

Selective attention in dynamic vision

Richard Howarth*

Dept. of Computer Science,
Queen Mary and Westfield College,
Mile End Road, London E1 4NS, UK

Hilary Buxton

Cognitive & Computing Sciences
University of Sussex,
Falmer, Brighton BN1 9RH, UK

Abstract

Selective attention can enable a vision system to perform effectively. The high-level vision component developed here uses Bayesian networks combined with a deictic representation to (1) create a dynamic structure to reflect the spatial organisation of the data and (2) measure task relatedness. Together these give attentional focus making the reasoning relevant to the task.

1 Introduction

One important use of high-level vision is to guide the reasoning of the whole vision system, acting as an attentional mechanism that exploits knowledge of the specific problem being solved. Here we develop a general framework for such an attentional mechanism and its application to understanding dynamic scenes.

Although the example used in this work is taken from the application domain of road traffic, this work is concerned with the general vision problem of understanding what we see. The case being considered here is a fixed official-observer¹ (camera) watching a dynamic outdoor scene for which a low-level vision system has been built that returns 3D pose positions of those dynamic objects visible to the camera. This processing is made feasible by having a known scene layout and expected object types. Further prior knowledge is used by the higher-level reasoning system which includes more extensive data about the spatial-layout and typical object behaviour [6].

In this approach, we also identify some key features to bias the attention of the running system so that it searches for these features, identifying any occurrences. Often when we interact with the world it is with some purpose in mind. This purpose enables us to grade the available features and select the essential properties from the various sensory inputs. Without such selectivity we would be overwhelmed by the available data. In this paper we show how task bias can be used to make the reasoning performed relevant to the task at hand.

*This work has been funded by SERC under a CASE award with the GEC Marconi Research Centre.

¹The name official-observer denotes the viewpoint of the system, i.e. the camera's field-of-view.

2 Issues

The starting point for this work is a temporally ordered stream of 3D pose updates representing the movements of dynamic scene objects in a known scene. The two key problems encountered when processing this data are: (1) The *computational load* of determining what all the objects are doing. For instance, we may not want to consider all object interactions. (2) Even in the relatively compact representation which comprises the set of pose positions there is the problem of *managing* the temporally evolving mass of input data.

A fundamental step in the approach taken here separates the simple and complex operations that can be performed upon the input data, and uses the results from the simple operations to guide the application of appropriate complex functions. The goal here is to reduce the irrelevant application of the more computationally expensive complex operations by making the reasoning performed relevant to the task at hand. In the context of understanding what a scene object is doing, simple operations include those that return values for positions, velocity, nearest-object, while complex operations perform computations such as path-prediction and other-object-location. This decomposition of the computation is also combined in this approach with the simplifications provided by the deictic representation outlined in the next subsection and with the agent based formalism using Bayesian network updating to give a general framework for implementation.

2.1 Object representation

Related to the issue of selective attention by the official-observer is set of properties associated with the object being observed. Gibson [4] called these properties "affordances". Here we are trying to describe the behaviour of dynamic scene objects. Generally, these objects have an intrinsic front so that when we describe the actions of such objects, we use the object's own local coordinate system. In effect, we have translated our own frame of reference to that of the object. This is one property of the deictic representation² used to describe the activities of the scene objects. Other properties are used to

²For details concerning deictic representation see [2; 11].

simplify the “typical-object-model”³ that is used to select which data gathering “visual-operators”⁴ should be activated. There are two classes of visual-operator used here. Peripheral visual-operators are used to perform simple computations upon all pertinent data, and attentional visual-operators are used to perform specific local data-gathering computations. The typical-object-model also determines which values, if any, should be retained (latched) for the next clock tick. During each clock cycle, each “foveated object” (called an “agent”) uses its typical-object-model to select which attentional visual-operators to run on the next clock-tick. The results from these selected visual-operators are fed back to the appropriate agent providing task related features upon which to base its next selection.

The central premise behind this agent selective viewpoint is that an agent need not name and describe every object in the domain, but instead should register information only about objects that are relevant to the task at hand. At any one moment the agent’s internal representation should register only the features of a few key objects and ignore the rest.

The system is currently set up with just three agents.⁵ The egocentric data gathering of each agent is integrated by the official-observer. This is simplified by the fact that the official-observer originally selected which objects to attend and thus the execution of the typical-object-model is like the official-observer considering the possible next actions of an object it is attending.

2.2 Selection task

The typical-object-model run by each agent operates at the wrong computational level to act as an attentional controller for the official-observer. This guiding task should be based upon both (1) the peripheral visual-operator outputs, and (2) the results from each of the agents. These outputs correspond to the simple and complex operations mentioned above. We can summarise this by considering a *general case*:

Definition 1 *Given two operators OP1 and OP2 where (1) OP1 is much less complex and runs faster than OP2, (2) OP1 does not return an answer that is as useful as OP2, (3) OP1 is true in all situations where OP2 would provide a useful result (4) OP1 is false in all situations where the preconditions of OP2 would not be met or where we do not want to apply OP2. If these conditions are fulfilled, OP1 can act as a guide for the application of OP2.*

When considered as part of the system, the general case is complicated by (1) the data-association-problem⁶ at the observer-level, (2) the selection of which OP2s to

run is dependent upon factors other than the OP1 results alone, for example, later in this paper we will describe how observer-level results are used to influence the allocation of agents.

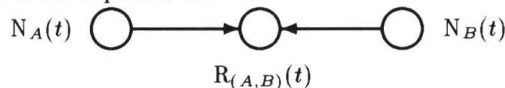
3 The basic observer-level

In this section we have chosen functions for OP1 and OP2 that reflect the example used in section 4. These functions are used for spatial reasoning about the visual data. One of the most elementary spatial primitives in spatial reasoning concerns the relative nearness of one object to another. In the example below we use a proximity measure as our *simple* operator using the assumption that objects generally interact with those that are nearby. The nearness measure is based on a distance metric scaled to give an exponentially higher value to nearer objects and discretised to the integer range $[0, \dots, 5]$ (this is also the range used by $N_i(t)$ in table 1). The more complex operation takes this result and determines the bearing of the nearest (other) object relative to the object’s frame-of-reference. The result is a deictic reference such as *behind-me*, *beside-me* or *infront-of-me*. (Note that conditions (1,2,3,4) are fulfilled – the complex operator only works when given the result saying which, if any, is the nearest object.)

We are using Bayesian networks [9] which combine an inference mechanism for uncertainty together with an efficient propagation of evidence and expectations. Recently, a few researchers have used Bayesian networks in dynamic domains [3; 8] where the focus is reasoning over time. This is not the only focus here, we also use another Bayesian network to assess the *task relatedness* of the actions selected so far, providing a *task-directed* attentional mechanism.

3.1 Selecting interesting features

To combine the proximity information that develops over time we have used a dynamic form of Bayesian network (DBN). That is, the network evolves structurally over time to capture the changing proximity relationships between scene objects. In the case outlined here, the graph formed describes which object is near another one. The node values hold data that contributes towards a measure reflecting how interesting this relationship is to the official-observer watching the scene. The basic graph primitive is depicted as:



This graph describes a mutual proximity relationship between objects *A* and *B*, capturing the primitive notion of object *A* being near *B* and *B* being near *A*. The relationship node *R* denotes this mutual proximity and holds a belief value that reflects how interesting this is. The states and purpose of the two node types are given in table 1. The inputs to DBN are taken from the peripheral visual-operator outputs (called “peripheral-aspects”). These outputs are given once every system-cycle, which means that the values held at time *t* are from the current outputs. At the end of each system-cycle, the storage is “moved” (by pointer reallocation)

³Denotes the concept of representing the “typical” behaviour exhibited by the class of object being observed. In effect a set of rules for *mapping* a given “situation and recent history” in the form of key features to effector actions.

⁴As used by Agre and Chapman [1].

⁵Pylyshyn and Storm [10] present psychophysical evidence that there is a numerical limit of four or five on the number of objects that can be tracked at one time.

⁶The problem of binding related data together.

NODE	STATES AND FUNCTION
$N_i(t)$	{not-near, nearby, close, very-close, touching} The proximity of object i to it's nearest neighbour at time t .
$R_{(i,j)}(t)$	{ignore, watch} The interestingness of the relationship between objects i and j .

Table 1: The DBN node types.

so that what was held at t is now held at $t-1$. When the peripheral-aspects become available, they are used to re-construct the DBN.

The DBN is composed of temporally separated subgraphs that are interconnected. DBN nodes are accessed via their respective subgraph, providing an implicit temporal reference relative to *now*. Each DBN subgraph has vertices (V) and edges that are denoted by edge-destinations (ED) and edge-sources (ES). Denoting edges in this way enables us to describe the links between subgraphs. In the example here, we will use only two subgraphs, however, a general definition is: $DBN = (TS, \{SG_1, \dots, SG_n\})$ where TS is a list of subgraph indices ordered to reflect their current temporal status. $SG_i = (V_i, ES_i, ED_i)$ where V_i are SG_i 's vertices, ES_i are SG_i 's edge-sources and ED_i are SG_i 's edge-destinations. The edge names are defined so that in addition to referring to the vertex-index they also refer to the relative time, e.g. edge-index = (deictic-time-index, vertex-index) where deictic-time-index $\in \{prev, now, next\}$.⁷ The construction of the DBN graph consists of two stages, **node-building** and **node-linking**, which are described with algorithms in the longer version of this paper [7].

Here we are using a DBN graph structure with two types of node, N and R, and with directed edges that describe one of two types of relationship, spatial or temporal. The directed edges hold the fixed conditional probabilities given in table 2. The spatial relationship is specified by the matrix $M_{R|N}$, capturing the belief that a proximity relationship becomes increasingly more interesting as two objects are identified as being nearer to each other. Figure 1(a) shows that when we evolve the network over time we make use of maintained official-observer object references to the image data. If, at the current time point, A and B are *still* near to each other we can form temporal links from the previous node to the current ones as shown in figure 1(b). From the relative temporal position of the previous time point the temporal links are to the next time point. These directed temporal edges each hold one of the matrices representing temporal continuity, here this is either $M_{N|N}$ for proximity, or $M_{R|R}$ for the pairwise relationship. Figure 1(c) shows how we extract a tree from the DBN graph which is done for the parsimonious reason that a tree is easier to evaluate than a multiply connected network.

⁷We also use t to refer to *now*, $t-1$ for *prev* and $t+1$ for *next*.

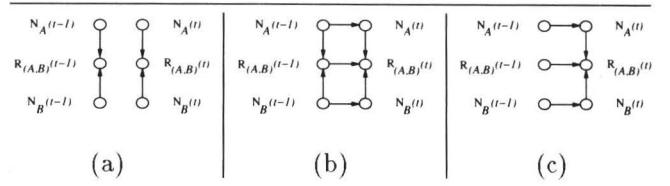


Figure 1: Graph linking structure.

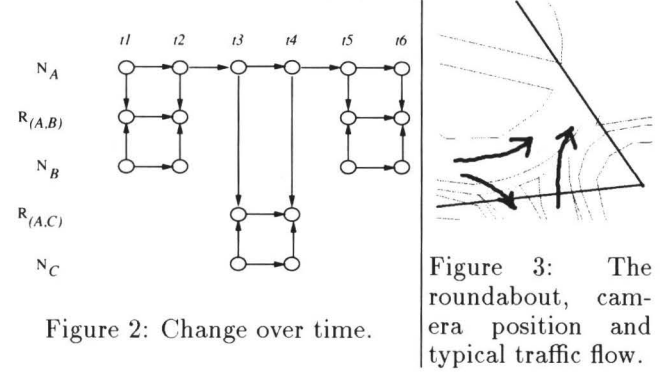


Figure 2: Change over time.

Figure 3: The roundabout, camera position and typical traffic flow.

This graph simplification preserves the results obtained from the previous value propagation allowing them to contribute effectively to the formation of the new belief about the relationship that now holds between object A and B . In the example developed in this paper each tree produced only holds values from the current and previous time points. If the DBN graph contains more than one relationship, then a tree is formed for each relationship. Once all propagations have completed the most interesting relationship can be selected.

The DBN network changes structurally over time to reflect the relationships between the objects of interest. For example, figure 2 shows what happens if N_B is missing for two time points ($t3$ and $t4$) due, say, to occlusion and the nearest neighbour to N_A now becomes N_C . When N_B is detected again it is identified as being nearest and the new links and nodes in the network are created to reflect this.

The DBN itself does not hold an overall picture of what is happening. For the creation of a fuller picture we use a separate Bayesian network called TASKNET.

3.2 Meta level

The DBN provides likelihood measures of which pairs of markers, if any, are worth further attention. Further attention involves the allocation of an agent, a limited resource, that we want applied to the objects that are the most relevant to the current official-observer task. To do this we use the results from the agent level, which are composed according to their DBN relationship. The resulting composite is input to the pair's dedicated TASKNET which builds a coherent interpretation of the temporally evolving object relationship. Each marker pair (i, j) , has a distinct TASKNET which we can denote $TASKNET_{(i,j)}$. The TASKNETs have a temporally fixed tree structure with conditional probabilities that are defined before runtime to reflect a preferential bias towards a feature that is deemed to be most in-

teresting. The input nodes represent key features relevant to the task the TASKNET has been constructed to identify. The output root node represents the overall belief, based on the evidence collected so far, in a set of candidates consisting of the wanted task, related but unwanted tasks, and the default unknown task.

Allocating an attentional process does the following: (1) ensure that a TASKNET is collecting information related to the pair, (2) ensure that an agent is running on both selected objects. When an uninteresting situation is recognised, we terminate attention by instantiating the value held in the pair's current DBN reference node. On the next clock tick when the DBN is updated and the inference algorithm run, if the relationship is still present, a high **ignore** value placed in the reference node at what is now $t-1$ will ensure that, unless a potentially interesting indicator is present at time t , the relationship will be ignored. This high **ignore** value is temporally propagated. Using this method, once a relationship is identified it can be ignored or continue to be watched. This attentional system has three properties that are used to guide the reasoning of the runtime system. (1) *Focus-of-attention* – only the selected set of target hypothesis and their associated functions are updated. (2) *Terminate-attention* – once a target hypothesis has been confirmed or denied to an acceptable degree of confidence we can stop all activities associated with it. (3) *Selective-attention* – we can dynamically select the current most interesting hypothesis to watch.

4 Road traffic example

The overall goal of the project is the formation of conceptual descriptions that capture what is happening in the scene. We would like this conceptual description to evolve in conjunction with the scene events and not be dependent upon the observation of a whole episode before saying what happened. The uses for this work include identification of unusual or illegal traffic behaviour where the system could be set up by the road side and used to select when to record relevant passages of video footage for subsequent analysis. Such intelligent surveillance of a scene must be capable of recognising the situated actions of the subjects being watched and measuring these observations against some model of correct behaviour. Although a surveillance system does not actively participate in the world, it still requires models of perception and action. These models need to incorporate suitable representations of space and time. The application domain of road traffic surveillance is selected on the pragmatic grounds of having access to data collected at a German roundabout and processed by the VIEWS project vision system (some details of which are given in [14]).

The results from the low-level vision system take the form of 3D pose positions which can be transformed to provide a ground plane projection. This makes it easier to reason about vehicle interactions, position and shape. Figure 3 shows the camera's field-of-view and, although less obvious, the camera position does affect the contents of the frame updates because we are dependent upon what is visible from the camera position not what is

visible from the overhead view.

The purpose of this example is to show that the DBN and TASKNETs can pick out a pair of vehicles that are performing an overtaking episode. This "overtaking" is likely to be non-intentional behaviour which just depends upon how the vehicles on the road are organised (and contributes to the uncertainty of identification). We can model a **I-am-being-overtaken** episode from five temporally ordered deictic event primitives [**behind-me**] [**behind-and-beside-me**] [**beside-me**] [**beside-and-infront-of-me**] [**infront-of-me**]. The conjunction of deictic references that point to the same thing is needed because we are not dealing with a subject that is a spatial point – a vehicle has spatial extent, a front, sides, and rear [5].

The DBN network will be using the proximity measure described earlier together with three conditional probability matrices shown in table 2, which are for the node types given in table 1. These matrices were derived from careful consideration of the possible input values. The results from the DBN (marker pairs of the form (i, j)) are used to select which agents to run on which markers. Agents produce two types of output, "effectors" and "agent-events". Effector output is used to select which attentional visual-operators to run on the next clock tick, and agent-event output is used as TASKNET input. Each agent-event is made of two components, the marker reference of the nearest object and the deictic event primitive $\in \{\text{behind-me, beside-me, infront-of-me, blocking-me}\}$. The input given to $\text{TASKNET}_{(i,j)}$ is derived from the composition (performed according to the rules given in table 3) of the agent-events provided by the agents running on markers i and j . In this example the TASKNETs have a very simple structure consisting of two nodes, one of each type given in table 4. $\text{CE}_{(i,j)}$ is the input node, $\text{LE}_{(i,j)}$ is the root/output node, and they are related according to the conditional probabilities given in table 5. The result from the agent-event composition is given as an evidence node to their TASKNET. The result of running the Bayesian inference algorithm is used to guide agent allocation as described earlier by identifying the likelihood that the relationship is either **overtaking**, **following**, **queueing** or **unknown**.

The high-level vision system described here has been implemented in Lisp. The results from the frame sequence in figure 4 are both incorporated in the figure and shown in table 6. The filled in vehicle shapes are uninteresting peripheral objects, the outlined shapes are selected objects, and the number near each vehicle is its marker reference. The sequence begins with two vehicles being selected because they are near each other. At frame 24, they are identified as following one another and are ignored. In frame 48, the pair (3,2) are selected because they were near each other in frame 36. In frames 48 to 72 the pair (3,2) are not mutually close together. At frame 84, possible overtaking or following is identified and identifying overtaking is given preference since it is the task objective. During frames 96 and 108 one of the vehicles occludes the other from the camera. By frame 132 overtaking is positively identified.

$M_{R N} =$		ignore	watch
	not-near	1.0	0.0
	nearby	0.5	0.5
	close	0.3	0.7
	very-close	0.2	0.8
	touching	0.0	1.0
$M_{R R} =$		ignore	watch
	ignore	0.9	0.1
	watch	0.1	0.9

$M_{N N} =$		not-near	nearby	close	very-close	touching
	not-near	0.6	0.3	0.1	0.0	0.0
	nearby	0.2	0.6	0.2	0.0	0.0
	close	0.0	0.2	0.6	0.2	0.0
	very-close	0.0	0.0	0.2	0.6	0.2
	touching	0.0	0.0	0.1	0.3	0.6

Table 2: DBN conditional probabilities for the road traffic example.

VIEWPOINTS		COMPOSITE	
behind-me	behind-me	→	back-to-back
behind-me	beside-me	→	trans-overtaking-back
behind-me	infront-of-me	→	following
behind-me	blocking-me	→	back-blockage
beside-me	infront-of-me	→	trans-overtaking-front
beside-me	blocking-me	→	possible-blockage
beside-me	beside-me	→	overtaking-passing
infront-of-me	blocking-me	→	head-blockage
infront-of-me	infront-of-me	→	head-on
blocking-me	blocking-me	→	impasse

$M_{LE LE} =$		overtk	follow	queue	unk
	overtaking	0.7	0.1	0.1	0.1
	following	0.1	0.7	0.1	0.1
	queueing	0.1	0.1	0.7	0.1
	unknown	0.1	0.1	0.1	0.7

Table 3: Composition rules.

NODE	STATES AND FUNCTION
$CE_{(i,j)}$	{back-to-back, trans-overtaking-back, following, back-blockage, trans-overtaking-front, possible-blockage, overtaking-passing, head-blockage, head-on, impasse} The composite event of the marker pair (i,j) .
$LE_{(i,j)}$	{overtaking, following, queueing, unknown} The likely-episode of the marker pair (i,j) .

$M_{LE|CE} =$

	overtk	follow	queue	unk
back-to-back	0.0	0.0	0.0	1.0
trans-overtaking-back	1.0	0.0	0.0	0.0
following	0.0	1.0	0.0	0.0
back-blockage	0.0	0.2	0.3	0.5
trans-overtaking-front	1.0	0.0	0.0	0.0
possible-blockage	0.0	0.2	0.2	0.6
overtaking-passing	1.0	0.0	0.0	0.0
head-blockage	0.0	0.2	0.3	0.5
head-on	0.0	0.0	0.0	1.0
impasse	0.0	0.1	0.5	0.4

Table 5: TASKNET conditional probabilities for the road traffic example.

Table 4: The TASKNET node types.

time	pairs	ignore	watch	m o	event	overtk	follow	queue	unk	likely episode	ignore
0	(1 . 0)	0.500	0.500								
12	(1 . 0)	0.500	0.500								
24	(1 . 0)	0.500	0.500	0 1	infront-of-me	0.100	0.700	0.100	0.100	following	t
				1 0	behind-me						
36	(1 . 0)	0.820	0.180	0 1	infront-of-me	0.100	0.700	0.100	0.100	following	t
				1 0	behind-me						
	(3 . 2)	0.500	0.500								
48	(1 . 0)	0.820	0.180								
60	(1 . 0)	0.705	0.295								
72	(1 . 0)	0.631	0.369								
84	(3 . 2)	0.500	0.500	2 3	behind-me	0.400	0.400	0.100	0.100	overtaking	
				3 2	infront-of-me						
				3 2	beside-me						
96											
108											
120	(1 . 0)	0.500	0.500								
	(3 . 2)	0.177	0.823	2 3	beside-me	0.700	0.100	0.100	0.100	overtaking	
				3 2	infront-of-me						
				3 2	beside-me						
132	(3 . 2)	0.100	0.900	2 3	beside-me	0.700	0.100	0.100	0.100	overtaking	
				3 2	beside-me						
144	(3 . 2)	0.118	0.882	2 3	beside-me	0.700	0.100	0.100	0.100	overtaking	
				3 2	beside-me						
156	(3 . 2)	0.120	0.880	2 3	beside-me	0.700	0.100	0.100	0.100	overtaking	
				3 2	beside-me						

Table 6: The results.

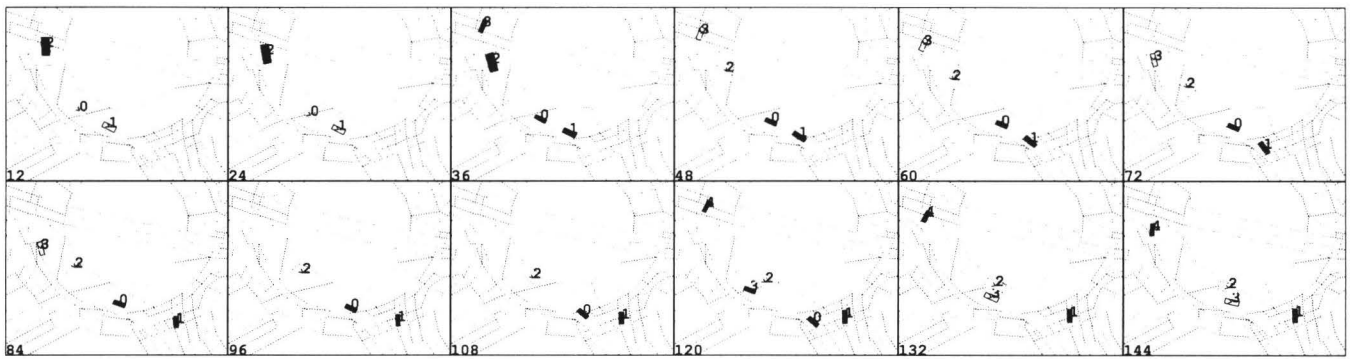


Figure 4: Selecting which pairs of vehicles are involved in overtaking.

5 Discussion and Conclusion

Many of the aspects of the attentional control system outlined in this paper have been influenced by previous work. The use of key features and markers is based on work by Agre and Chapman [1] although it was necessary to extend their techniques to include multi-agent viewpoints which contribute to the official-observer view for surveillance. The imposition of the local viewpoints in the deictic representation of spatial relationships also simplifies and decomposes the possible interactions between moving objects to allow an efficient response. Dean et al. [3] first proposed DBNs using clique trees and a fixed, temporally-repeated tree structure but here a more general, dynamic restructuring is required to model the set of interactions. Extracting the relevant subgraph (tree or singly-connected network) by a priori identification of older edges which can be removed seems novel and extending this to more complex DBN graphs is challenging! The use of a task-based Bayesian network is mentioned in Rimey and Brown [12] where it is used to actively direct the camera using geometric relationships. Here, however, our application requires scene surveillance based on spatial (and temporal) reasoning relative to a static camera. The combination of dynamic network propagation with task-related evaluation promises a highly effective attentional control mechanism which can be distributed under the agent formalism to deliver real-time performance although we are interested in the competence at the moment.

In particular, the results reported in section 4 show that the approach works well for simple tasks like monitoring overtaking episodes in the roundabout sequence. Future work will investigate (1) how well it copes with more complex, multi-object interactions and (2) whether the task-level can be extended to recognise temporal sequences of tasks which have a plan-like structure. In addition, instead of just starting from the precomputed pose positions, it would be more effective to experiment with a running vision system that was actively interfaced to the visual-operator primitives. The attentional mechanism could then help reduce the visual search involved in computing the pose information [14]. Visual search is a notoriously complex problem [13] which would benefit from guidance that exploits knowledge of the task to be

performed.

References

- [1] P. E. Agre and D. Chapman. Pengi: An implementation of a theory of activity. In *AAAI 87*, pages 268–272, 1987.
- [2] K. Bühler. The deictic field of language and deictic words. In R. J. Jarvella and W. Klein, editors, *Speech, Place, and Action: Studies in Deixis and Related Topics*, pages 9–30. John Wiley, 1982.
- [3] T. Dean, T. Camus, and J. Kirman. Sequential decision making for active perception. In *Proc. Image Understanding Workshop*, pages 889–894, 1990.
- [4] J. J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin Company, 1979.
- [5] R. J. Howarth. The control of spatial reasoning for high-level vision. In *ECAI-92 Workshop on Spatial Concepts*, 1992.
- [6] R. J. Howarth and H. Buxton. An analogical representation of space and time. *Image and Vision Computing*, 10(7):467–478, 1992.
- [7] R. J. Howarth and H. Buxton. Selective attention in dynamic vision. Technical Report 610, Queen Mary and Westfield College, 1992.
- [8] A. E. Nicholson and J. M. Brady. The data association problem when monitoring robot vehicles using dynamic belief networks. In *ECAI-92*, pages 689–693, 1992.
- [9] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [10] Z. W. Pylyshyn and R. W. Storm. Tracking multiple independent targets: Evidence for a parallel tracking mechanism. *Spatial Vision*, 3(3):179–197, 1988.
- [11] G. Retz-Schmidt. Various views on spatial propositions. *AI Magazine*, 9(2):81–97, 1988.
- [12] R. D. Rimey and C. M. Brown. Where to look next using a Bayes Net: Incorporating geometric relations. In *ECCV-92*, pages 542–550, 1992.
- [13] J. K. Tsotsos. On the relative complexity of active vs. passive visual search. *International Journal of Computer Vision*, 7(2):127–142, 1991.
- [14] A. Worrall, M. R.F., S. G.D., and B. K.B. Model-based tracking. In *British Machine Vision Conference 1991*, pages 310–318. Springer-Verlag, 1991.