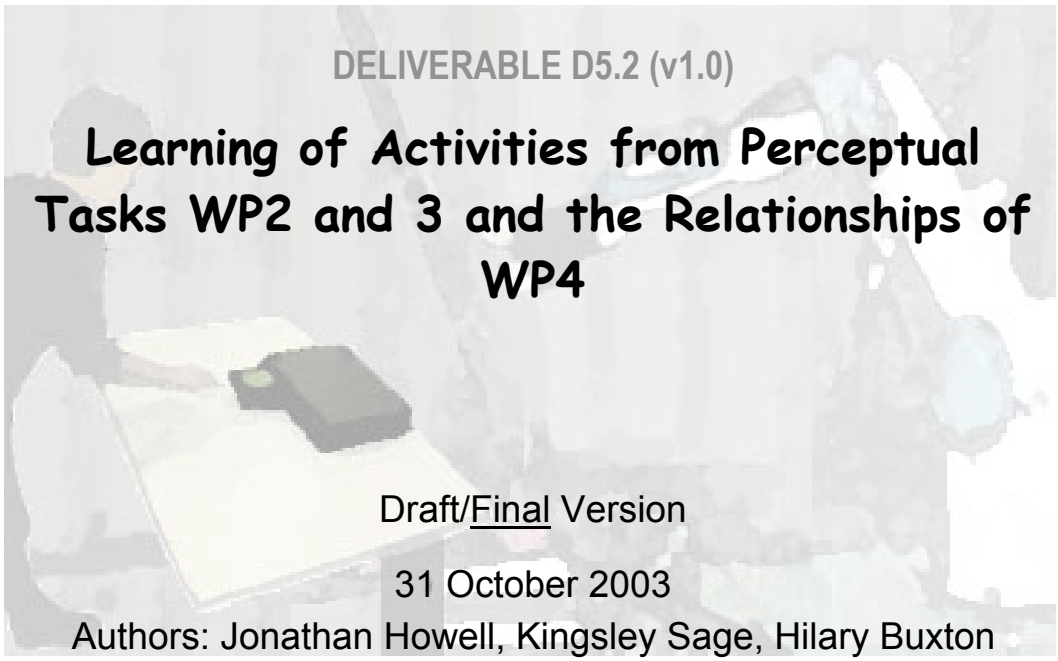

DELIVERABLE D5.2 (v1.0)

Learning of Activities from Perceptual Tasks WP2 and 3 and the Relationships of WP4

Draft/Final Version

31 October 2003

Authors: Jonathan Howell, Kingsley Sage, Hilary Buxton

Project acronym: **ACTIPRET**
Project full title: **Interpreting and Understanding Activities of Expert Operators for Teaching and Education**

Action Line IV.2.1: **Real Time Distributed Systems (Cognitive Vision)**
Contract Number: **IST-2001-32184**

Contents

1 Introduction 3
2 Overview of Recognition of Activities 3
2.1 Control Policy 4
2.2 Visual Index 5
2.3 Activity Hypothesis Creation 5
2.4 Hypothesis Control and Concept Creation 6
3 Automation and Learning 8
3.1 Learning Activities 8
3.2 Learning in the expert phase 8
3.3 Learning the structure of activities in the learning phase 9
4 Future Work 10
5 References 11

1 Introduction

This deliverable document provides details of the learning of activities in the Activity Reasoning Engine (ARE) of the ActIPret Demonstrator (AD) from perceptual tasks WP2 and 3 and the relationships of WP4. Section 2 describes how activities are currently recognised using reasoning based on recognition, rules and hand-coded Bayesian Belief Networks (BBNs) with a set of spatial relationships. Section 3 discusses how these principles can be adapted for learning.

2 Overview of Recognition of Activities

This deliverable develops the ideas introduced in Deliverable 5.1: the form of the conceptual language and activity definitions for use within the ActIPret Demonstrator (AD). These allow the production of activity plans from observed scenarios. The current implementation of the ARE is for `expert mode' operation only

The activity planning functionality originally intended for the Activity Reasoning Engine (ARE) component has been transferred to the USER(HMI) component. The ARE remains responsible for activity recognition and attentional control, and USER(HMI) still controls the AD operational phase (see Deliverable 1.3).

Fig. 1 shows how activity reasoning is performed in the ARE. Early attentive and full attentive cues based on hand gesture information and object relationships are requested from the GR and ORG components. Once received, these cues either create or update current activity hypotheses in conjunction with object state information contained in the Visual Index. Confirmed hypotheses become concepts and are passed to the USER(HMI) to be added to the scenario activity plan. Any object references attached to them are retained as Permanent Objects, see Section 2.2, which can be cross-referenced with later activities.

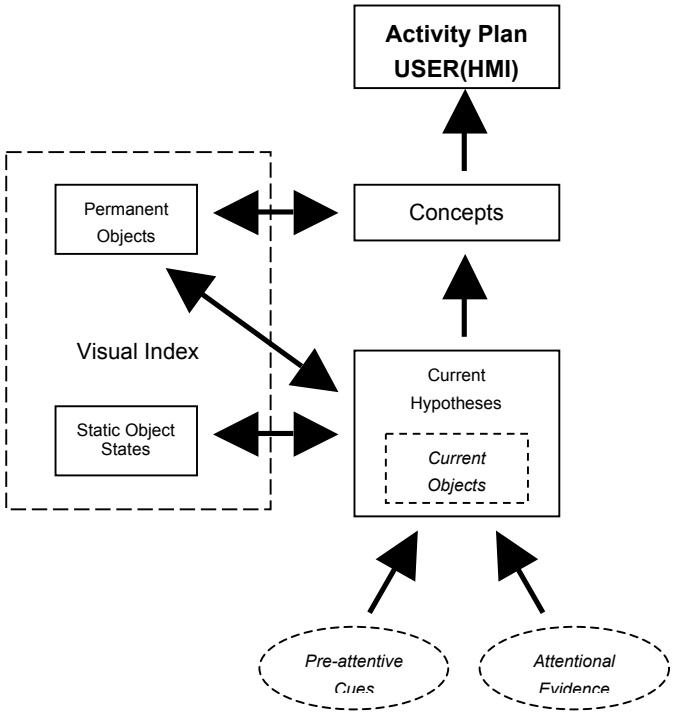

Figure 1. Diagram of activity dataflow in the ARE.

2.1 Control Policy

The Control Policy controls the overall operation of the AD and is represented a hierarchy of meta rules or overriding priorities for processing across the entire framework. Currently, we use simple production rules but these could be replaced by a Dynamic Decision Network (DDN).

Three rules are currently used in the ARE:

- **Rule 1:** IF (no beliefs) THEN 'Find a Hand'

  'Find a Hand' is implemented via the *'HandInfo'* service, connecting the GR and ARE. If the service has not been established, it is started. This in turn, starts a *'TrackObject'* service between the HT and GR, which provides hand candidate information. The existence of such hand candidates will change the internal ARE object state 'HandExists' from false to true.

- **Rule 2:** IF ('HandExists' is true) THEN 'Traverse Hypothesis Creation Tree'

  This rule controls the creation of activity hypothesis Dynamic BBNs according to various hand states and predictive cues, see Fig. 2. Once the ARE has a hand reference object, it can start the two Object Detector and Tracker (ODT) component services: *'ObserveObjectRelations'* and *'GetLostFoundObjects'*, see Section 2.4.

- **Rule 3:** IF (predictive cues found) THEN 'Update Current Hypotheses'

This rule minimises computational load by only evaluating current activity hypotheses when relevant new information is available. Hypotheses that 'time out' due to lack of confirmatory evidence can either be allowed to persist until some unrelated new activity begins or cleared in a periodic round-up in slack processing time. Fig. 3 shows an example hypothesis BBN.

2.2 Visual Index

The Visual Index is a store of mid- or long-term information, everything relevant to the current scenario that cannot be stored in the short-term activity hypotheses, which are discarded after use. The two main parts of the Visual Index are the permanent object references and internal state information for the scenario, see Fig. 2:

The Permanent Objects are object references linked to activity concepts, which need to be retained throughout the life of the scenario for the construction of the Activity Plan at the end. The permanent object references can also be used by current activity hypotheses, allowing any object to be referred to by more than one concept.

The internal states for objects in the scene consist of two parts:

- For a variable number of hands: hand(s) exist, hand(s) empty/full, hand(s) moving/static.

- For any manipulable object(s) (in this case, cdplayer): object-specific state (eg. on/off, open/closed, empty/full) as defined in the scenario definition file.

2.3 Activity Hypothesis Creation

The creation of new activity hypotheses is based on pre-attentive cues only, see Fig. 2.

Low-level information about hand candidates provided via the *'HandInfo'* service is used to update two internal states within the Visual Index: 'HandExists' and 'HandMoving'.

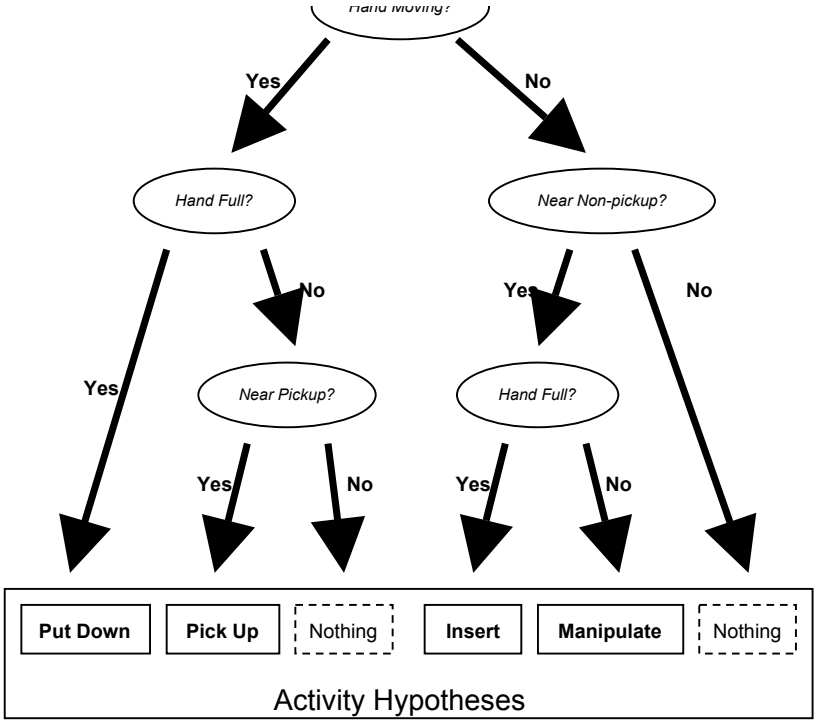

Figure 2. Binary tree for generation of activity hypotheses in the ARE, based on early attentive hand and object cues.

Four early attentive cues are used:

- **Hand Moving** – this is flag that is set whenever the change in position of the hand(s) in the scene goes above some threshold value. This is a fundamental predictive cue, as no new hypotheses can be created without it.

- **Hand Full** – this is maintained as an internal state within the Visual Index. Initially, it is assumed that any hands are empty when first found in the scene. They can switch to full only after a 'Pick Up' activity has been observed, and then switch to empty again only after a 'Put Down' activity.

- **Hand Near Pickupable Object**

- **Hand Near Non-Pickupable Object**

2.4 Hypothesis Control and Concept Creation

Rule 3 of the Control Policy (see Section 2.1) restricts the updating of current activity hypotheses. Any hypothesis that has a terminal query node value above a certain threshold is converted into a concept.

When a concept is created, any object references attached become permanent: this allows common references in the Activity Plan to be unified at the end of the scenario. Any other hypotheses contained in the ARE are then deleted, based on the reasoning that only one activity can be observed by the AD at any one time (mutual exclusivity).

Activities are represented internally as Bayesian Belief Networks (BBNs) using a causal inference forward chaining procedure (from evidence to query variables) defined over Directed Acyclic Graphs (DAGs). An example BBN for the activity 'pickup' is shown in Fig. 3.

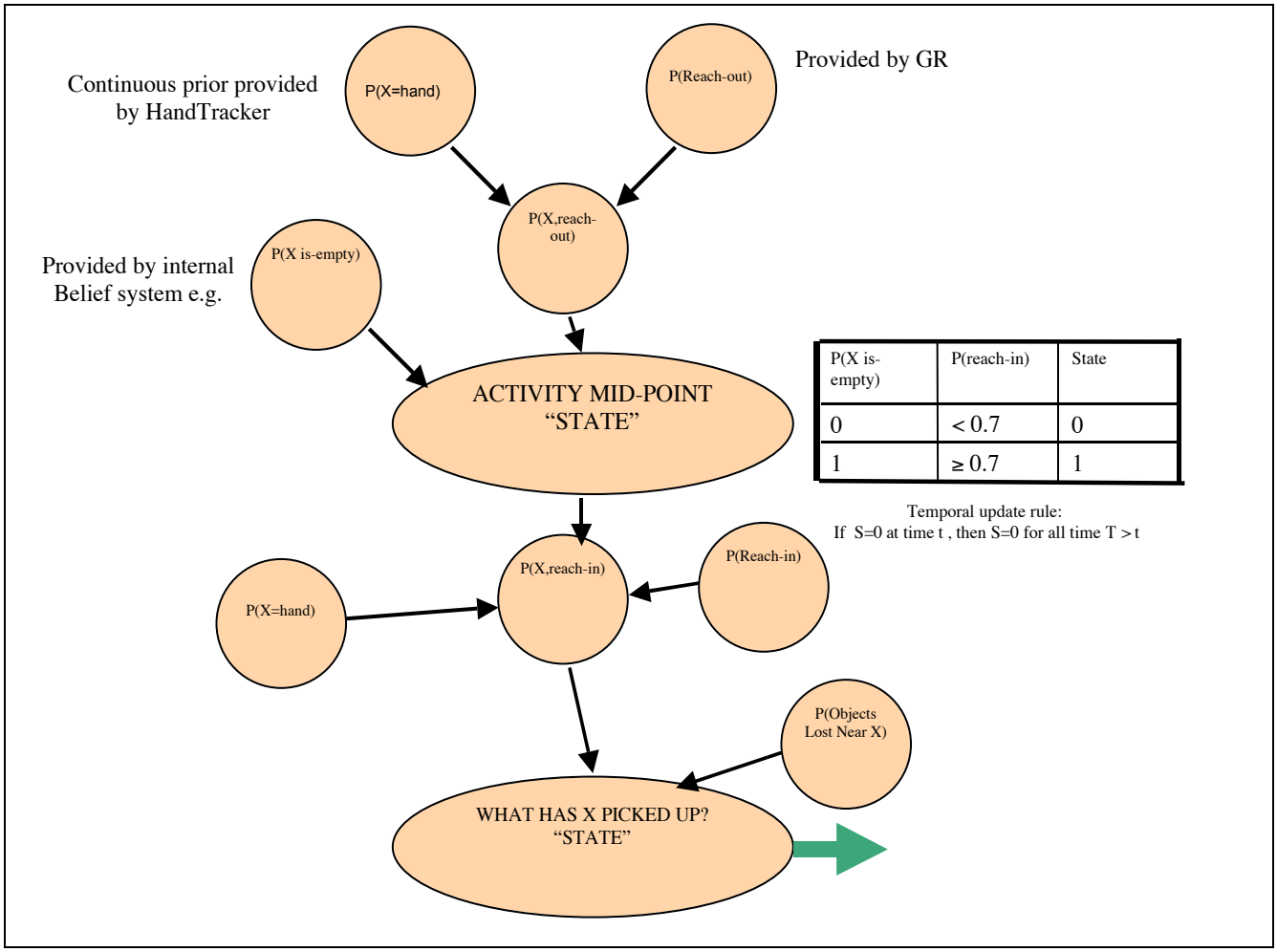

Figure 3. Example activity BBN representing the 'pick-up activity.

The root nodes of the BBN correspond to data derived from the early attentive and full attentive cues. The BBN then specifies how to combine this data to probabilistically hypothesise that the activity has occurred. The directed arcs of the BBN specify conditional independence assumptions. The probability value associated with the terminal leaf or 'query' node quantifies belief that the activity has occurred.

As well as taking data in real time from the root nodes, BBNs can have additional internal nodes that enable them to represent emerging activities in a useful manner. The BBN in Fig. 3 represents an activity at just one time step. This model is then rolled out over time to provide a Temporal Bayesian Belief Network (TBBN), or Dynamic Belief Network (DBN). In the case of 'pickup' this involves propagating each node to itself from one time step to the next via a temporal update rule. For the root nodes, this simply means ensuring that these values are updated in real time. The temporal update rules for the internal nodes are more complex and determine the BBNs ability to represent sequenced activity sub-structure. An example of such a temporal update rule is shown in Fig. 3 for the internal 'mid-point state'.

The rule specifies that the value of this node ‘latches’ at 1 when the necessary input conditions are achieved. This value remains latched even when the input nodes are no longer subsequently satisfied. This rule allows the BBN to represent two serial elements of activity sub-structure in one compact model. The Activity Reasoning Engine will require additional mechanisms to manage such hypotheses to ensure that partly fulfilled hypotheses do not remain in the system unchecked. We plan to use a temporal erosion scheme so that the belief in such internal nodes decreases as a function of time. Partly fulfilled hypotheses that do not result in concept formation within a set period will be garbage collected.

Currently in the AD framework, the ‘RecogniseGesture’ service supplied to the ARE from the GR component is implemented by a TDRBF network [1,4]. This provides confidence levels for either the ‘Reach In’ or ‘Reach Out’ gestures having been observed. To cope with lateral movements, it is planned to add a positionally-independent third gesture.

The activity BBNs use 4 types of full attentive evidence:

- **Object Lost Near Hand** - The ORG reports the loss of any tracked object that had been sufficiently close to the hand(s) in the scene. This cue provides confirmation for a `Pick Up' activity hypothesis via the `hand empty' internal state changing to `hand full', allowing it to be transformed into a concept.

- **Object Found Near Hand** - The ORG reports any new objects found to the ORG. This cue will provide crucial confirmation for a `Put Down' activity hypothesis, allowing it to be transformed into a concept.

- **‘Reach In’ Gesture Observed**

- **‘Reach Out’ Gesture Observed**

3 Automation and Learning

3.1 Learning Activities

The learning of activities is a key aspect of ActIPret and arises at 2 different scales:

- In the expert phase where we are interested in learning the sequencing of activities to form a high level description of the scenario as an Activity Plan;
- In the learning phase where we are interested in learning the structure of the activities themselves.

3.2 Learning in the expert phase

In the expert phase, the structure of the activity BBNs is known apriori and we are concerned primarily with determining the correct sequencing of activities to form the high level description of the expert carrying out the scenario i.e. the Activity Plan. This description is formed from two different types of information:

- declarative data that declares the existence of cetain objects and their properties; and
- the sequences of concepts derived from the activity BBNs.

The declarative data represents additional apriori scenario domain knowledge that is relevant to carrying out the scenario task. Some of this data can be derived from the early attentive cues (such as there exists a hand with a given reference label) and some of it is non-visually derivable data that is provided by an external agent (such as objects with categorisation CDs can be picked up and objects with categorisation CD-player cannot be picked up).

The number and topology of concept sequences will depend on the number of scenario exemplars and subsequent interaction by the expert. If the expert teaches the system just one scenario exemplar, then the sequence data would just consist of 1 linear set of concepts. This single ordering of activities would be the only one that the system would know about and be able to measure against in the tutor phase. Any deviation from that strict sequencing would be identified as an error by the novice user. If the expert teaches the system using a number of different activity sequences then the resulting Activity Plan would consist of a number of parallel, but independent, sequences. Parsing of the Activity Plan during the tutor phase would enable the system to determine whether the novice user's activity sequence was compatible with any of the parallel sequences during the tutor phase.

A further level of sophistication would enable to expert to demonstrate scenario sub-sequences with a hierarchical overall structure. The sub-sequences would correspond to sections of activities that could be carried out in a number of different temporal orders. Learning at this level of sophistication would require additional input from the expert in the form of sub-structure begin and end markers and probably some kind of visual Activity Plan editor.

3.3 Learning the structure of activities in the learning phase

The discussion thus far has assumed the availability of hand coded activity BBNs that use a combination of root nodes derived from early attentive and full attentive cues and internal nodes to determine whether an activity has occurred. The activity BBNs are coded as encapsulated C++ classes and are intended to be scenario independent as far as practicable.

It is desirable to be able to learn the structure of relevant activity BBNs during the learning phase. That is, the expert provides visual examples of what it means to pickup an object and the system generates its own activity BBN accordingly. This is a very challenging task. In the expert mode, the system relies on the Control Policy meta rules to know which services to switch on and off. These meta rules then determine which early attentive and full attentive cues are available. The presence of these cues then provide root node inputs to the activity BBNs. If we want to learn activity BBNs in the learning mode, there are a number of issues:

- if we assume that the Control Policy rules are available, then the problem reduces to establishing activity BBNs that combine the cues as root nodes together with some additional internal structure and a set of temporal update rules that produce a network with a maximum probability output at the terminal node at the point in time when the expert signals that the activity is complete and lower values at the terminal nodes at other times; and
- if we assume no Control Policy knowledge, we need knowledge about as many cues as possible simultaneously as well as all structure and update rules for the previous

case. This would involve starting as many different services as possible for all objects and trying to extract the activity relevant cues and building the activity network structure accordingly.

Assuming no Control Policy task control is too difficult a problem to contemplate, but a scheme for the automatic deduction of activity network structure assuming a Control Policy is feasible, although challenging. It would require both positive and negative examples of the activity and some additional input from the expert as to how successful completion of the activity affects some of the early attentive cues (e.g. the early attentive cue 'hand full' is inferred by the successful completion of the pickup activity. There is no component within the system that is capable of determining the state of this cue by visual means alone). The learning process would also require the expert to somehow specify the value of the terminal node as a learning goal. This could be a Boolean proposition (the terminal value is 0 and becomes 1 at some time t) or continuous (the expert uses a graph to visually determine a value for the terminal node as a function of time, using the continuous value to indicate that the activity is becoming more or less likely as it emerges with time). With these inputs, the task of learning the activity BBN becomes one of finding an activity model in a space of all possible models with root nodes, internal nodes and temporal update rules that best reproduce the terminal node values specified over time with the additional constraint that the model be as reasonably small as possible. This last point is important as otherwise the learning process would use an arbitrarily large system of internal nodes that may well be able to produce the desired functionality but tell us little about the real structure of the activity, and would not provide a readily interpretable or computationally efficient solution. This is a similar process in principle to Structural Expectation Maximisation (SEM) learning [2] for general Bayesian Networks.

The fit between a candidate activity BBN and the training data would be determined by a scoring process. This process would reward correct classification of positive and negative examples but also incorporate a penalty term for network complexity (number of internal nodes). There are a number of types of search space variables:

- **The root nodes:** a small number in the case of the pickup example;
- **Node connectivity:** can be represented as a N*N matrix where N is the number of nodes although we do have the constraint that the network is a DAG;
- **How to combine multiple inputs to any node:** this could take the form of a simple Conditional Probability Table (CPT) in the case of all discrete parent nodes, or more complex parameterised schemes such as Gaussians or softmax (logistic) functions in the case of continuous or mixed discrete and continuous parent nodes; and
- **Temporal update rules for each node:** could be as simple as update a real time value to an arbitrarily complex activation function based on node values in current and previous time steps.

As the search space for even simple activity BBNs is highly combinatorial, we will investigate whether optimisation techniques such as genetic algorithms are appropriate.

4 Future Work

Even with multiple active cameras, there will remain problems of ambiguity and occlusion that we believe could be solved with an eye-tracking camera. Cognitive studies of eye

movement [3, 5-7] suggest that the subject's focus of attention is engaged in on-going purposeful activty and that eye movements are closely related to the objects being manipulated in the tasks. Our initial tests conducted using the CD scenario have shown clear prediction of future activities using gaze position.

There will still be insufficient information from visual input for the effective recognition of some fine motor-control activities, such as pressing buttons. We can go some way in deducing what the intent of the subject is, but this will be essentially guesswork, reliant on compliancy (we assume our experts will not try to deceive us).

5 References

1. Buxton, H., Howell, A. J. and Sage, K. (2002) `The Role of Task Control and Context in Learning to Recognise Gesture', *Cognitive Vision Workshop*, Zürich, Switzerland, September 2002

2. Friedman, N. (1998), *'The Bayesian structural EM algorithm'*, in G. F. Cooper & S. Moral, eds., `Proc. Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI '98)', Morgan Kaufmann, San Francisco, CA.

3. Hayhoe, M. M., Shrivastava, A., Mruczek, R., & Pelz, J.B. (2003). `*Visual memory and motor planning in a natural task*'. Journal of Vision, **3**(1), 49-63.

4. Howell, A.J., Sage, K., and Buxton, H. (2003) `*Developing Task-Specific RBF Hand Gesture Recognition*', Proc. International Gesture Workshop, GW 2003, Springer Lecture Notes in Artificial Intelligence, Genova, Italy, 2003.

5. Johansson, R. S., Westling, G., Bäckström, A. and Flanagan J. R. (2001) `*Eye-hand coordination in object manipulation*'. J. Neurosci. 2001 **21**:6917-6932.

6. Johansson, R. S. (2003) `*Use of Vision in the Control of the Hand in Manipulation*', Active Vision 5 workshop, University of Sussex, Sept 2003.

7. Land M, Mennie N, Rusted J. (1999) `*The roles of vision and eye movements in the control of activities of daily living*', Perception. **28**(11):1311-28.

8. Sage, K., Howell, A.J. and Buxton, H. (2003) `*Developing Context Sensitive HMM Gesture Recognition*', Proc. International Gesture Workshop, GW 2003, Springer Lecture Notes in Artificial Intelligence, Genova, Italy, 2003.