

Content-based control of goal-directed attention during human action perception

Yiannis Demiris and Bassam Khadhour^{*}

Abstract

During the perception of human actions by robotic assistants, the robotic assistant needs to direct its computational and sensor resources to relevant parts of the human action. In previous work we have introduced HAMMER (Hierarchical Attentive Multiple Models for Execution and Recognition) (Demiris and Khadhourⁱ, 2006), a computational architecture that forms multiple hypotheses with respect to what the demonstrated task is, and multiple predictions with respect to the forthcoming states of the human action. To confirm their predictions, the hypotheses request information from an attentional mechanism, which allocates the robot's resources as a function of the saliency of the hypotheses. In this paper we augment the attention mechanism with a component that considers the content of the hypotheses' requests, with respect to the content's reliability, utility and cost. This content-based attention component further optimises the utilisation of the resources while remaining robust to noise. Such computational mechanisms are important for the development of robotic devices that will rapidly respond to human actions, either for imitation or collaboration purposes.

^{*}Yiannis Demiris and Bassam Khadhourⁱ are with the Department of Electrical and Electronic Engineering, Imperial College London y.demiris@imperial.ac.uk

1 Introduction

To increase the utility and flexibility of robotic assistants, it is useful to equip them with the capacity to understand the actions of humans in their environment. Such understanding can lead to easy programming of robotic assistants to perform new tasks through human demonstration (Demiris and Johnson, 2003; Schaal et al., 2003), as well as the potential for collaborating with humans to perform a common task. This quest leads to several challenges: while observing a demonstration, it is not clear what aspect of the demonstration needs to be attended too, either for imitating it or for determining the best course of action in a collaboration scenario. As a result, there is no obvious top-down (goal-directed) control as to where to direct the robot's attention. While saliency-based approaches for the control of attention (Itti and Koch, 2000) can be very useful, for example by directing the attention to the most rapidly moving object, a more goal-directed attentional control is needed (Demiris and Khadhour, 2006).

In the next few sections we will review our approach to the goal-directed control of attention (Demiris and Khadhour, 2006), in dynamic scenes involving human actions. We subsequently present our experimental setup consisting of a robot camera observing a human during different behaviour demonstrations, and demonstrate how a *content-based* control of top-down (or goal directed) attention requests can optimise the use of the computational and sensor resources of the observer robot.

2 BACKGROUND

An important issue that remains unresolved in imitation and collaboration research is where the attention of the observer should be focused when a demonstrator performs an action. Whilst there is little agreement on a universal definition of attention (Tsotsos, 2001), from an engineering point of view it can be defined as a mechanism for allocating the limited perceptual and motor re-

sources of an agent to the most relevant sensory stimuli. The control inputs to the attention mechanism can be divided into two categories: stimulus-driven (or bottom-up) and goal-directed (or top-down). Stimulus-driven attention models work by attaching levels of saliency to low-level features of the visual scene, e.g. colour, texture or movement, and then deriving the corresponding saliency maps, fusing them, and applying a winner-take-all strategy to direct the attention to the most salient part of the scene (Itti and Koch, 2000). However, it is well known from human psychophysical experiments that top-down information can also influence bottom-up processing (e.g. (Wolfe, 1994; Treue and Trujillo, 1999)). The top-down information can be derived from the task requirements in hand, or from other sources of knowledge about the observed action, for example from linguistic descriptions (Steels and Baillie, 2003). Wolfe (Wolfe, 1994) put forward the hypothesis that the goal-directed element of attention selects bottom-up features that are relevant to the current task (in what is termed visual search) by varying the weighting of the feature maps. However, the fundamental question of what determines the features that are relevant to the task has not been answered in a principled way. This is the case particularly when the tasks are performed by someone else without the observer having access to the internal motivational systems of the demonstrator. The task demonstrated is not known in advance, therefore the top-down selection of features to attend to is not obvious, and an online method for selecting features dynamically is needed.

In previous work we have utilised the motor theory of perception (Demiris and Hayes, 2002) for deriving a principled mechanism for the goal-directed control of attention during action perception. The particular method we employ in the HAMMER architecture relies on the concept of motor simulation. Mental simulation theories of cognitive function (Hesslow, 2002), of which motor simulation is an instance, advocate the use of the observer's cognitive and motor structures in a dual role: on-line, for the purposes of perceiving and acting overtly, and off-line, for simulating and imagining actions and their consequences

(Jeannerod, 2001).

The key contribution of this paper is the extension of this mechanism to add content-based factors to the arbitration process.

3 Overview of the HAMMER architecture

HAMMER is organized around, and contributes towards, three concepts:

- The basic building block involves a pair of inverse and forward models (Narendra and Balakrishnan, 1997; Wolpert and Kawato, 1998; Demiris and Hayes, 2002). These are used in the dual role of either executing or perceiving an action, an idea first proposed and implemented in (Demiris, 1999; Demiris and Hayes, 1999).
- These building blocks are arranged in a hierarchical, parallel manner (Demiris and Johnson, 2003).
- The limited computational and sensor resources are taken explicitly into consideration: we do not assume that all state information is instantly available to the inverse model that requires it, but formulate them as requests to an attention mechanism. We will describe how this provides a principled approach to the top-down control of attention during imitation.

HAMMER is using the concepts of inverse and forward models (Narendra and Balakrishnan, 1997; Karniel, 2002; Wolpert and Kawato, 1998). An inverse model (akin to the concepts of a behavior, a controller, or action) is a function that takes as inputs the current state of the system and the target goal(s), and outputs the control commands that are needed to achieve or maintain those goal(s). On the other side of the spectrum a forward model (akin to the concept of internal predictor), is a function that takes as inputs the current state of the system and a control command to be applied on it and outputs the predicted next state of the controlled system. Inverse and forward models can be learned

through reinforcement learning, and motor babbling (Dearden and Demiris, 2005).

The building block of HAMMER is an inverse model paired with a forward model. When HAMMER is asked to rehearse or execute a certain action, the inverse model module receives information about the current state (and, optionally, about the target goal(s)), and it outputs the motor commands that it hypothesises are necessary to achieve or maintain these implicit or explicit target goal(s). The forward model provides a prediction of the upcoming states should these motor commands get executed. This estimate is returned back to the inverse model, allowing it to adjust any parameters of the action.

Interestingly, as first proposed and demonstrated in (Demiris, 1999; Demiris and Hayes, 1999), the pairs of inverse and forward models can be used in a dual role, to both plan and execute an action, as well as perceive it when demonstrated by another agent. For HAMMER to determine whether a visually perceived demonstrated action matches a particular inverse-forward model coupling, the demonstrator's current state as perceived by the imitator is fed to the inverse model. The inverse model generates the motor commands that it would output if it was in that state and wanted to execute this particular action. The motor commands are inhibited from being sent to the motor system. The forward model outputs an estimated next state, which is a prediction of what the demonstrator's next state will be. This predicted state is compared with the demonstrator's actual state at the next time step. This comparison results in an error signal that can be used to increase or decrease the inverse model's confidence value, which is an indicator of how closely the demonstrated action matches a particular imitator's action.

Multiple pairs of inverse and forward models can operate in parallel (Demiris, 1999; Demiris and Hayes, 1999, 2002). Fig. 1 shows the basic structure. When the demonstrator agent executes a particular action the perceived states are fed into all of the imitator's available inverse models. As described earlier, this generates multiple motor commands (representing multiple hypotheses as to what

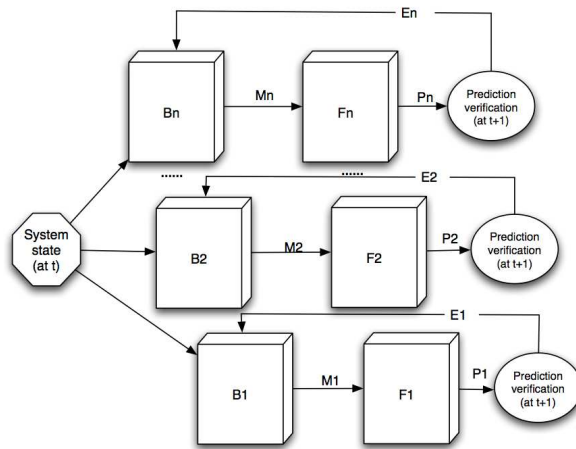


Figure 1: The basic architecture, showing multiple inverse models (B_1 to B_n) receiving the system state, suggesting motor commands (M_1 to M_n), with which the corresponding forward models (F_1 to F_n) form predictions regarding the system's next state (P_1 to P_n); these predictions are verified at the next time state, resulting in a set of error signals (E_1 to E_n)

action is being demonstrated) that are sent to the corresponding forward models. The forward models generate predictions about the demonstrator's next state: these are compared with the actual demonstrator's state at the next time step, and the error signal resulting from this comparison affects the confidence values of the inverse models. At the end of the demonstration (or earlier if required) the inverse model with the highest confidence value, i.e. the one that is the closest match to the demonstrator's action is selected. This architecture has been implemented in real-dynamics robot simulations (Demiris, 1999; Demiris and Hayes, 1999, 2002), and robotic platforms (Demiris and Johnson, 2003; Johnson and Demiris, 2004) and has offered plausible explanations and testable predictions regarding the behaviour of biological imitation mechanisms in humans and monkeys (review in (Demiris, 1999; Demiris and Johnson, 2007)).

4 Top-down control of attention

The architecture as stated so far assumes that the complete state information will be available for and fed to all the available inverse models. However, the

sensory and memory capacities of the observer are limited, and obtaining the complete state information is costly (for example, it might require multiple saccades), so in order to increase the efficiency of the architecture, we do not seek to obtain all the state information for all the inverse models. Since each of the inverse models requires a subset of the global state information (for example, one might only need the arm position rather than full body state information), we can optimize this process by allowing each inverse model to request a subset of the information from an attention mechanism, thus exerting a top-down control on the attention mechanism. Since HAMMER is inspired by the simulation theory of mind point of view for action perception, it asserts the following: for a given action, the information that the attention system will try to extract during the action's demonstration is the state of the variables the corresponding inverse model would have to control if it was executing this action. For example, the inverse model for executing an arm movement will request the state of the arm when used in perception mode. This novel approach provides a principled way for supplying top-down signals to the attention system. Depending on the hypotheses that the observer has on what the ongoing demonstrated task is, the attention will be directed to the features of the task needed to confirm one of the hypotheses. Since there are multiple hypotheses, thus multiple state requests, the *saliency* of each request can be made a function of the confidence that each inverse model possesses. This removes the need for ad-hoc ways for computing the saliency of top-down requests. Top-down control can then be integrated with saliency information from the stimuli itself, allowing a control decision to be made as to where to focus the observer's attention. An overall diagram of this is shown in figure 2.

In the experiments we reported in (Demiris and Khadhour, 2006), we investigated a number of arbitration mechanisms, and scheduling algorithms. We found that the best strategy to use for allocation was (what was termed as) RR-HCAW: initially the resources are distributed equally across behaviours, (RR: Round Robin algorithm), switching to a strategy that favours the behaviour

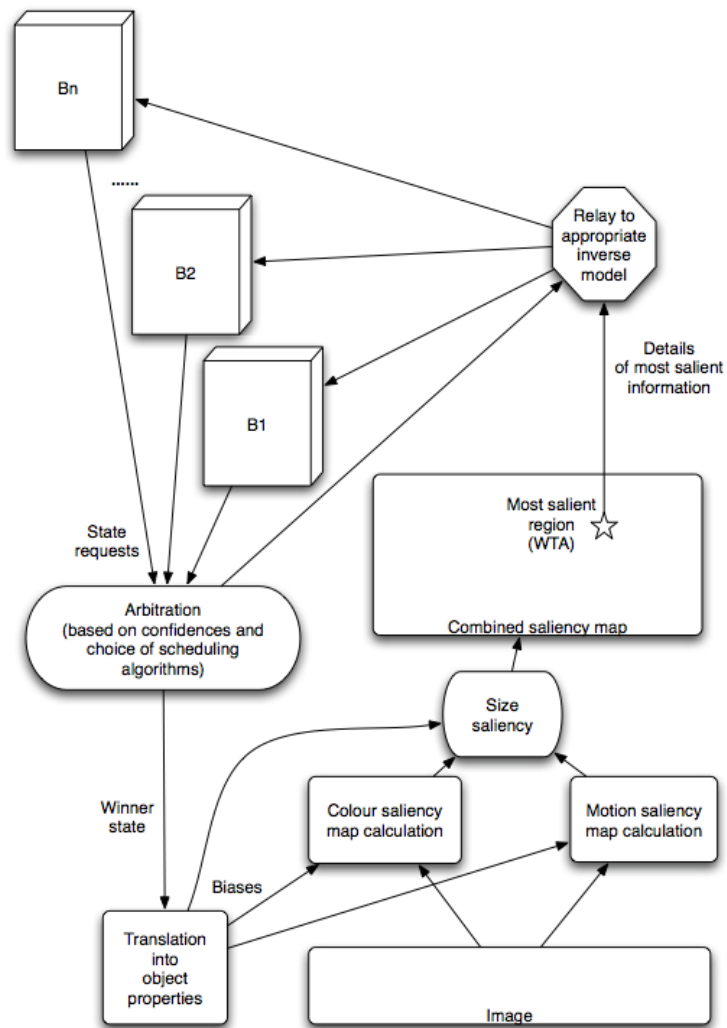


Figure 2: The HAMMER architecture (Demiris and Khadhour, 2006), incorporating the attention systems); B1 to Bn are the available inverse models, and the arbitration block has to decide which of their requests to satisfy.

with the highest confidence (HCAW: High confidence Always Wins) once such behaviour can be clearly identified.

5 Introducing the content based control of attention

In the work we have reported so far (Demiris and Khadhour, 2006), distribution of resources is done on a single behaviour basis; one behaviour is selected and its request is serviced. By introducing content based control, the architecture refines its selection process by going on step further and examining the *content* of each request, across all behaviours; as we will see, this will allow for better scheduling of resources, and significantly improves the performance characteristics of the architecture of (Demiris and Khadhour, 2006).

5.1 Adding content-based factors

In the work reported in this paper, in addition to the confidence values which are used again to reflect how well a hypothesised behaviour matches the demonstrated one, a number of new factors that add content-based control are used. We list below the new factors that are used, grouped in three categories: *Reliability* of content, *Utility* of content, and *Cost* of content; their usage can be seen in figure 4.

5.1.1 Reliability of content

- Is the request asking for the location of a dynamic or static object¹?

Dynamic objects would require much more frequent update than static objects, because their location can change constantly. A static object however is most likely to have its current location matching that of the memory from the last time it was seen. Hence it would not need frequent

¹A dynamic object is an object which can move by itself, e.g. the hand. Whereas a static object requires force from a dynamic object to move it.

updating, and a short-term memory component holding the details of the various object locations can be used (this can, for example, save us a saccade). We can thus give less priority to requests for locations of static objects that had their locations updated recently in the memory, giving more importance to dynamic objects' requests.

- When was the last time this request was attended to? Is the current location of the object asked for in this request valid? This knowledge can be used to validate whether the location of an object (be it static, dynamic, or static in contact with a dynamic object) has expired.
- Can a simple prediction of the object's location be performed?

While the object is not being attended to, a prediction is generated to estimate its current location. This is based on whether the object is static, dynamic or static in contact with a dynamic object. If the object is dynamic (or static in contact with a dynamic object), then a simple linear extrapolation method is used to predict its next location.

5.1.2 Utility of content

- How many behaviours are submitting this request?

More importance is given to a more popular request that is required by a larger number of behaviours. This is because it would help enable more behaviours to be processed.

- Will the content attended to also serve the request of the winning behaviour? In other words, is it being made by the behaviour that is currently selected by the scheduling algorithm? If it is, the request's priority is increased, since it would always remain top priority for the attention model to serve the most-likely behaviour, to ensure correct recognition of the demonstrated action.

- How many behaviours will be completely satisfied if this request is attended to?

This is different from “How many behaviours are requesting this request?” since a behaviour can have more than one request, meaning that answering a request may not be sufficient to completely satisfy the behaviour that made it. This information is only used to grade the importance of a request.

- When was the last time the behaviour was attended?

This ensures that behaviours which have been left unattended for a long time, are given a chance, even if they are not winning behaviours.

5.1.3 Cost of obtaining the Content

- How far is the requested content from the currently attended location, thus what is the cost of the saccade to the new location? This information is important in order to extend the model’s ability to deal with camera saccades to objects that lie outside the current camera input.

The previous memory of the object’s location is used, with the help of the prediction mentioned earlier, to estimate the distance (and direction) of the saccade needed in order to update its location. The distance of the saccade acts as a cost function, that is used as an input to optimal path criteria. This enables the determination of the shortest path needed to update the objects that needs saccading to.

Experiments that simulate saccades were also incorporated, and a saccade cost was introduced. This saccade cost was proportional to the distance the camera has to saccade in order to view the object. The cost was normalised to a value between 1 and 10. Therefore, for an image that has a width of x pixels and a height of y pixels, and assuming that the centre of the attention box is at (x_1, y_1) and the location for the object to saccade to is (x_2, y_2) , then the cost is calculated as:

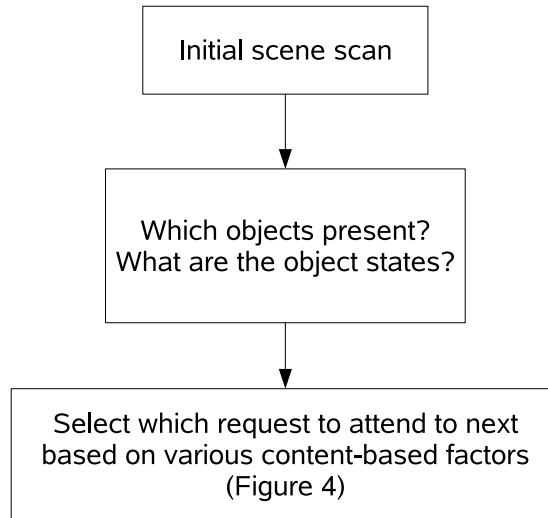


Figure 3: A general overview of the architecture

$$SC = \text{round}(10 \frac{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}{\sqrt{x^2 + y^2}})$$

The result of this calculation would freeze the allocation of resources for SC number of frames (which simulates the time it takes for the camera to do the saccade).

In some of the experiments shown later, the objects used were placed outside the visual scene to force saccades. Throughout the experiments, only a simulation of the saccade, with its associated cost value (SC) based on the distance of the object from the attention box was implemented.

Figure 3 shows the top-level of how the *content-based* approach will work. First, it is necessary to perform an overall initial scan to identify which objects (from the robot's database) are present in the scene and to identify the state of these objects, whether they are together, or isolated, etc. The content based control is subsequently activated, following the steps described in figure 4 which are described below. Based on the selection of this block, the behaviours that can be computed are then processed.

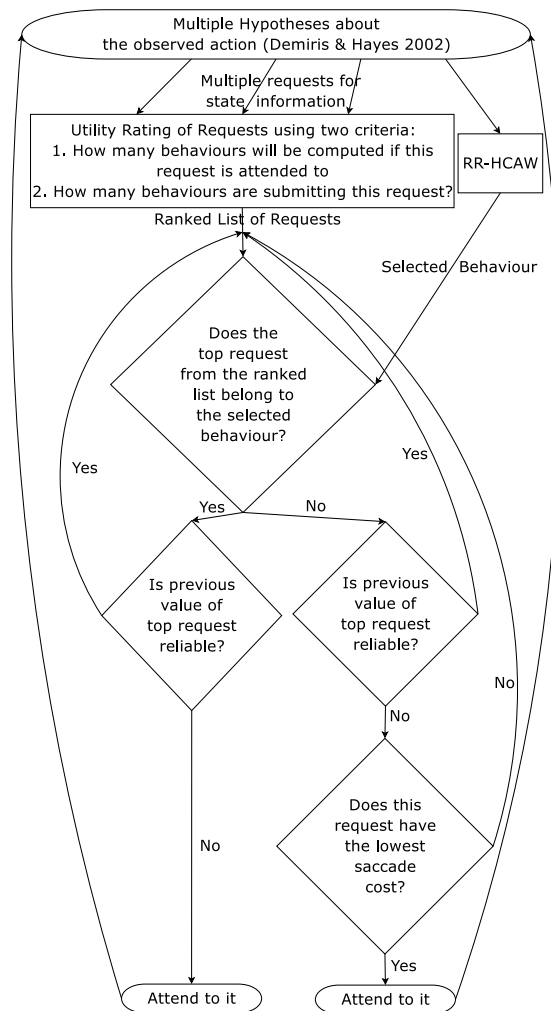


Figure 4: A detailed overview of the content-based control architecture

5.2 The content based control algorithm

There are two main goals that are desired from this new approach, listed here in their priority order:

- To give priority to the winning behaviour using HCAW scheduling algorithm when there is a clear winner. Otherwise giving priority to all behaviours using RR scheduling algorithm when there is no winner yet (which is often the case during the initialisation stages)². This goal will insure that the performance of the content-based control mechanism can, at least, maintain the performance of the previous work (Demiris and Khadhour, 2006)
- To maximise the number of behaviours being computed through intelligent allocation of the resources between the requests, whilst carrying out the first goal above.

Given the two goals above, the main part of the algorithm in figure 4 is the block, “Does the top request from the ranked list belong to the selected behaviour?”. The input to this block is a ranked list of requests and the selected behaviour from the RR-HCAW algorithm. If this request *is* being made by the winning behaviour, then it will be attended to, unless there is no need to, as it would only produce redundant information. One example for this is when a request is asking for the location of a *static* object that has only been attended to recently, and would therefore remain unchanged.

If the request however is *not* being made by the currently winning behaviour, then the algorithm will still attend to it, on the condition that it is being made by a behaviour that remained idle for some long period. In this case however, since there is time for the attention model to allocate resources to requests that serve a behaviour other than the winning behaviour, a further mini-competition takes place to find the lowest saccade cost. A simple optimal path algorithm

²On the implementation level, the RR-HCAW switching technique from the basic HAMMER implementation was used again to decide on when to give priority to every behaviour equally, and when to give priority to the winning behaviour.

is used to find the nearest location that requires the smallest saccade. If there are other similar requests that are nearby, requiring a small saccade, then these too are attended to, in order to minimise the overall number of saccades needed throughout the experiment. These further saccades will only continue to take place, however, if the winning behaviour is still being satisfied, or has only been compromised for a very short period of time.

As mentioned in the previous section, determining whether the request is worth attending to, or just a waste of time because of redundancy, was possible using the reliability of content factors. These factors are therefore used to serve the block labelled “Is previous value of top request reliable?” in figure 4.

The block at the top of the algorithm in figure 4 receives all the requests from all the behaviours. The function of this block is to try to prioritise these requests according to two factors: How useful it will be to answer it (i.e. how many behaviours will be computed as a result of attending to it) and How popular is the request (or how many behaviours are submitting it). Currently these are combined by giving priority to the more important one (namely, how many behaviours will be computed as a result). If then there are two requests that are equal, the next factor is used to make a decision (namely, how many behaviours are submitting it). If then there is still a draw, one of the two requests is chosen at random to go first.

The content-based architecture will have the advantage of enabling multiple behaviours to be computed at a time, instead of only one behaviour previously. As a result of this, the resulting graphs for the content based control should look more similar to the original graphs (without the attention mechanism), resulting in a smaller loss of information. The additional overhead of the scheduling algorithm however is very little the algorithm produces a ranked list of requests based on simple counter increment operations (of the constant order of $(\text{Max number of Requests} * N) + N$ in the worst case scenario (where all behaviours are asking all needed states, where N is the number of behaviors), followed by a maximum of three binary decision operations.



Figure 5: The experimental setup involves an ActivMedia Peoplebot observing a human demonstrator acting on objects

6 Experiments

6.1 Experimental Setup

We implemented and tested this architecture on an experimental setup involving an ActivMedia Peoplebot, following the setup of (Demiris and Khadhour, 2006; Demiris and Johnson, 2003); in these experiments the on-board camera was used as the only sensor. A description of the low-level implementational details of inverse and forward models, and the low-level vision processing can be found in (Demiris and Khadhour, 2006). A human demonstrator performed an object oriented action (an example is shown in figure 5), and the robot, using HAMMER, attempted to match the action demonstrated (an example is shown in figure 6), with the equivalent in its repertoire. In the experiments reported here, the robot captured the demonstration at a rate of 30Hz, with an image resolution of 160×120 , and the demonstrations lasted an average of 3 seconds. In the following sections, we will describe the performance of the architecture with and without two implementations of the attention mechanism (with and without content-based control). We will compare its performance against previous implementations of this approach (Demiris and Hayes, 2002; Demiris and Johnson, 2003) to demonstrate the performance improvements that the content-based attention-control subsystem of HAMMER brings.

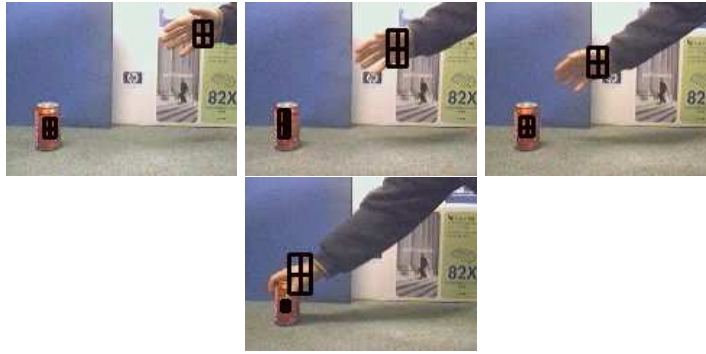


Figure 6: Representative frames from a video sequence involving a human reaching for an object

6.2 Demonstrated Behaviour Set

The following seven new behaviours were implemented in these experiments to demonstrate and test the model architecture explained earlier.

- Behaviour 1 - Move empty hand up; requests: Hand location only.
- Behaviour 2 - Move empty hand down; requests: Hand location only.
- Behaviour 3 - Move empty hand left; requests: Hand location only.
- Behaviour 4 - Move empty hand right; requests: Hand location only.
- Behaviour 5 - Put objects together; requests: Hand location, object A and object B.
- Behaviour 6 - Separate two objects; requests: Hand location, object A and object B.
- Behaviour 7 - Move two objects together; requests: Hand location, object A and object B.

In these behaviours, there are three objects, one dynamic (the hand), and two static (A and B). The hand has 7 requests in total (i.e. made by all behaviours), whilst objects A and B have 3 requests each. A coke can and an orange were chosen for objects A and B respectively.

6.3 Experimental Design

Three different sets of experiments were designed, in order to demonstrate how the added content-based control can improve the attention model in three different ways.

- Experiments to show that it will generally improve the recognition results for the winning behaviour. We performed three sets of experiments to demonstrate this point, for each of the 7 behaviours, with the following conditions:
 1. No attention mechanism - at each frame all the behaviours receive the state information they require, regardless of how long it takes (or how costly it is) to collect it, and all compute and update their confidences. This condition is the same used in previous experiments (Demiris and Hayes, 2002; Demiris and Johnson, 2003), and will serve as the reference condition for comparing the performance of the attention subsystem. Under this condition, the system contains complete knowledge of the whole state information and the resources to supply all hypotheses with the necessary information for them to be confirmed/dismissed. By definition, this condition will perform better than any attention system that we can devise, since any attention system will result in sub-sampling of the state information, thus cannot possibly result in more accurate recognition. However, these are idealistic circumstances, and embedded systems cannot be assumed to have the necessary resources to have complete state information. Thus the need for attention mechanisms, and the no-attention condition is included as the reference signal, i.e. the theoretically ideal condition that cannot always be achieved.
 2. RR-HCAW, the best performing strategy from the content-free allocation strategies (Demiris and Khadhour, 2006), which combines the Round-Robin (RR) implementation and the HCAW (Highest Confi-

dence Always Wins) condition. Initially all behaviours are given equal treatment (RR), but when one of them takes a clear lead (its confidence becomes higher than 50% of the average confidence of all behaviours with positive confidences), the arbitration mechanism switches to HCAW.

3. Content-Based approach, in addition to the mechanisms of experiment 2).

- Experiments to quantify explicitly the cost of involving saccades, which are necessary for more realistic situations where not all the information required are in the current field of view.
- Experiments to show that the content based control allows the system to recover from initial errors by (as much as possible) keeping track of other behaviours to allow them to recover later, where the *final* winning behaviour is not the same as the *initial* winning behaviour.

In all conditions, the same formula is used to update the confidence of the inverse models, and follows the procedure of Demiris and Khadhour (2006): the data requested, once retrieved, are sent to the corresponding inverse model that asked for them. The motor command of that inverse model is generated and sent to the corresponding forward model, which forms a prediction of what the next state will be. We then calculate the prediction error and use it to update the confidence of the inverse model. Since we are using a qualitative prediction (e.g. hand closer to target), the prediction will either be correct or not. The confidence of the inverse model is then updated according to the following rule:

$$C(t) = \begin{cases} C(t-1) + 1 & \text{if prediction is correct} \\ C(t-1) - 1, & \text{if prediction is incorrect} \end{cases} \quad (1)$$

Figure 7 serves as a legend for the other figures, and shows the assigned labels (B1 to B7) for the different behaviours.

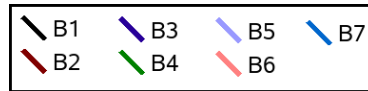


Figure 7: The corresponding labels for the behaviours used in the experiments

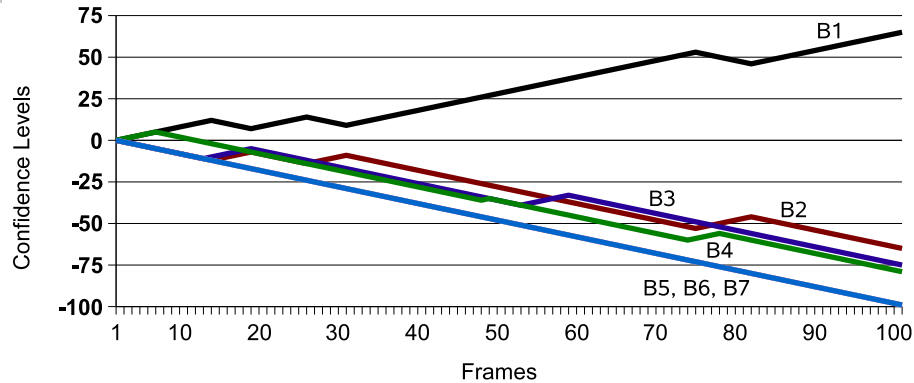


Figure 8: B1 demonstrated. Evolution of the confidences when no attention mechanism is employed

6.4 Experiments to show recognition improvement using content-based control

Figure 8 gives the evolution of the confidences of all the behaviours, whilst behaviour 1 (as an example) was being demonstrated, when no attention mechanism is employed.

Figure 9 gives the evolution of the confidences of all the behaviours, whilst behaviour 1 (as an example) was being demonstrated, using the attention model of our previous work (Demiris and Khadhour, 2006) that only employs resource scheduling algorithms working on the confidence values.

Figure 10 gives the evolution of the confidences of all the behaviours, whilst behaviour 1 (as an example) was being demonstrated, when the content-based approach is added.

Figure 1 gives a summary of the performance of the attention mechanism for each of the behaviours. In all conditions, all behaviours were recognised correctly. Results of the final confidence value of the demonstrated action are

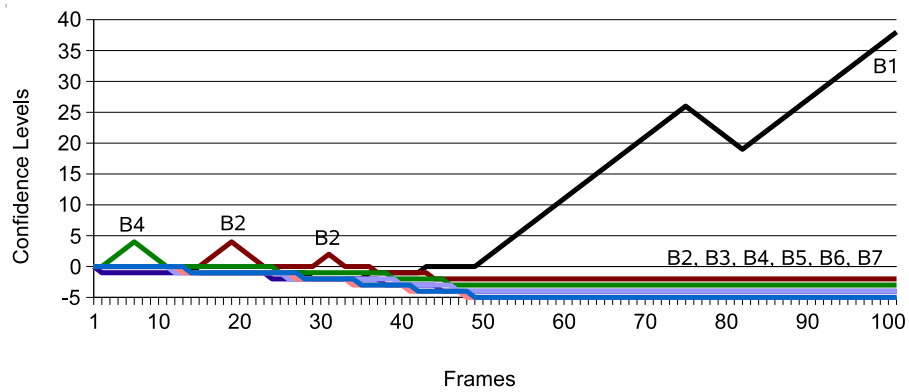


Figure 9: B1 demonstrated. Evolution of the confidences using the RR-HCAW condition

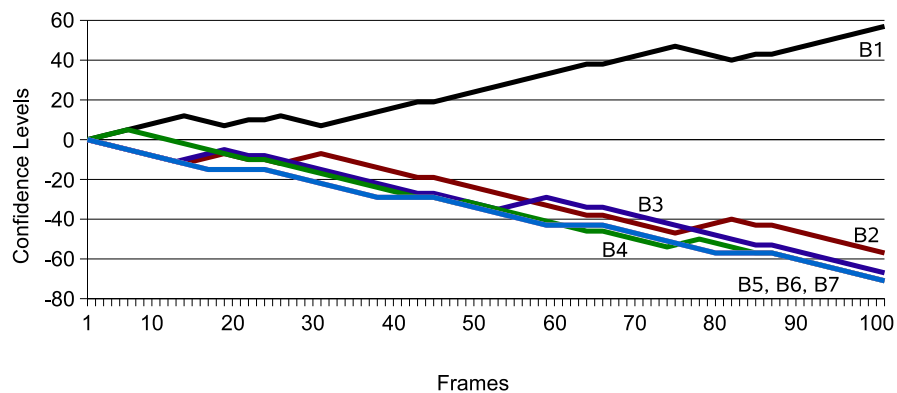


Figure 10: B1 demonstrated. Evolution of the confidences using a content-based approach

Demonstrated Behaviour	No Attention		RR-HCAW		Content-Based	
	Conf	%	Conf	%	Conf	%
1	65	100	38	58.46	57	87.69
2	63	100	50	79.37	59	93.65
3	43	100	35	81.4	35	81.4
4	61	100	60	98.36	59	96.72
5	63	100	61	96.83	61	96.83
6	39	100	32	82.05	36	92.31
7	45	100	40	88.89	45	100
Average	54.14	100	45.14	83.62	50.29	92.66

Table 1: Confidence for each behaviour for each condition as a percentage of the confidence reached with no attention present

shown for all experimental conditions. Percentages are calculated by making a relative comparison between each confidence value and the corresponding one from the version of the architecture without the attention mechanism. The results show that with the content-based control activated, the confidence levels of each of the behaviours are closer to those when no attentional control is present than the ones of the RR-HCAW condition, the best performing non-content-based attentional control strategy (Demiris and Khadhour, 2006). These results indicate that content-based control can maintain the benefits brought by an attention mechanism, but at the same time with less impact on the reliability of the predictions.

6.5 Introducing saccades

The purpose of this subsection is to introduce experiments that quantify the effects of saccades in content-based control. Saccades in general are known to be very demanding, because of the movement required to perform them, during which, no new data can be collected, hence resulting in the possibility of less computations. As a result of this, the allocation of resources becomes even more valuable, and must be allocated correctly in order to still achieve high confidence values.

The result of this would freeze the allocation of resources for *SaccadeCost* number of frames (which simulates the time it takes for the camera to do the

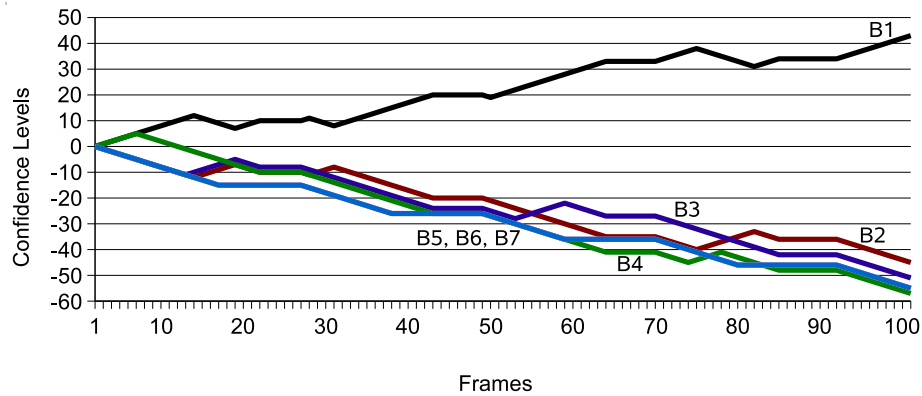


Figure 11: B1 demonstrated. Evolution of the confidences using a content-based approach for scenarios requiring saccades

saccade, during which, all processes must be frozen).

Where possible, objects A and B were placed outside the visual scene to incorporate saccades. Throughout the experiments, only a simulation of the saccade, with its associated cost value (*SaccadeCost*) based on the distance of the object from the attention box was implemented, instead of an actual physical saccade.

As representative examples of the results that were obtained from the experiments, figures 11 and 12 gives the evolution of the confidences of all the behaviours, whilst behaviour 1 and 3 was being demonstrated respectively, using the added *content-based* control.

Table 2 gives a summary of the performance of the attention mechanism for all the other behaviours. In all conditions, all behaviours were recognised correctly. Results of the final confidence value of the demonstrated action are shown for all experimental conditions. Percentages are calculated by making a relative comparison between each confidence value and the corresponding one from the plain version of the architecture without the attention mechanism.

Looking at the graph in figure 11 more closely, it can be seen that there are periods of inactivity, where no computations are taking place, these occur about every 20 frames or so. This represents the saccade cost, which the model

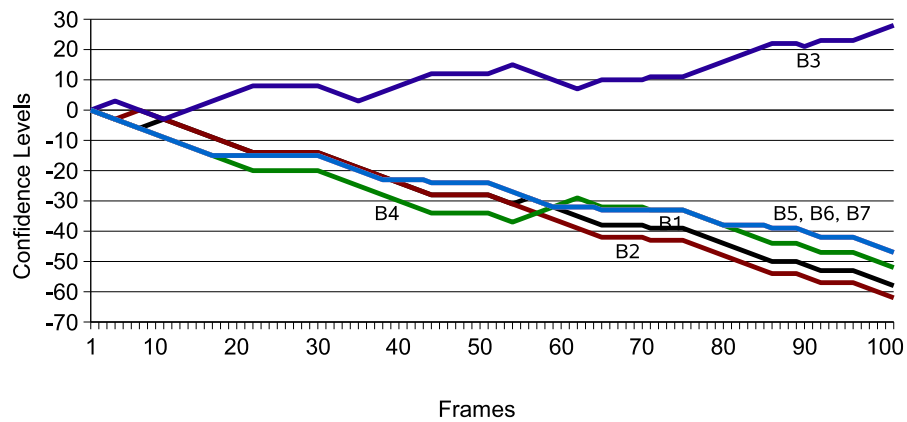


Figure 12: B3 demonstrated. Evolution of the confidences using a content-based approach for scenarios requiring saccades

Demonstrated Behaviour	No Attention		Content-Based	
	Conf	%	Conf	%
1	65	100.0	43	66.2
2	63	100.0	38	60.3
3	43	100.0	28	65.1
4	61	100.0	45	73.8
5	63	100.0	55	87.3
6	39	100.0	36	92.3
7	45	100.0	45	100.0
Average	54.1	100.0	41.4	77.9

Table 2: Confidence for each behaviour for each condition as a percentage of the confidence reached with no attention present for scenarios requiring saccades

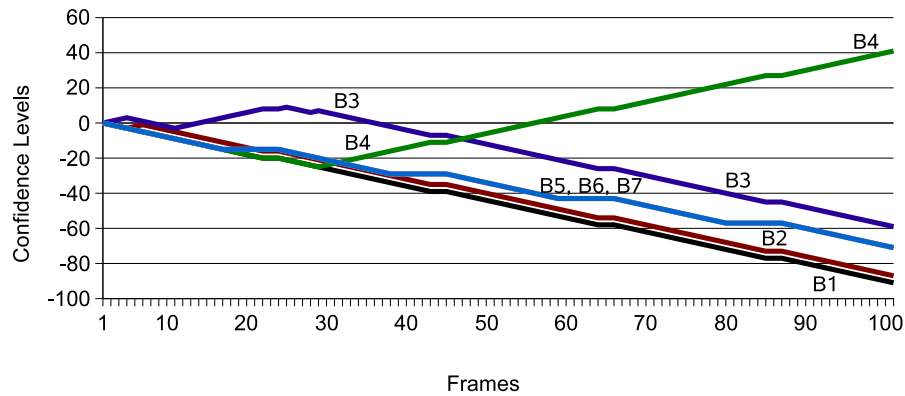


Figure 13: In this experiment, the demonstrator started by demonstrating B3 by moving his hand to the left, and then intentionally changing the direction to move it to the right, hence demonstrating B4

chooses to go through, sacrificing computations, in order to attend to the static objects that fall outside the scene, refreshing their static locations, which were designed to expire after 20 frames in these experiments.

Despite the significant loss in the computations, it can be seen from table 2 that all the behaviours were still identified correctly, with a good average confidence value of 77.85% for the final winning behaviours.

The attention model with the added content-based control is powerful enough to deal with saccades, causing a loss in recognition rate of only 15% (since the recognition rate in the previous section *without* saccades was 92.66%)

6.6 Experiments to demonstrate recovery

The purpose of this section is to demonstrate that the new added content-based control will enable good tracking of all the behaviours, such that it can recover by recognising the correct demonstrated behaviour in case of an error occurring at the beginning.

In order to achieve this, an experiment was designed where the demonstrator starts off by demonstrating B3 and then quickly alternates to demonstrate B4. Figure 13 shows the outcome of this experiment.

The attention model is able to recover and correctly recognise that it was

mainly B4 which was demonstrated, rather than B3.

Since the model managed to recover from this worse case scenario, where the eventual winning behaviour was scoring negatively initially, the model should therefore have no problem either to deal with easier scenarios, where the eventual winner scores positively at the beginning. One such possibility can be to deal with two behaviours that are similar in nature at the beginning, and only differ towards the end. In this case, both behaviours would score positively at the beginning, and therefore recovery will be even easier than what was demonstrated in this experiment.

The reason why the added content-based control has an advantage over the previous attention model that did not have it, is because it allows the computation of more than one behaviour at a time, keeping track of all the behaviours as much as it can. Whereas previously, this was not the case, and only the winning behaviour was being updated, hence recovery would not be as efficient.

7 Discussion

By allocating the limited computational resources according to a *content-based* approach, that is based on behaviour needs/requests, one can see further significant performance gains.

The higher confidence values for the winning behaviours, and the improved way in allocating the resources allowed the attention model to be pushed to work with more demanding situations, such as those requiring saccade generations. It was shown through experiments that the attention model still managed to correctly identify winning behaviours in these situations with a high average experimental confidence value of 78%.

Furthermore, using this approach produced graphs with shapes that no longer just clearly identified the winning behaviour, as was previously achieved, but they also better matched the shape of the original graphs of the perfect

scenario with no attention mechanism at all.

This means that most of the information about the other behaviours are retained, which has also resulted in better performance under more noisy conditions, as well as assisting, in some rare circumstances, to correctly identify the demonstrated behaviour, even if it didn't have the highest confidence value to start with.

Overall therefore, the addition of the content-based control to the attention model has improved the attention model in three useful ways:

- It generally improved the recognition results for the winning behaviours.
- It is therefore better equipped at handling tougher situations such as saccades.
- It kept good track of other behaviours too, minimising ambiguity in scene understanding, hence also allowing a later recovery for a demonstrated behaviour that did not start out as the winning behaviour.

The problem we have addressed, which we have termed utilization of resources, can be seen a question of sampling how coarse can your sampling become while retaining accuracy of the results, compared to perfect sampling (here represented by the no-attention condition as we explained above). In the RR-HCAW condition, given that only one inverse model is granted its request, we obtain $1/N$ samples per time unit, where N is the number of inverse models putting forward a request, saving $N-1/N$ sampling requests per time unit; it might the case that our hardware setup can actually handle N requests per time unit; however we cannot assume that this is always the case, particularly in embedded systems. Even if the computational resources are sufficient (and in this case they are), when you introduce non-sharable resources (e.g. Pan and tilt camera) there will be cases (for e.g. any situation where the action observed takes place over an area larger than your field of view, necessitating saccades)) where perfect sampling will not be possible.

The sampling in the content-based approach remains the same – only one request is serviced, but we now look at the requests at the content level and rank them accordingly; the key point here is that if the request we serviced is made by more than one behavior, we distribute the results to all such behaviors, which can only improve the results (as we see above) by bringing them closer to the theoretically optimal but not always achievable situation (no attention).

8 Conclusions

Attention control is an important aspect of understanding human actions. In our work, we have employed a motor theory approach to the control of attention, putting forward HAMMER as a principled method for calculating the goal-directed importance of visual information. In this paper we have demonstrated that by examining the content of the requests of multiple hypotheses, and calculating its utility, reliability and cost, we can further optimise the sensor usage of the robotic assistant during the observation of human actions. This will enable robotic assistants to respond to human actions rapidly and efficiently.

References

- Dearden, A. and Demiris, Y. (2005). Learning forward models for robotics. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1440–1445.
- Demiris, J. (1999). *Movement imitation mechanisms in robots and humans*. PhD thesis, University of Edinburgh, Scotland, UK.
- Demiris, J. and Hayes, G. M. (1999). Active and passive routes to imitation. In Dautenhahn, K. and Nehaniv, C., editors, *Proceedings of the AISB Symposium on Imitation in Animals and Artifacts*, pages 81–87.
- Demiris, Y. and Hayes, G. (2002). Imitation as a dual route process featuring predictive and learning components: a biologically-plausible computational

- model. In Dautenhahn, K. and Nehaniv, C., editors, *Imitation in Animals and Artifacts*, chapter 13, pages 327–361. MIT Press.
- Demiris, Y. and Johnson, M. (2003). Distributed, prediction perception of actions: a biologically inspired architecture for imitation and learning. *Connection Science*, 15(4):231–243.
- Demiris, Y. and Johnson, M. (2007). Simulation theory of understanding others: A robotics perspective. In Dautenhahn, K. and Nehaniv, C., editors, *Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*, pages 89–102. Cambridge University Press.
- Demiris, Y. and Khadhour, B. (2006). Hierarchical, attentive multiple models for execution and recognition. *Robotics and Autonomous Systems*, 54:361–369.
- Hesslow, G. (2002). Conscious thought as simulation of behaviour and perception. *Trends in Cognitive Sciences*, 6(6):242–247.
- Itti, L. and Koch, C. (2000). A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision Research*, 40(10-12):1489–1506.
- Jeannerod, M. (2001). Neural simulation of actions: a unifying mechanism for motor cognition. *NeuroImage*, 14:103–109.
- Johnson, M. and Demiris, Y. (2004). Abstraction in recognition to solve the correspondence problem for robot imitation. In *Proceedings of TAROS*, pages 63–70, Essex.
- Karniel, A. (2002). Three creatures named forward model. *Neural Networks*, 15:305–307.
- Narendra, K. S. and Balakrishnan, J. (1997). Adaptive control using multiple models. *IEEE Transactions on Automatic Control*, 42(2):171–187.
- Schaal, S., Ijspeert, A., and Billard, A. (2003). Computational approaches to motor learning by imitation. *Phil. Trans. R. Soc London B*, 358:537–547.

- Steels, L. and Baillie, J. (2003). Shared grounding of event descriptions by autonomous robots. *Robotics and Autonomous Systems*, 43:163–173.
- Treue, S. and Trujillo, J. C. M. (1999). Feature-based attention influences motion processing gain in macaque visual cortex. *Nature*, 399(575-579).
- Tsotsos, J. (2001). Motion understanding: Task-directed attention and representations that link perception with action. *International Journal of Computer Vision*, 45(3):265–280.
- Wolfe, J. M. (1994). Visual search in continuous, naturalistic stimuli. *Vision Research*, 34:1187–1195.
- Wolpert, D. M. and Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Networks*, 11:1317–1329.