & \quad \exists e,t_2\, [Happens(e,t_2) \wedge t_1 < t_2 < t_3 \wedge \\ & \qquad [Terminates(e,f,t_2) \vee Releases(e,f,t_2)]] \end{aligned}$$

The formula $HoldsAt(f,t)$ means that fluent f is true at time t, $Initially(f)$ means that f is true at time 0, $Happens(e,t)$ means that an action of type e occurs at time t, $Initiates(e,f,t)$ means that fluent f starts to hold after an action of type e at time t, $Terminates(e,f,t)$ means that f ceases to hold after an action of type e at time t, and $Releases(e,f,t)$ means that fluent f is not subject to the common sense law of inertia after an action of type *e* at time *t*. The frame problem is overcome by applying predicate completion to *Happens*, *Initiates*, *Terminates*, and *Releases* [Shanahan, 1997a].

The background theory □ now incorporates axioms (EC1) to (EC3), and includes a collection of *Initiates*, *Terminates*, and *Releases* formulae that describe the effect of robot actions on the world, and the consequent impact of the world on the robot's sensors. The basic abductive characterisation of perception can then be upgraded as follows. Given,

- a narrative □ of robot actions expressed in terms of *Happens* formulae, and
- a description □ of sensor data expressed in terms of *HoldsAt* formulae,

the task of perception is to find descriptions □ of objects in the environment such that,

$$\square \square \square \square \square \models \square.$$

As usual, □ must be consistent and can mention only abducible predicates. Competing □s can be ordered using the measure of explanatory value defined in Section 2.

5 Abducing Visual Events

With its capacity to represent robot actions and their effects, the modified abductive formulation of Section 4 can encompass various forms of active perception. In [Shanahan & Witkowski, 2001], for example, a similar definition is used to characterise the perception of a mobile robot with infra-red proximity sensors as it explores an unknown office-like environment. In this case, the sensor events are very simple. — The forward proximity sensors go high when the robot encounters an obstacle, the left-hand proximity sensors go low when it passes an open doorway, and so on. — The detection and classification of *visual* sensor events, on the other hand, is more subtle, and presents a far greater challenge to a logic-based approach. Shortly, we'll see an example of the use of the event calculus to define a class of visual events using the idea of an object's *aspect graph* [Koenderink & van Doorn, 1979]. But the next step is to tailor the foregoing treatment of perception to sequences of camera frames.

First, rather than a single hypothesis, the abductive process must generate a sequence of hypotheses $\square_1,\square_2,...,\square_n$ in response to a sequence of frames $\square_1,\square_2,...,\square_n$. Second, since there is nothing in the raw data to link features in one frame with features in the next, the abductive definition has to be extended to deal with these inter-frame correspondences explicitly. Fortunately, the help of an off-the-shelf tracking algorithm can be enlisted here, such as the feature-based KLT tracker [Shi & Tomasi, 1994]. The tracker can supply a list of apparent correspondences to prime the abductive process. However, the final responsibility for deciding whether the apparent identity of two features is real or not should be left to the abduction.

These considerations lead to the following definition. As before, the background theory □ must contain axioms (EC1) to (EC3), and has to describe both the effects of robot actions on the world and the impact of the world on the robot's sensors, in this case its vision system. However, to strengthen the effect of formulae that constrain the permissible hypotheses, which up to now have been included in □, they can now be conjoined in a formula $\square_{IC}$, which appears on the right-hand-side of the turnstile. This a standard manoeuvre in the abductive logic programming literature, where such formulae are called *integrity constraints* [Kakas, *et al.*, 1992]. Given,

- a hypothesis $\square_i$ describing objects in the world,
- a narrative □ of robot actions,
- a description □ of a single frame of raw image data, and
- a description □ of apparent correspondences between features in □ and features in $\square_i$,

the task of perception is to find $\square_{i+1}$, and $\square_C$, where,

- $\square_{i+1}$ is a description of objects in the world, and
- $\square_C$ is a set of hypothesised correspondences between features in □ and features in $\square_{i+1}$

such that,

- $\square_{i+1} \models \square_i$, and
- $\square \square \square \square \square_{i+1} \square \square_C \models \square \square \square \square \square_{IC}$.

The usual restrictions apply to $\square_{i+1}$, and $\square_C$. They must be consistent, and can mention only abducible predicates. Note that the form of abduction defined in Section 2 is a special case of the new definition, in which $\square_i$, $\square_C$, □, and $\square_{IC}$ are all empty.

In the next section, this definition is used to capture a form of active visual perception in which a humanoid robot manipulates objects in its workspace, thereby causing them to present a series of different aspects to the robot's view. Because only certain sequences of aspects are possible for an object of a given shape, by nudging it around, the expectations of competing hypotheses about the nature of the object can be tested.

6 Aspect Graphs and the Event Calculus

In [Koenderink & van Doorn, 1979], attention was drawn to the fact that slight rotations of an object typically do not bring about a qualitative change in the aspect it presents to the viewer. Indeed, for simple regular-sided 3D shapes, the number of qualitatively distinct aspects they can present is small, and only certain transitions between these aspects are possible. These can be represented in a graph, known as the shape's *aspect graph*, in which qualitatively distinct aspects are the nodes and the possible transitions between them are the arcs (Figure 3). In the terminology of qualitative spatial reasoning, an aspect graph represents the structure of a *conceptual neighbourhood* [Cohn & Hazarika, 2001].

Aspect graphs are well-suited to the representation of shape in the context of an upper-torso humanoid, since knowledge of each object in such a robot's workspace is acquired through the aspect it presents to the robot's camera, and that aspect can change as a result of the the robot's arm movements.

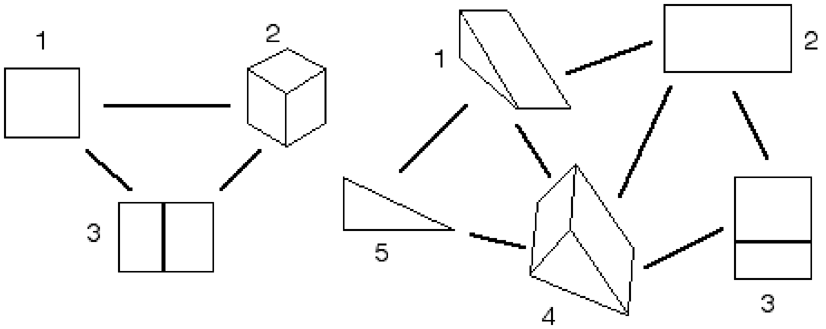

Figure 3: Partial Aspect Graphs of a Cuboid and a Wedge

In what follows, the formula $Arc(s,y_1,y_2)$ means that there is an arc between aspect types y_1 and y_2 in the aspect graph representing shape s. In particular, the shapes *Cuboid* and *Wedge* will be assumed. The cuboid or wedge aspect types labelled *n* in Figure 3 are denoted, respectively, by the terms *Wedge*(*n*) or *Cuboid*(*n*).

Section 2 of the present paper presented an abductive account of the transition from edges in an image to hypothesised regions. This is the first stage of abduction in what is in fact a three-layer process. The next task is to extend this account to the second layer — the transition from regions to aspects. Then, with the introduction of a temporal dimension, the formalisation can be further extended to the third layer — the transition from sequences of aspects to 3D shapes.

In anticipation of the introduction of time, predicates that have up to now been atemporal must now be regarded as fluents, and are therefore reified using the *HoldsAt* predicate. For example, where previously we wrote *Region*(*r*), we now write *HoldsAt*(*Region*(*r*),*t*). First-layer axioms such as (L1) and (L2) in Section 2 must be amended accordingly. Here's an example of an axiom that will be included in the second-layer background theory □. The second-layer description □ of the raw data comprises formulae of the form *HoldsAt*(*Region*(*r*),*t*). These are supplied by the first layer of abduction. In effect, the disjunction of all the □s output from the first layer becomes the □ input to the second layer.

$$\begin{aligned}&HoldsAt(Region(r),t)\ \square && \text{(R1)}\\ &\quad \square\, x,v,a\ [HoldsAt(Occupies(x,v),t)\ \square\\ &\qquad HoldsAt(Aspect(x,a),t)\ \square\ RegionOf(r,a)]\end{aligned}$$

The fluent $Occupies(x,v)$ means that object x occupies the 3D volume of space v. In effect, the assertion that there exists an x and v such that $HoldsAt(Occupies(x,v),t)$ means that object x exists at time t in a physical sense (in addition to the purely logical sense). The fluent $Aspect(x,a)$ means that object x is presenting aspect a to the viewer. Note that a is an aspect instance not an aspect type. Used abductively, this axiom explains a visible region in terms of a physical object and the 2D aspect it instantaneously presents.

In addition, the following formula will be among the integrity constraints in $\square_{IC}$. The formula $Type(a,y)$ means that aspect a is of type y. The formula $Shape(x,s)$ means that object x has shape s, where a shape is represented as an aspect graph.

$$\begin{aligned}&HoldsAt(Aspect(x,a),t)\ \square && \text{(R2)}\\ &\quad \square\, s, y_1,y_2\ [Type(a,y_1)\ \square\ Shape(x,s)\ \square\ Arc(s,y_1,y_2)]\end{aligned}$$

The *Shape* predicate is made abducible at the second layer of abduction, alongside *Occupies* and *Aspect*. Without the addition of this integrity constraint, hypotheses are not obliged to suggest a shape for each object they posit. Indeed, as Pylyshyn argues, it is sometimes necessary to provide "a direct (preconceptual, unmediated) connection between elements of a visual representation and certain elements in the world [that] allows entities to be referred to without being categorized or conceptualized" [Pylyshyn, 2001]. However, the inclusion of (R2) will be assumed in what follows.

The role of the third layer of abduction is to make hypotheses about a moving object's overall 3D shape based on its changes of aspect. To see how this final layer works, we'll formalise an example of active perception with a humanoid robot, in which the robot nudges an object in order to see it from a new angle. In what follows, the terms $Move(x)$ and $Stop(x)$ respectively denote the actions of setting object x in motion and stopping it. The fluent $Changing(x,a)$ holds when object x is in motion and presenting a changing aspect a. (In general, if an object is presenting a changing aspect to the viewer, then some of its faces are shrinking while others are expanding.)

$$\begin{aligned}&Initiates(Move(x),Changing(x,a),t)\ \square && \text{(A1)}\\ &\quad HoldsAt(Aspect(x,a),t)\end{aligned}$$

$$Terminates(Stop(x),Changing(x,a),t) \qquad \text{(A2)}$$

While an object is stationary, the *Aspect* fluent is subject to the common sense law of inertia. But between a *Move* and a *Stop* action, the *Aspect* fluent becomes non-inertial, and its value becomes dependent (in a trivial way) on the *Changing* fluent.

$$Releases(Move(x),Aspect(x,a),t) \quad \text{(A3)}$$

$$Initiates(Stop(x),Aspect(x,a),t) \; \square \; HoldsAt(Aspect(x,a),t) \quad \text{(A4)}$$

$$HoldsAt(Aspect(x,a),t) \; \square \; HoldsAt(Changing(x,a),t) \quad \text{(A5)}$$

If an object is presenting a changing aspect, then it is heading towards an aspect transition. Let *Trans*(*x*) denote an event that occurs when object *x* undergoes a qualitative change in aspect. The time at which this transition occurs will depend on the exact trajectory of the object's motion. However, this trajectory is neither knowable, nor relevant at a qualitative level of abstraction.

For this reason, a non-inertial "determining fluent" is used to substitiute for the details of the trajectory [Shanahan, 1999]. In general, the formula *ByChance*(*f*,*t*) means that the determining fluent *f* holds at time *t*. In an abductive context, the *ByChance* predicate is made abducible, allowing actions with non-deterministic effects to be handled correctly, or, as in the present case, events with a non-deterministic time of occurrence. In the following axiom, the term *Delay*(*x*,*d*) denotes a determining fluent that holds when the time to the next aspect transition for object *x* is *d*.

$$\begin{aligned} &Happens(Trans(x),t_2) \; \square \quad \text{(A6)} \\ &\quad \square \, e,t_1,x,a,d \; [Happens(e,t_1) \; \square \\ &\qquad Initiates(e,Changing(x,a),t_1) \; \square \\ &\qquad\quad \square \; Clipped(t_1,Changing(x,a),t_2) \; \square \\ &\qquad\qquad ByChance(Delay(x,d),t_1) \; \square \; t_2 = t_1 + d] \end{aligned}$$

The effect of a *Trans*(*x*) action is that object *x* presents a new aspect to the viewer. The particular new aspect presented depends on various factors that are, again, neither knowable nor relevant at a qualitative level, such as the object's precise initial orientation and location. Accordingly, another determining fluent is used to conjure away these details in the formalisation. The determining fluent *AtTrans*(*x*,*a*) holds if object *x* is on the point of transition to aspect *a*.

$$Initiates(Trans(x),Changing(x,a),t) \; \square \; ByChance(AtTrans(x,a),t) \quad \text{(A7)}$$

$$Terminates(Trans(x),Changing(x,a),t) \; \square \; HoldsAt(Changing(x,a),t) \quad \text{(A8)}$$

The use of this determining fluent may seem like a form of "cheating" here. But the trick is to impose further constraints on the permissible aspect transitions based on the set of aspect graphs of known object shapes. Here's an example of such a constraint, which is to be included in $\square_{IC}$.

$$\begin{aligned} &Initiates(Trans(x),Changing(x,a_2),t) \; \square \quad \text{(A9)} \\ &\quad \square \, a_1,y_1,y_2,s \; [HoldsAt(Aspect(x,a_1),t) \; \square \\ &\qquad Type(a_1,y_1) \square \; Type(a_2,y_2) \; \square \\ &\qquad\quad Shape(x,s) \; \square \; Arc(s,y_1,y_2)] \end{aligned}$$

When these axioms are used abductively, the *Shape* predicate is again made abducible, and the function of this integrity constraint is to reduce the set of shapes that can be hypothesised for a given object to those that are consistent with some known aspect graph. The upshot is that, with each new aspect an object presents, the perceptual system gets closer to identifying its shape uniquely.

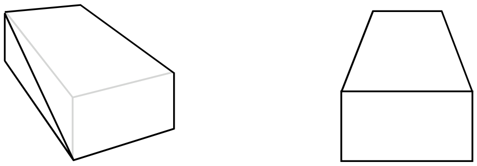

Figure 4: Two Views of a Block

Now let's look at an example of the third layer at work. (In what follows, the issue of tracking and finding inter-frame correspondences will be set aside, for the sake of simplicity.) Figure 4 shows, in an idealised and exaggerated way, two successive frames depicting a block from different angles, both taken after the robot has executed a *Move* action. In the left-hand frame, taken at time T_1, part of the block is obscured by a shadow which has generated a false edge along the side of the block and hidden three of the other edges. Suppose there are two known shapes — a cuboid and a wedge. Let $\square_{1,1}$ and $\square_{1,2}$ correspond to two second-layer hypotheses that explain this frame in terms of, respectively, a cuboid and a wedge. The input $\square$ to the third layer will be $\square_{1,1} \quad \square_{1,2}$. Suppose $\square_{1,1}$ includes the following formulae.

$$HoldsAt(Occupies(X,V_1),T_1)$$

$$HoldsAt(Aspect(X,A_1),T_1)$$

$$Type(A_1,Cuboid(2))$$

$$Shape(X,Cuboid)$$

And suppose $\square_{1,2}$ includes the same first two formulae, plus the following.

$$Type(A_1,Wedge(1))$$

$$Shape(X,Wedge)$$

Given this frame alone, the third layer of abduction trivially generates two corresponding hypotheses using axiom (EC1). The first hypothesis $\square_{1,1}$ includes the following formulae.

$Initially(Occupies(X,V_1))$

$Initially(Aspect(X,A_1))$

$Type(A_1,Cuboid(2))$

$Shape(X,Cuboid)$

The second hypothesis $\square_{1,2}$ is analogous, but for a wedge shape. Because of the shadow, neither hypothesis is assigned a significantly higher explanatory value than the other. As we'll see, when the second frame, taken at time T_2, is taken into account, the ambiguity is resolved. Let the new input to the third layer be $\square = \square_{2,1} \quad \square_{2,2}$, where $\square_{2,1}$ and $\square_{2,2}$ correspond to two second-layer hypotheses that explain the second frame, again in terms of, respectively, a cuboid and a wedge. So $\square_{2,1}$ will include the following formulae.

$HoldsAt(Occupies(X,V_2),T_2)$

$HoldsAt(Aspect(X,A_2),T_2)$

$Type(A_2,Cuboid(3))$

$Shape(X,Cuboid)$

$\square_{2,2}$ will be analogous, but for a wedge shape, and in particular will include the following formula.

$Type(A_2,Wedge(3))$

$Shape(X,Wedge)$

Now consider a third-layer hypothesis $\square_{2,1}$ that includes all the formulae in $\square_{1,1}$ plus the folrmulae,

$ByChance(Delay(X,\square),T_1)$

$ByChance(AtTrans(X,A_2),T_1+\square)$

for some $\square$ such that $T_1+\square < T_2$. Assuming $\square$ and $\square_{IC}$ contain all the relevant formulae from above and $\square$ describes the robot's *Move* action, then $\square_{2,1}$ is an explanation for $\square$, according to the definition in Section 5. In other words, we have,

$$\square \; \square \; \square \; \square \; \square_{2,1} \models \square \; \square \; \square_{IC}.$$

In particular, thanks to axiom (A6), $\square_{2,1}$ entails the occurrence of a $Trans(X)$ event at time $T_1+\square$, which initiates the fluent $Changing(X,A_2)$, where the type of A_2 is $Cuboid(3)$. By contrast, the competing hypothesis $\square_{2,2}$, in which X is a wedge, is *not* an explanation. This is because, while $\square_{2,1}$ is consistent with the constraint (A9), $\square_{2,2}$ is not. This, in turn, is because the transition from $Wedge(1)$ to $Wedge(3)$ is not permitted by the aspect graph.

7 Concluding Remarks

There are several major strands of further research in pursuit of the theoretical ideas in this paper. First, the repertoire of shapes that can be handled by the background theory $\square$ must be widened. The large body of extant literature on aspect graphs can be exploited here. Other kinds of conceptual neighbourhood diagram can also be used. For example, in [Randell, *et al.*, 2001], an occlusion calculus is developed which maps three-dimensional bodies and viewpoints to their corresponding two-dimensional images, and identifies an exhaustive and pairwise disjoint set of visual occlusion relations. When worked into the event calculus, the resulting conceptual neighbourhood diagram can be used to abduce the spatial arragement of bodies in space to explain the spatial relations on their corresponding detected images.

In addition to working on these knowledge representation issues, a larger body of off-the-shelf low-level vision techniques (such as stereo matching, optical flow, and colour-based segmentation) must be recruited to supply cues for the high-level abductive process. Abductive techniques for effecting a logic-based merging of low-level data from several such sources are also under investigation, drawing on existing work in the field of sensor fusion.

Although the emphasis of the present paper is theoretical, implementation of the framework described here is well underway, along the lines indicated in Section 3. At the time of writing, the high-level abductive reasoning component is complete and working well with single frames. Work is ongoing to get the system to assimilate multiple frames, and to close the loop so that the robot can nudge the objects it has recognised autonomously.

Acknowledgments

This work was supported by EPSRC grant GR/S02891, "Spatial Reasoning and Perception in a Humanoid Robot". Thanks to Mark Witkowski.

References

Aloimonos, J., Weiss, I. and Bandyopadhyay, A. 1987. Active Vision. In *Proc. 1st International Conference on Computer Vision*, pp. 35–54.

Ballard, D.H. 1991. Animate Vision. *Artificial Intelligence*, vol. 48, pp. 57–86.

Cohn, A.G. and Hazarika, S. 2001. Qualitative Spatial Reasoning: An Overview. *Fundamenta Informatica*, vol 46(1–2), pp. 2–32.

Cohn, A.G., Magee, D., Galata, A., Hogg, D. and Hazarika, S. 2003. Towards an Architecture for Cognitive Vision Using Qualitative Spatio-Temporal Representations and Abduction. In *Spatial Cognition III*, eds. C.Freska, C.Habel & K.F.Wender, Springer, pp. 232–248.

Hazarika, S. and Cohn, A.G. 2002. Abducing Qualitative Spatio-Temporal Histories from Partial Observations. In *Proc. 2002 Knowledge Representation Conference (KR 2002)*, pp. 14–25.

Kakas, A.C., Kowalski, R.A. & Toni, F. 1992. Abductive Logic Programming. *Journal of Logic and Computation*, vol 2(6), pp. 719–770.

Koenderink, J.J. and van Doorn, A.J. 1979. The Internal Representation of Solid Shape with Respect to Vision. *Biological Cybernetics*, vol. 32, pp. 211–216.

Lespérance, Y., Levesque, H.J., Lin, F., Marcu, D. Reiter, R. and Scherl, R.B. 1994. A Logical Approach to High-Level Robot Programming: A Progress Report. In *Control of the Physical World by Intelligent Systems: Papers from the 1994 AAAI Fall Symposium*, ed. B.Kuipers, New Orleans, pp. 79–85.

Levesque, H. 1996. What Is Planning in the Presence of Sensing? In *Proc. 1996 American Association for Artificial Intelligence Conference (AAAI 96)*, pp. 1139–1146.

Pirri, F. and Finzi, A. 1999. An Approach to Perception in Theory of Actions: Part I. *Electronic Transactions on Artificial Intelligence*, vol 3, pp. 19–61.

Pirri, F. and Romano, M. 2002. A Situation-Bayes View of Object Recognition Based on SymGeons. In *Proc. Third International Cognitive Robotics Workshop.*

Poole, D. 1993. Logic Programming, Abduction and Probability: A Top-Down Anytime Algorithm for Estimating Prior and Posterior Probabilities. *New Generation Computing*, vol. 11, pp. 377–400.

Poole, D. 1995. Logic Programming for Robot Control. In Proc. *1995 International Joint Conference on Artificial Intelligence (IJCAI95)*, pp. 150–157.

Pylyshyn, Z.W. 2001. Visual Indexes, Preconceptual Objects, and Situated Vision. *Cognition*, vol. 80, pp. 127–158.

Randell, D., Witkowski, M. and Shanahan, M.P. 2001. From Images to Bodies: Modelling Spatial Occlusion and Motion Parallax. In *Proc. 2001 International Joint Conference on Artificial Intelligence (IJCAI 01)*, pp. 57–63.

Remolina, E. and Kuipers, B. 2004. Towards a General Theory of Topological Maps. *Artificial Intelligence*, vol. 152, pp. 47–104.

Santos, P. and Shanahan, M.P. 2002. Hypothesising Object Relations from Image Transitions. In *Proc. 2002 European Conference on Artificial Intelligence (ECAI 02)*, pp. 292–296.

Scherl, R.B. and Levesque, H. 2003. Knowledge, Action, and the Frame Problem. *Artificial Intelligence*, vol. 144(1–2), pp. 1–39.

Shanahan, M.P. 1996a. Robotics and the Common Sense Informatic Situation. In *Proc. 1996 European Conference on Artificial Intelligence (ECAI 96)*, pp. 684–688.

Shanahan, M.P. 1996b. Noise and the Common Sense Informatic Situation for a Mobile Robot. In *Proc. 1996 American Association for Artificial Intelligence Conference (AAAI 96)*, pp. 1098–1103.

Shanahan, M.P. 1997a. *Solving the Frame Problem: A Mathematical Investigation of the Common Sense Law of Inertia*. MIT Press.

Shanahan, M.P. 1997b. Noise, Non-Determinism and Spatial Uncertainty. In *Proceedings 1997 American Association for Artificial Intelligence Conference (AAAI 97)*, pp. 153–158.

Shanahan, M.P. 1999. The Event Calculus Explained. In *Artificial Intelligence Today*, eds. M.J.Wooldridge & M.Veloso, Springer-Verlag Lecture Notes in Artificial Intelligence no. 1600, Springer-Verlag (1999), pp. 409-430.

Shanahan, M.P. 2000. Reinventing Shakey. In *Logic-Based Artificial Intelligence*, ed. J.Minker, Academic Press, pp. 233–253.

Shanahan, M.P. 2002. A Logical Account of Perception Incorporating Feedback and Expectation. In *Proc. 2002 Knowledge Representation Conference (KR 02)*, pp. 3–13.

Shanahan, M.P. and Witkowski, M. 2001. High-Level Robot Control Through Logic. In *Intelligent Agents VII*, Springer, 2001, pp. 104–121.

Shi, J. and Tomasi, C. 1994. Good Features to Track. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR 94)*, pp. 593–600.