Abductive Visual Perception with Feature Clouds

David Randell and Mark Witkowski

Department of Computing
Imperial College London
180 Queen's Gate
London SW7 2AZ, U.K.
{d.randell, m.witkowski}@imperial.ac.uk

Abstract

This paper describes a logical approach to embodied perception and reasoning in the context of Cognitive Robotics that use feature clouds to encode an explicit 3D description of bodies of arbitrary structural complexity. We extend and apply the principles of abductive perception in order to provide robots with an explicit, flexible, and scalable three-dimensional representation of the world for object recognition, localisation and general task execution planning. We show how feature clouds require neither a complex logically formulated geometrical-based description of the intended modelled domain; nor are they necessarily tied to any particular type of feature-detector. Feature clouds provide the means to (i) unify and encode qualitative, quantitative and numerical information as to the position and orientation of objects in space; (ii) encode viewpoint and resolution dependent information; and (iii) when embedded within a hypothetico-deductive reasoning framework, provide the means to integrate psychophysical and other domain-independent constraints.

1 Introduction

This paper describes a logical approach to embodied perception and reasoning in the context of Cognitive Robotics that use *feature clouds* to encode an explicit 3D description of bodies of arbitrary structural complexity. We extend and apply the principles of abductive perception developed by Shanahan [2002, 2005; Shanahan and Randell, 2004] in order to provide robots with an explicit, flexible, and scalable three-dimensional representation of the world for object recognition, localisation and general task execution planning. We show how feature clouds require neither a complex logically formulated geometrical-based description of the intended modelled domain [Shanahan, 2004; Randell, 2005]; nor are they necessarily tied to any particular type of feature-detector. Feature clouds provide the means to (i) unify and encode qualitative, quantitative and numerical information as to the position and orientation of objects in space; (ii) encode viewpoint and resolution dependent information; and (iii) when embedded within a general abductive-driven *hypothetico-deductive* reasoning framework, provide the means to integrate low-level psychophysical-based cues and *domain-independent* constraints.

It is not our intention to create a specific computer vision algorithm, but rather to develop theoretical underpinnings and to establish a formal framework to describe processes of perception, to which detailed algorithms and techniques as used in computer vision and computer graphics may later be incorporated. This framework presupposes that the robot exists in a three dimensional world, with objects that conform to various commonsense notions of physical existence, which may include occupancy of volumetric space, having opacity and rigidity and being solid, in the sense that distinct objects do not volumetrically overlap. These assumptions will be taken as holding here. As long as incoming sensor data satisfies these working assumptions, the underlying model can be interpreted in terms of physical embodiment, whether or not the model is implemented on a physical robot. In this respect, we use a Webots™ robot simulator [Cyberbotics, 2006], providing two-dimensional images of three-dimensional objects arranged around a virtual robot. The Webots™ simulator provides a useful prototyping environment in anticipation of implementing the representation and methodology used here on real-world robots with machine vision sensors.

The remainder of the paper proceeds as follows. In section 2 we present an overview of abductive perception and establish a scenario for the formalism that will be developed later in the paper. Section 3 details the notion of a feature cloud and how variations in the effect of objects on the detector apparatus under differing viewpoints and conditions is managed. Section 4 establishes the formalism used in the remainder of the paper. Section 5 describes the essential constraints between objects. Section 6 considers the role of spatial relationships between objects. Section 7 considers the management of evidential support for the hypothetico-deductive framework. Section 8 develops a simple worked example to illustrate the points made.

2 Abductive Perception an Overview

The abductive treatment of perception developed here is a variation and extension of that presented in [Shanahan, 1996 and 1997]. There we have a background theory Σ comprising a set of logical formulae that describe the effect of a robot's actions on the world, and a set of formulae that describe the effect of the world on the robot's sensors. Then, given a description, Γ, derived from the robot's sensor data, the abductive task is to generate a consistent set of explanations, Δ, such that $\Sigma \wedge \Delta \models \Gamma$.

Abduction does not necessarily guarantee a single solution in the construction of Δ. In [Shanahan, 2002, 2005; and Shanahan and Randell, 2004] this is addressed by rank-ordering competing hypotheses in terms of their assigned *explanatory value*s and selecting those with the highest explanatory content. The explanatory value reflects the extent to which the sensor data evidence presented in Γ supports any particular interpretation of an object derived from the background theory. While we use a similarly defined explanatory value measure, we also introduce two additional measures, *distinctiveness value* and *rank order*. Unlike the single *a posteriori* explanatory value used by Shanahan, we explicitly factor out these three measures, and use all three separately in the perceptual process. Thus the distinctiveness value defined on feature types can be compared to the task of measuring the *salience* of features that appears in other machine-vision based research (e.g. [Itti and Koch, 2001]). These additional measures guide the hypothesis generation process discussed later.

Key to this process is the deployment of "detectors", devices that make these assertions into Γ when the specific conditions they are tuned to occur. In the scenario we describe here, these detectors are assumed to be derived from low-level vision processing operations. Such operation might include line finding or edge detection routines, or, more usefully, the detector will respond to some distinctive, though not necessarily a unique, property of the physical object being observed (see the work of [Lowe, 2004] and [Schmid and Mohr, 1997] for recent advances in this area). Detected features must be reliably localizable onto the image plane (as indicated in figure 1), but need not be invariant on rotation or scale. It is our intention to allow a wide range of detector types to be assimilated within the logical framework described. We recognise that objects in the world may give rise to different effects on the detectors under differing circumstances, such as when viewed from different distances or from varying angles. We refer to these variations as the *appearances* of a feature.

Each different appearance associated with a detector is assigned a *Type* and a location on the image plane, which will serve to identify the source of each Γ assertion within the system. The ability of detectors to respond differentially to different features is central to this process, and we define various measures to quantify and exploit such variation. It may be noted that some features, such as line segments, are common to many objects, and, as such, provide little discriminatory evidence. Low-level features may be composited into *compound features*, which typically increases their discriminatory power.

The new abductive strategy for processing sensor data proceeds as follows and is illustrated in figure 1. Firstly, distinctive elements of Γ are matched to object descriptions in Σ to generate a set of candidate hypotheses in Δ that may explain the sensor data. In this respect individual sensor data items may be said to *afford* (by analogy with [Gibson, 1979]) their own explanation. Based on this partial evidence, each inferred host-object is hypothesised to occupy a specific place in the space about the robot. Then using deduction coupled with the manipulation of 3D linear-transforms (as prediction) the ramifications of this projection are expanded. The feature cloud is used to determine the expectations of other features associated with the hypothesised object(s) in question. The sensor data is then re-consulted to refine the explanatory value by determining the extent to which each hypothesis is supported by the observed data. On the basis of this, hypotheses with low explanatory value are rejected (and sensor data items they would have explained are released, though still in need of an explanation). The process is repeated until all the sensor data elements of Γ have a coherent explanation, and where all the ground hypotheses in Δ satisfy all the constraints applicable to the domain.

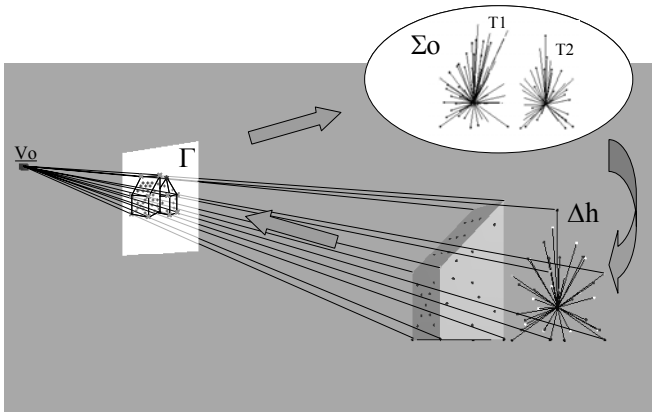

Figure 1: The main perceptual cycle

The background theory Σ has two parts. First we factor out the generic object descriptions which are ultimately encoded as feature clouds (denoted by Σo), and secondly, we factor out a set of domain-independent constraints which both restrict and determine how and whether hypotheses can coexist in Δ (denoted by Σc). First amongst these is the commonsense belief that two volumetric bodies may abut but not overlap (i.e. they may not share a volume in common). Δ is similarly partitioned, first into sentences describing many alternative hypothesised models (Δh), and Δi, the single model representing the stable interpretation of the scene, which may then be used for planning, problem solving or other tasks.

A cognitive agent cannot deny or disprove sensor data, it can only interpret it in a manner consistent with the use it intends to make of the data. It follows from this, that the

abductive process leads to three primary outcomes when interpreting sensor data: (i) to accept the interpretation based on the prevailing model, (ii) to reject or ignore the interpretation as, for example, stemming from sensor noise, or arising from inconsistency with the assumed domain model or (iii) update the existing domain model to accommodate the new previously unassimilated sensor data (Σo). This paper will concern itself only with the first case, where a description exists in Σo and an interpretation may be placed on the incoming data consistent with that description.

3 Feature Clouds

A feature cloud is a data structure that encodes a heterogeneous and spatially distributed set of sensor-detected features and may be contrasted to other model-based representations in visual perception-based applications such as generalised cylinders [Marr, 1982], symgeons [Pirri, 2005, for instance], or superquadrics [Chella *et al*, 2000]. Each feature is individually mapped to a *position vector* and local coordinate system. The cloud is partitioned into subsets of features that are pre-assigned to a set of inferred, volumetric regions. The whole takes on the form of a hierarchically organised tree-structure where the subdivision of a host body into sub-parts proceeds until all their named features and their viewpoint-dependent appearances eventually appear as terminal leaf-nodes.

Each feature cloud is represented by sets of *vector pencils;* where each pencil comprises a set of straight-line segments intersecting at a single point. Pencils mapping features to their appearances are interpreted as *lines of sight* fanning out into space. As each viewpoint also acts as the origin of another vector pencil whose end-points potentially locate a set of features, the overall geometrical form of an object with its set of features and their viewpoint indexed manifold appearances can be likened to a stellated polyhedron with the vertices mapping to the view points.

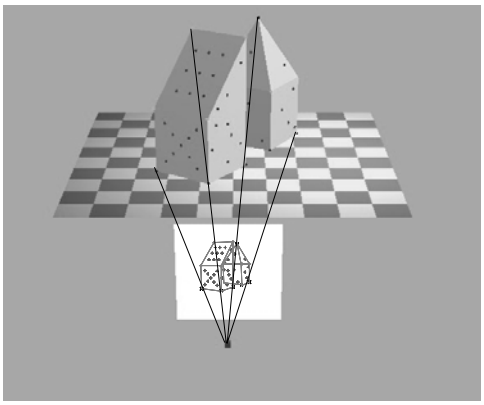

Figure 2: A scene view

Figure 2 shows a pair of towers being imaged from a viewpoint with the projected image plane of the camera shown straddling between the two. The white image plane area shows the locations at which various feature points arise, which are used to populate Γ (see also figure 5a).

Figure 3 shows the feature clouds of the towers, excluding the vector pencils associated with the relative, viewpoint-dependent information.

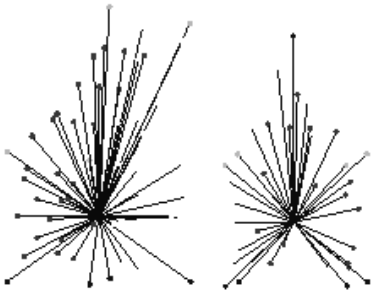

Figure 3: feature cloud representation of the towers

Figure 4 illustrates the effect of appearances of features. A single feature (for instance, the corner of the larger cube) appears differently from alternate viewpoints, and each is assigned a separate appearance type. Large cones in figure 4 arranged radially from features represent the range for a given series of detectors. These take the form of lobes. The small cones indicate individual appearances of detected features (as projected onto a notional image plane) from specific viewpoints.

Note that it is the properties of the detector that determine the angular and positional range of the appearance type. The scope of detectors (and hence appearances) may overlap, may be ambiguous - in that distinct features may give rise to the same appearance. Features may not be detected at all from some viewpoints, either because the detectors are not sensitive to it, or by virtue of self-occlusion.

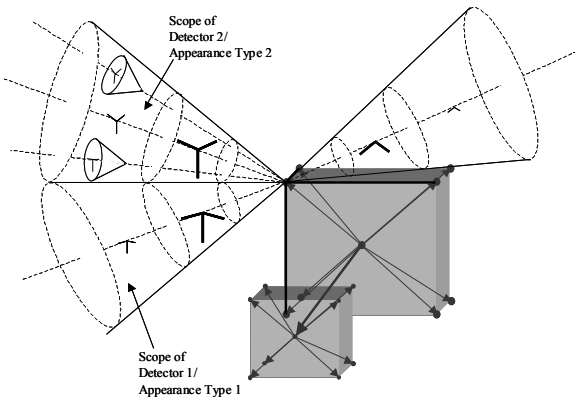

Figure 4: Appearances of features from different viewpoints.

Feature clouds unify and extend *point clouds* [Linsen, 2001] and *aspect graphs* [Koenderink and van Doorn, 1979; Shiffenbauer, 2001] in several ways. Firstly, unlike the point cloud, a feature cloud directly encodes sensor extracted features rather than simple surface points.

Secondly, the feature cloud encodes structural part-whole relationships between the component object parts of the inferred host bodies. Thirdly, unlike point clouds and classical (infinite resolution) aspect graphs, feature clouds directly encode viewpoint and sensor resolution dependent information. Finally, the feature cloud encodes information from which the relative distinctiveness of its feature types can be computed.

4 The Formalism

We assume general predicate logic with equality and classical set-theory, and extend this to allow reified terms, and (possibly empty) lists i.e. *[x1,...,xn]* to appear in formulae. Vector variables: $\underline{v0}$, $\underline{v1}$,...,$\underline{vn}$, are specifically singled out, and these are denoted in the text by the use of underscores. Greek letters (both lower and upper case) are reserved for meta-logical predicate variables, individual *wffs*, and sets of sentences. The context in which they appear will make this distinction clear. Where numbers, standard arithmetical and other mathematical operators appear in formulae, we will assume that have their usual meanings. We will further assume that in the implemented theory, the mathematical evaluation of functions runs concurrently with term-unification.

4.1 Objects, features and appearances

The primitive formal ontology ranges over bodies (which we call *objects*), surfaces, features, appearances, points and vectors, each of which are denoted by disjoint sets. Objects include 3D volumetric solids (e.g. solid polyhedra), features are localizable surface discontinuities (e.g. from edges or vertices of polyhedra), while appearances cover the 2D projected images of these in relation to an assumed viewpoint.

We use a simple generic data structure represented by the schema: $\Phi(x0,\underline{v0},Type0,[x1,...,xn])$, that respectively maps an individual *x0*, to its position vector $\underline{v0}$, sortal *Type0*, and list of component parts *[x1,…,xn],* and where $\Phi \in \{Object,Feature,Appearance\}$. Within the formalism the schema is used to encode a hierarchical structure with objects mapping to surface features, and with these mapping to a set of their viewpoint dependent appearances. At the descriptive level of features the structure encodes a *viewpoint independent* description of objects, and extending to their viewpoint-*dependent* appearances is where interpreted logical formulae are grounded in sensor data.

Axiom (A1) encodes the hierarchical decomposition between objects and their features, and between features and their appearances. Given an object, this is decomposed into object parts, or features; features are further decomposed into feature parts (compound features) or appearances, and appearances into sub-appearances or the empty list as a terminal node:

$$(A1)\ \Phi(x0,\underline{v0},Type0,[x1,...,xn]) \rightarrow \Psi 1(x1,\underline{v1},Type1,[...])\ \&...\&\ \Psi n(xn,\underline{vn},Typen,[...]),$$

where:
if $\Phi = Object$, *then*
$\forall i\ i=1$ *to* n: $\Psi i = Object$ *or,*
$\forall i\ i=1$ *to* n: $\Psi i = Feature$, *and*
if $\Phi = Feature$, *then*
$\forall i\ i=1$ *to* n: $\Psi i = Feature$, *or*
$\forall i\ i=1$ *to* n: $\Psi i = Appearance$,
else, $\forall i\ i=1$ *to* n: $\Phi = \Psi i = Appearance$

We will allow the overloading of the predicate symbols *Object/4*, *Feature/4* and *Appearance/4*, and define their monadic counterparts; to this end we add the following definition schema, which serves to provide a compact description of this form:

$$(D1)\ \Phi(x0) \equiv def.\ \exists \underline{v0},Type0,[x1,...,xn]\ \Phi(x0,\underline{v0},Type0,[x1,...,xn])$$
where:
$\Phi \in \{Object,Feature,Appearance\}$

The hierarchical structure is mirrored in spatial relationships between *vectors* where: $vector(\underline{v1},\underline{v2})$ maps vector $\underline{v1}$ to vector $\underline{v2}$; *vector/2* is axiomatised to be irreflexive and transitive (and by implication, is asymmetrical):

$$(A2)\ [\Phi(x0,\underline{v0},Type0,[x1,...,xn])\ \&\ \Psi 1(x1,\underline{v1},Type1,[...])\ \&\ldots\&\ \Psi n(xn,\underline{vn},Typen,[...])] \rightarrow vector(\underline{v1},\underline{v0})\ \&\ldots\&\ vector(\underline{vn},\underline{v0})$$

Vectors locate the *centroid* (notional position) of any object, feature or appearance ($\underline{v1}$ … $\underline{vn}$) with respect to the centroid of its supervenient feature or object ($\underline{v0}$), or of a viewpoint (actual or notional) in space. Using this scheme the position of any sub-part of an object may be determined relative to any other subpart by straightforward vector summation. Every logical operation between objects and their parts implies a corresponding vector operation. This is key to hypothetico-deductive framework outlined previously.

Vectors are cached out as tuples. These tuples encode the position of a point in 3-space and an assumed local coordinate system. The representation of vectors need not unduly concern us here as we are describing the *logical properties* of the theory. The important point is that the vectors allow us to reconstruct the position, scale and pose of a hypothesised 3D surface feature or object from an arbitrary (matched) set of sensor-detected features.

Where, individuals appear as terminal nodes in the tree structure, we define these as *atoms*; normally these take the form of appearances:

$$(D2)\ Atom(x) \equiv def.\ \Phi(x,\underline{v},type0,[]),$$
where:
$\Phi \in \{Object,Feature,Appearance\}$
i.e. x is an atom if it has no sub-parts

We also define objects that form no component-part of any other object, which we call *maximal* objects. The underlying intuition is to interpret these as commonsense categories of individuated macroscopic objects, such as tables and chairs, while ruling out gerrymandered mixtures

such as a table top and chair-leg as a similarly conceived unitary object:

(D3) MaxObject(x) ≡def.
¬∃y,v,Type, [...,x,...][Object(y,v,Type,[...,x,...])]
i.e. x is a maximal object if it forms no sub-part of another object.

We recognise that the commonsense notion of maximal object can change with context and scale (granularity), but will assume here that the choice becomes fixed by virtue of the definitions placed in Σo.

The mereological connection is intentional. It also explains our adoption of the mereological part-whole relation *P(x,y)* used to definine overlap *O(x,y,)*, and other classical mereological relations, for example: *PP(x,y)* ("proper-part") and *DR(x,y)* ("is discrete from") – see, for example [Randell *et al*, 1992]. We will assume this extension and embed these concepts into our existing framework, thus:

(A3) Φ(x0,v1,Type0,[x1,...,xn]) →
[∀xi ∈ [x1,…,xn] Ψ(xi,vi,Typei,[…]) → PP(xi,x0)]

where:
if *Φ=Object, then*
Ψ∈{Object,Feature}
else: Φ= Ψ=Feature
i.e. if x0 is an object or feature, then every sub-part of x0 is also a proper part.

Each object, feature and appearance is assigned a *sortal* predicate denoted by "*Type0*" in the general schema: *Φ(x0,v0,Type0,[x1,...,xn]).* These predicates are distinguished from other monadic predicates used in the theory, e.g. *Opaque(x).* Sortals are universals that provide a principle for distinguishing, counting and reidentifying particulars [Lacey, 1976]. In addition to these monadic predicates, we also use a set of relations, e.g. *TotallyOccludes(x,y,v)* all of which appear in either ∑ or Δ depending on whether or not a hypothesised set of individual objects are generated.

4.2 Surfaces

Surfaces are not explicitly represented in the feature cloud, but are factored out and treated separately. We model physical surfaces as a Deluanay triangulation of an arbitrary topology manifold surface in 3D space. Solids consequently take the form of polyhedra whose faces are ultimately decomposed into a finite set of triangles, which we call *facets*. This allows the entire surface of a solid and any part of that surface to be delineated by a set of edge connected planar facets whose vertices map to features. The surface of a solid and a facet is represented thus: *Surface(surface(x),BoundingSurface,[f1,…,fn]),* where given the function *map(<f1,…,fn>) = <v1,…,vn>, n≥4, v1* to *vn* are a set of non-coplanar points; and *Surface(x,Facet,[f1,f2,f3]), where map(<f1,f2,f3>)= <v1,v2,v3>,* and *v1* to *v3* denote three non-collinear points. In our modeling domain, all our surfaces are assumed to be opaque. Straddling in between our primitive facets and entire surfaces of individual modelled solids, are faces of reconstructed solids which are defined as *maximal* convex planar polygons: *Surface(x,Face,[f1,…,fn])* where given the function *map(<f1,…,fn>) = <v1,…,vn>, n≥4, v1* to *vn* are a set of coplanar points.

The difficulty of automatically reconstructing surfaces from point clouds is well known. Reconstructed surfaces should not only be topologically equivalent to the sampled surface, they should also provide a close geometrical approximation. To get a close approximation to the physical surface of an arbitrary shaped solid when using a point cloud, the sampling often needs to be dense, with the result that the number of facets generated are large.

5 Constraints

Σc contains our constraints. These include geometric and commonsense information about the everyday world: for example, that objects such as chairs and tables do not volumetrically overlap, and that any opaque body or surface seen from a viewpoint, necessarily occludes a similar surface or body lying behind it. Constraints are exploited by: (i) pruning arbitrary juxtapositions of features and hypothesised bodies, and (ii) providing a principled means for constructing new generic objects when sensor data cannot be assimilated into Σ. Also included here are two example axioms that show how the physical property of opacity is handled and embedded into our logical framework:

(A4) ∀x1,x2 [[MaxObject(x1) & MaxObject(x2)
& ¬(x1=x2)] → DR(x1,x2)]
i.e. distinct maximal objects are discrete. In this model, physical objects occupy individual spaces, and as a consequence do not intersect each other.

(D4) Opaque(x1)≡def.
∀x1,x2,v1,v2,Type1,Type2,..,v0[[Φ1(x1,v1,Type1,[…])&
Φ2(x1,v2,Type2[…]) & Behind(x2,x1,v0)] →
¬Detected(x2,v0)]

where:
Φ=Object, Φ2∈{Object,Feature}
i.e. x is opaque if from each viewpoint v, whatever is behind x is not visible.

(A5) ∀x0 [Opaque(x0) ↔ Opaque(surface(x0))]
i.e. an object is opaque if its surface is opaque

(A6) ∀x0 [Opaque(surface(x0)) →
∀y[P(y,surface(x0)) → Opaque(y)]]
i.e. if the surface of an object is opaque, then every part of the object's surface is opaque

While not developed here, we also assume other temporal and geometric constraints, e.g. that at any point in time, and for every robot (one robot in this particular case) exactly one robot viewpoint exists. This has the direct consequence (from the underlying geometry assumed) that only one appearance of a detected feature is presented in

the image. In the case of stereo vision, however, the viewpoint is interpreted as the fusion of the paired images into a single, notional viewpoint.

6 Spatial Relations

Related to these geometrical and psychophysical constraints are a set of spatial occlusion relations (*c.f.* Randell *et al*. [2001], Randell and Witkowski [2002]) of which total occlusion and partial occlusion are given here:

(D5) TotallyOccludes(x,y,v0) ≡def.
Opaque(x) & Object(x,…)& Object(y,…)&
∀z[[Feature(z,…)& Object(y,…,[…,z,…]) &
Expected(z,v0) & Behind(z,x,v0)] →
¬Detected(z,v0)]

(D6) PartiallyOccludes(x,y,v) ≡def.
Occludes(x,y,v) & ¬TotallyOccludes(x,y,v)&
¬Occludes(y,x,v)

The predicate *Behind/3* is an object-level *primitive* relation whose truth-value is computed using information directly encoded in our hypothesised 3D model.

7 Measuring Uncertainty and Evidential Support

We will also require *meta-level* definitions governing what it is for a feature to be expected (to be visible) and detected:

(D7) Expected(x,v0)≡def. Feature(x,…) &
¬∃yTotallyOccludes(y,x,v0)],
where: {Feature(x,…),¬∃yTotallyOccludes(y,x,v0)}⊆ Δh
i.e. feature x is expected from viewpoint v0, if no object occludes it

(D8) Detected(x,v0) ≡def. Feature(x,…,[…,y,…]) &
Appearance(y,v0,Type,[…]),
where: Feature(x,…,[…,y,…]) ∈Δh, and
Appearance(y,v0,Type,[…]) ∈Γ
i.e. feature x is detected from viewpoint v0 if an appearance of x is registered in Γ.

D7 indicates that feature is expected if it has been hypothesised into Δh and is not occluded by anything from the actual current viewpoint. D8 indicates that such a feature has been matched to a specific appearance in the incoming data, Γ. These predicates are used to define four exhaustive cases, which are directly applied to the task of confirming, verifying and refuting hypotheses:

(i) *Expected(x,v)&Detected(x,v):* strong positive support, no new explanation required;
(ii) *Expected(x,v)&¬Detected(x,v)*: weak negative support, new explanation required;
(iii) *¬Expected(x,v)&Detected(x,v)*: novel event, new explanation required, and
(iv) *¬Expected(x,v)&¬Detected(x,v)*: weak positive support, no new explanation required.

In the particular case where (iii) occurs, detected features can give rise to several alternative *causal* explanations, namely: (a) the result of sensor noise, (b) image processing errors, or (c) arising from physical features of imaged bodies, but whose object types do not appear in *Σo*. The distinction between these cases is handled as follows. Only when an interpreted feature is causally explained by a physical feature that has a specific location and pose in space, and whose constancy is verified by changing the relative position of the viewpoint and matching this against the model, do we treat this as a physical feature needing to be assimilated into *Σo*. This level of constancy typically fails in case of sensor noise, and where image processing errors arise.

7.1 Mapping hypothesised objects to sensor data

As our theory encodes both mathematical and logical information we use a two-pronged attack to find an optimal (or equal best) explanation for our sensor data by: (i) evaluating alternative (logical) models that satisfy our axioms, and (ii) solving a geometrical correspondence problem between features identified in a 2D image and those features represented in a 3D model.

In the former case, the correspondence is determined symbolically using the inferential map between objects, features, and their appearances directly encoded as feature clouds; while in the latter case, the correspondence reduces to a well known and much studied topic in photogrammetry and computer vision [e.g. Haralick *et al*., 1994; DeMenthon and Davis, 1995; Horaud *et al*., 1997; Hu and Wu, 2002; David *et al*., 2004]. While no unique analytical solution exists, for n≥3 registration points, it is known that at most four solutions exist. We argue that this low upper-bound result (when combined with the hypothetico-deductive framework used here) does not make the computational task of generating (and filtering out alternative hypotheses) significantly less tractable.

Of the many algorithms that have been developed to solve this geometrical correspondence problem, POSIT [Dementhon and Davis, 1995] and SoftPOSIT [David *et al*., 2004] are of particular note. For example, POSIT assumes a match of ≥4 non coplanar registration points and that their relative geometry is known. This requirement is relaxed in SoftPOST. In terms of robustness to noise (i.e. clutter) and speed, implementations of POSIT claim real-time performance [Dementhon and Davis, 1995]. See also [Horaud *et al*., 1997; Hu and Wu, 2002] for real-time vision, simulated and robotics applications. These results assume a geometry predicated on points; but in our case additional constraints are applied, for example those supplied by viewpoint and sensor (resolution) dependent information. We now define the various measures that permit this mapping to be determined.

7.2 Distinctiveness, rank order and explanatory measures

Following Shanahan [2002, 2005] and Poole [1992, 1993, 1998] we assign numerical measures (and probabilities) to logical formulae. This is handled in two complementary ways: (i) where the probability space is identified with the number of possible *term substitutions* between a target expression and that encoded in our generic object-language descriptions; and (ii) classically, as *possible worlds*, where each world is a *model* or assignment of values for variables for our axioms and interpreted sensor data.

We use *three* information measures: (i) the distinctiveness value *dv(F)* of a feature type, (ii) the rank-ordering *rank(O)* of an object type with respect to incoming sensor data, and (iii) the explanatory value $ev(x,\underline{v})$, of an individual hypothesised object given available sensor evidence.

7.2.1 Distinctiveness. First, we define the distinctiveness of a feature of type *F*. By convention the value of the probability of α, defined as $P(\alpha)$, lies between 0 and 1.

Let $S(\Phi)=\{fi|Feature(fi,\underline{v},\Phi,[...])\in\Sigma o\}$, *and* $S=\{fi|Feature(fi,\ldots)\in\Sigma o\}$, Let $|S|$ denote the cardinality of set *S*. Then:

$$dv(F) = 1 - \frac{|\underline{S(F)}|}{|S|} \qquad P(dv(F))=1- dv(F)$$

The *dv(F)* value is an *a priori* measure, and is based entirely on the informational content of features encoded in Σo. We do not index appearances as these presuppose a viewpoint-dependent description of features[1]; though other arguments for indexing appearances can be provided.[2]

7.2.2 Rank Order. Rank order is used to determine which objects defined in Σo are likely to be present in the scene based on the evidence presented by the incoming sensor data stream. Where before the *dv(F)* value was calculated solely on *a priori* information; now we measure the degree to which our object descriptions match incoming sensor data.

Definition: $\{t1/v1,...,tn/vn\}$ is a substitution where every *vi* is a variable, and every *ti* is a term different from *vi*, and where no two elements of the set have the same variable after the stroke symbol. If $P(v1,...,vn)$ is a *wff*, and $\theta=\{t1/v1,...,tn/vn\}$, then $P(v1,...,vn)\theta=P(t1,...,tn)$ [Chang and Lee, 1973.]

Definition: Given a substitution θ and a *Type*, θ is a *legal substitution* with respect to *Type* if: (i) There is no $t/v1\in\theta$, and $t/v2\in\theta$ s.t. $v1\neq v2$, and, (ii) it is not the case that $Feature(f,\ldots,[\ldots,ai,\ldots])\in\Sigma o$, $Feature(f,\ldots,[\ldots,aj,\ldots])\in\Sigma o$, $Appearance(ai,\ldots)\in\Sigma o$, $Appearance(aj,\ldots)\in\Sigma o$, $Appearance(a1,\ldots)\in\Gamma$ and $Appearance(a2,\ldots)\in\Gamma$, $a1\neq a2$, $a1/ai\in\theta$, $a2/aj\in\theta$, and $ai\neq aj$.

Definition: A valuation *v*, of *Type0* with respect to a legal substitution θ, $v(Type,\theta)$ is defined as follows:

$$v(Type0,\theta) = \frac{\Sigma\ |\{fi\}|\times dv(\Phi)\colon [Feature(fi,v1,\Phi,\ldots)\&\ \Omega]}{\Sigma\ |\{fj\}|\times dv(\Psi)\colon [Feature(fj,\underline{v2},\Psi,\ldots)\&\ \Xi]}$$

where: Σ, Γ *entails:*

$$\Omega \equiv [[[Object(x0,\underline{v0},Type0,[\ldots,fi,\ldots])\ \&\ Feature(fi,\underline{v1},\Phi,\ldots)] \rightarrow Appearance(ak,\ldots)]\ \&\ [Appearance(a1,\ldots) = Appearance(ak,\ldots)\theta\,]]$$

$$\Xi \equiv [Object(x0,\underline{v0},Type0,[\ldots,fj,\ldots]) \rightarrow Feature(fj,\underline{v2},\Psi,\ldots)],$$

where: $Appearance(a1,\ldots)\in\Gamma$, *and where:* $\{Object(x0,\underline{v0},Type0,[\ldots,fi,\ldots]),\ Object(x0,\underline{v0},Type0,[\ldots,fj,\ldots]),\ Feature(fi,\underline{v1},\Phi,\ldots),\ Feature(fj,\underline{v2},\Psi,\ldots)],Appearance(ak,\ldots)\}\subseteq\Sigma o$

Definition: The rank order of *Type*; *rank(x,Type)* is defined as the maximal valuation $v(Type,\theta)$, s.t. there is no $\theta 1$, s.t $v(y,Type,\theta 1)> v(x,Type,\theta)$.

A careful examination of possible substitutions between formulae used in the rank order shows that each ground expression in Γ that is mapped to an expression in Σo is restricted so that distinct appearances $\{a1,a2,...,an\}$ are mapped to distinct features $\{f1,f2,...,fn\}$ in Δh.

No explicit viewpoint variable appears in the function *rank(Type)*, even though a notional viewpoint of the robot is assumed and encoded in the appearance descriptions. This is because the pose estimate of an object is not tested until the explanatory value is calculated.

The rank order assigns a partial ordering of object types in the light of interpreted incoming sensor data and is used to determine the order in which the explanatory value of hypothesised objects is computed. It serves as a heuristic, to generate a "best-guess" causal explanation as to what the evidence supports. For this reason, the cognitive and computational role of the rank order can be likened to a model of attention as visual efficiency [Khadhouri and Demiris, 2005].

With rank ordering of all object types computed, the task remains to map clusters of detected features to individual hypothesised, host objects. Here the dv(F) values of the features of an assumed individual object as its bearer are consulted, and those with the greatest distinctiveness value preferentially selected. Sets of four of such features, whose appearances map to detected features in the image plane are selected and are cyclically passed to *POSIT* to compute the positional vector of the assumed host object that has these matched features. It at this stage in the computational process, an hypothesised object and its computed position vector relative to the assumed viewpoint, is used to calculate that object's explanatory value.

---

[1] This serves to make the measure indifferent to the imaging system used, where the focal length and field of view of view is inextricably tied to the detection or not of objects and their corresponding features.

[2] For example, establishing the salience of features for the purposes of tracking objects.

7.2.3 Explanatory Value. The role of the explanatory value closely follows that described in [Shanahan 2002; Shanahan and Randell, 2004]. In our case we measure the degree to which an hypothesised (maximal) object *x* seen from a viewpoint is explained by and predicts currently available sensor data: The explanatory value *ev(x,v)* is defined as follows: where the range of *ev(x,v0)* lies between +1.0 ('corroboration') and –1.0 ('refutation'); respectively between 1 and 0 for *P(ev(x,v0)):*

$ev(x0,v0)=(A+B)-(C+D)/(A+B+C+D)$

$A= [\Sigma|\{fi\}| \times dv(\Phi)$: *[Feature(fi,…, Φ,…) & Expected(fi,v0) & Detected(fi,v0)]*

$B= [\Sigma|\{fi\}| \times dv(\Phi)$: *[Feature(fi,…, Φ,…) & ¬Expected(fi,v0) & ¬Detected(fi,v0)]*

$C= [\Sigma|\{fi\}| \times dv(\Phi)$: *[Feature(fi,…, Φ,…) & Expected(fi,v0) & ¬Detected(fi,v0)]*

$D= [\Sigma|\{fi\}| \times dv(\Phi)$: *[Feature(fi,…, Φ,…) & ¬Expected(fi,v0) & Detected(fi,v0)],*

where, Σ,Δ entails:
[[MaxObject(x0,…) & Feature(fi,v1,Φ,…)] → Appearance(…,v0,…)],

and where a set of ground substitutions {θi,θj,θk...} exist such that {MaxObject(x0,…)θi, Feature(fi,…,Φ,…)θj, Appearance(…,v0,…)θk}⊆Δh

$P(ev(x,v0))= (ev(x,v0)+1) / 2$

We now show how these mathematical and logical constructs are to be used with an illustrative worked example.

8 A simple worked example

This section develops a minimal motivating example to illustrate how the formalism developed in the previous section may be used to process a set of appearance terms in Γ. We assume a very simple world in which only solid, opaque towers. Tower T1 is a solid comprising a cuboid with a wedge-shaped top, tower T2 is a cuboid with a pyramidal shaped top. Figure 5a shows the robot's view of the scene shown in figure 2. We also assume we have a set of implemented apex and corner detectors (✳, ×), and detectors that identify surface patches (+). Each symbol (+,×, ✳) indicates the vector position on the image plane of the corresponding detector centre. The object outlines are shown to assist the reader, *they do not form part of Γ.*

We now begin the process of matching features to our object types. First we use the distinctiveness value to pre-bias the search for features of a given type. This is used to order individual hypothesised object types as possible candidates that explain our incoming interpreted sensor data. Suppose, for the purposes of this example, that T2 has the highest rank order value, so this model is selected first. The search proceeds by selecting four non-coplanar points (required by *POSIT*) in our hypothesised 3D model and pairing these to four features in the image. The type of the feature appearance in the Σo model must match that of the appearance type in Γ. Equally the appearance vectors of the four selected features must converge to a single viewpoint, or the set will be inconsistent.

With the 2D-3D point-point registrations made, the reconstructed 3D pose of the hypothesised object is determined, and the hypothesised object instantiated into Δh. This is used (via a set of linear transforms produced by POSIT) to reconstruct, and hence predict, the position of all the features as they would appear in the 2D image, by simple projection. In each case, the explanatory value for the complete projected model (in Δh) compared to the sensor data (Γ) is computed.

Figure 5b illustrates the effects of a mismatch of image to model points. Figure 5c illustrates the effect of a correct match. In figure 5b, three of the model points are matched "correctly", the fourth is mismatched to an image point of

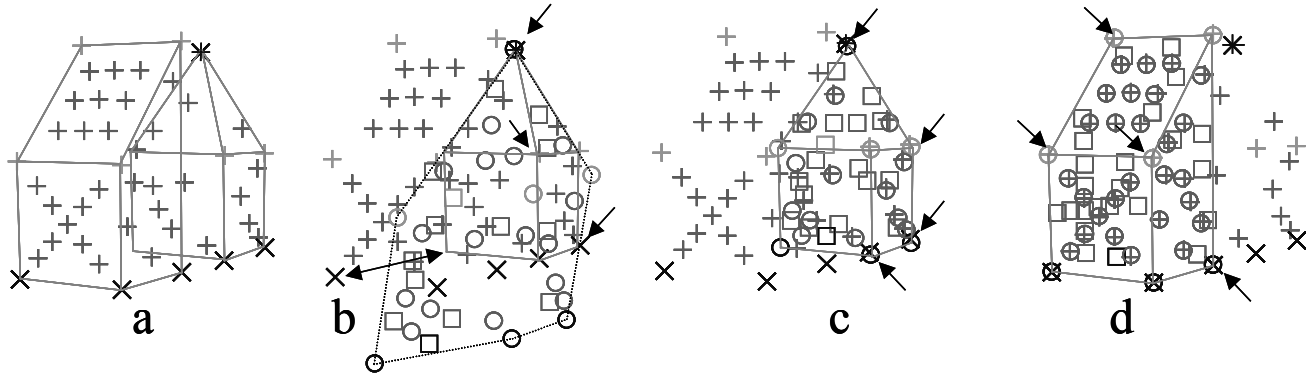

Figure 5: The robot's view of the world, showing the Γ field (a) and selected steps in the analysis (b-d)

Table 1: Feature Types for each calculation, and resulting explanatory value

	T2 distorted				T2 correct projection				T1 correct projection				T2 re-considered			
	E∧D	E∧¬D	¬E∧¬D	¬E∧D	E∧D	E∧¬D	¬E∧¬D	¬E∧D	E∧D	E∧¬D	¬E∧¬D	¬E∧D	E∧D	E∧¬D	¬E∧¬D	¬E∧D
Apex	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
Corner	0	6	1	2	4	2	2	0	7	0	0	0	4	0	2	0
Other	1	16	12	16	9	4	16	2	27	0	23	0	9	0	16	2
ev()	-0.51				0.58				1.0				0.94			

the same type, but actually from an adjacent object, as indicated by the arrows. The pose of the resultant model is clearly distorted and the predicted points may be compared to the sensor data. The dotted line (figure 5b) indicates the area of the projected model within which evidential support will be gathered.

In fig.5 circles represent feature points that are expected to be visible, and squares places where projected feature points are expected not to be visible, by virtue of self-occlusion. This sets up the conditions to evaluate evidential support for the hypothesised object as defined by combinations of definitions D7 and D8 described in section 7. Table 1 shows the image features and model points meeting each of the four support criteria. These evaluate to an explanatory value of –0.51. At this level of dis-confirmatory support the hypothesis is abandoned and a new match made (fig. 5c). The new predictions match well enough (+0.58) to accept the hypothesis pending further investigation.

In the next step the remaining distinctive features in Γ are used to hypothesise another model (figure 5d). With the match shown the model of Tower A gives rise to an explanation for the remaining sensor data features in Γ (ev = +1.0) and so $\Sigma \wedge \Delta \models \Gamma$. As a side effect, and as Tower A partially occludes Tower B, this now provides further explanation as to the incomplete match for Tower B. By removing the previously unexplained missing elements in the Expected & ¬Detected category the explanatory value of Tower B can be upgraded to +0.94.

In the background, the set of global constraints prune out potential models, e.g. ruling out models where maximal objects overlap, but also several object poses by properties predicted as arising from occlusion. In the latter case we can also see that given a Jointly Exhaustive and Pairwise Disjoint (JEPD) set of spatial occlusion relations, each spatial relation predicts different explanatory value ranges for the occluding and occluded objects. This then provides the abductive basis to infer our top-level relational descriptions of our objects in space.

9 Conclusions and Future Work

In terms of model matching, we now see the flexibility and power of the representation. The logical description provides the syntactic and semantic basis for abductively inferring host bodies using sparse information. Running concurrently with this, we also have the 3D geometrical manipulation of feature clouds, 3D model matching using 2D views, and a set of constraints arising from viewpoint-dependent properties of our implemented feature detectors. Taken as a whole, the two-pronged attack generates a set of target explanatory hypotheses (Popper's *bold conjectures*) supporting a high degree of potential falsifiability.

It should be remembered that a logical formulation of perception does not of itself solve the underlying problems encountered in practical visual processing. Rather it provides a formal framework with which to abstract critical factors in the perceptual process, as well as acting as a specification for logic based programming solutions.

Our approach models visual perception as a combination of bottom-up, sense driven, and top-down, expectation driven processes, and as such accords well with a substantial body of empirical and experimental data from visual psychophysics (e.g. [Rock, 1981]); and from neurophysiological evidence of a two way flow of information (e.g. [Lee and Mumford, 2003]). The approach is also closely related to that of active perception [Aloimonos *et al*., 1987; Ballard, 1991] and has potential applications to sensor fusion using multiple sensor modalities [Hall and Llinas, 2001].

The work presented here has a strong connection with the formal modelling of attention and granularity. In the former case attention brings to the fore resource allocation issues, and previously discussed salience and distinctiveness measures; while granularity directly relates to sensor dependent information, such as scale information and the resolution of detected features as a function of the viewing distance, focal length and pixel-array of the camera sensor used.

We also define a panoramic description of the domain as seen from the assumed robot's viewpoint (Δp) but which maintains previously assimilated information, but not currently in the robot's visual field. This structure maintains an abstraction of the juxtaposition of objects in the robot's immediate environment relative to its viewpoint. This has the effect of allowing the robot to reason about objects and their relationships with itself and each other, although they are outside the current field of view. Further, the panorama may be transformed according to anticipated or actual movements of the robot, to populate Δh with immediately verifiable hypotheses.

The weighting supplied distinctiveness measure does not take into account the number of actual instances of objects that occur in Δ, and thus one modification to this measure would to define dv(F) on Δ, and default to the Σ values only when no information as to the number of detected objects arises, for example, during an initialisation routine.

Acknowledgements

This work was supported by UK EPSRC under grant number EP/C530683/1, "Abductive Robot Perception: Modelling Granularity and Attention in Euclidean Representational Space". With thanks to Murray Shanahan for his detailed comments.

References

Aloimonos, J., Weiss, I. and Bandyopadhyay, A. (1987) "Active Vision", *1st Int. Conf. on Computer Vision*, pp. 35-54.Ballard, 1991

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

Chang, C-L. and Lee, R. (1973) *Symbolic Logic and Mechanical Theorem Proving*, Academic Press, Inc.

Chella, A., Frixione, M. and Gaglio, S. (2000) "Understanding Dynamic Scenes" *Artificial Intelligence*, **123**(1–2), pp. 89-132.

Cyberbotics (2006) "Webots 5: fast prototyping and simulation for mobile robotics", www.cyberbotics.com, accessed 28/02/06.

David, P., DeMenthon, D., Duraiswami, R. and Samet, H. (2004) "SoftPOSIT: Simultaneous Pose and Correspondence Determination", *International Journal of Computer Vision*, **59**(3), pp. 259-284.

DeMenthon, D. and Davis, L.S. (1995) "Model-Based Object Pose in 25 Lines of Code", *International Journal of Computer Vision*, **15**, pp. 123-141.

Gibson, J.J. (1979) *The Ecological Approach to Visual Perception*, Houghton Mifflin.

Hall, D.L. and Llinas, J (2001) *Handbook of Sensor Fusion*, Boca Raton: CRC Press.

Haralick, R., Lee, C., Ottenberg, K. and Nolle, M. (1994) "Review and Analysis of Solutions of the Three Point Perspective Pose Estimation Problem", *International Journal of Computer Vision*, **13**, pp. 331-356.

Horaud, R., Dornaika, F., Lamiroy, B. and Christy, S. (1997) "Object Pose: The Link Between Weak Perspective, Paraperspective, and Full Perspective", *International Journal of Computer Vision*, **22**(2), pp. 173-189.

Hu, Z.Y. and Wu, F.C. (2002) "A Short Note on the Number of Solutions of the Noncoplanar P4P Problem", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**(4), pp. 550-555.

Itti, L. and Koch, C. (2001) "Computational Modelling of Visual Attention", *Nature Reviews, Neuroscience*, **2**, March 2001, 11 pp.

Khadhouri, B. and Demiris, Y. (2005) "Compound Effects of Top-down and Bottom-up Influences on Visual Attention During Action Recognition", *Proc. IJCAI-05*, pp. 1458-1463.

Koenderink, J.J. and van Doorn, A.J. (1979) "The Internal Representation with Respect to Vision", *Biological Cybernetics*, **32**, pp. 211-216.

Lacey, A.R. (1976) *A Dictionary of Philosophy*; London: Routledge & Kegan Paul.

Lee, T.S. and Mumford, D. (2003) "Hierarchical Bayesian Inference in the Visual Cortex", *J. Opt. Soc. America* **A-20**(7), pp. 1434-1448.

Lowe, D. G. (2004) "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, **60**(2), pp. 91-110.

Linsen, L. (2001) "Point Cloud Representation", Tech. Report, Faculty of Computer Science, U. of Karlsruhe.

Marr, D. (1982) *Vision*, New York: W.H. Freeman

Pirri, F. (2005) "The Usual Objects: A First Draft on Decomposing and Reassembling Familiar Objects Images", *Proc XXVII Ann. Conf. of the Cognitive Science Society*, pp. 1773-1778.

Poole, D. (1992) "Logic Programming, Abduction and Probability", Proc. Int. Conf. on Fifth Generation Computer Systems (FGCS'92), pp. 530-538.

Poole, D. (1993) "Probabilistic Horn Abduction and Bayesian Networks", *Artificial Intelligence*, **64**(1), pp. 81-129.

Poole, D. (1998) "Learning, Bayesian Probability, Graphical Models, and Abduction", in: Flach, P. and Kakas, A. (Eds.), *Abduction and Induction: Essays on their Relation and Integration*, Kluwer, 1998

Randell, D.A. (2005) "On Logic, Logicism and Logic Based AI", *AISBQ*, **220**, Spring 2005, page 2.

Randell, D.A.; Cui, Z; and Cohn, A. G. (1992): "A Spatial Logic based on Regions and 'Connection'", in *Proc KR92*, 1992.

Randell, D.A., Witkowski, M. and Shanahan, M. (2001) "From Images to Bodies: Modeling and Exploiting Spatial Occlusion and Motion Parallax", *Proc. IJCAI-01*, pp. 57-63.

Randell, D.A. and Witkowski, M. (2002) "Building Large Composition Tables via Axiomatic Theories", *KR-2002*, pp. 26-35.

Rock, I. (1983) *The Logic of Perception*, MIT Press

Schmid, C., and Mohr, R. (1997) "Local Grayvalue Invariants for Image Retrieval", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **19**(5), pp. 530-534.

Shanahan, M.P. (1996) "Robotics and the Common Sense Informatic Situation", *ECAI-96*, pp. 684-688.

Shanahan, M.P. (1997) "Noise, Non-Determinism and Spatial Uncertainty", *AAAI-97*, pp. 153-158.

Shanahan, M.P. (2002) "A Logical Account of Perception Incorporating Feedback and Expectation", *KR-2002*, pp. 3-13.

Shanahan, M.P. (2004) "An Attempt to Formalise a Non-Trivial Benchmark Problem in Common Sense Reasoning", Artificial Intelligence, vol. **153**, pp. 141-165.

Shanahan, M.P. (2005) "Perception as Abduction: Turning Sensor Data into Meaningful Representation", *Cognitive Science*, **29**, pp. 103-134.

Shanahan, M.P. and Randell, D.A. (2004) "A Logic-Based Formulation of Active Visual Perception", *KR-2004*, pp. 64-72.

Shiffenbauer, R.D. (2001) "A Survey of Aspect Graphs", Technical Report, Polytechnic University of Brooklyn, Long Island and Westchester, Dept. Computer and Information Science, TR-CIS-2001-01