Preprint version; final version available at www.sciencedirect.com
Robotics and Autonomous Systems (2006), vol: 54(5), pp: 361-369
DOI: 10.1016/j.robot.2006.02.003

Hierarchical Attentive Multiple Models for Execution and Recognition of Actions

Yiannis Demiris *, Bassam Khadhouri

Biologically Inspired Autonomous Robots Team (BioART),
Department of Electrical and Electronic Engineering
Imperial College London
South Kensington Campus, SW7 2BT, London, UK

Abstract

According to the motor theories of perception, the motor systems of an observer are actively involved in the perception of actions when these are performed by a demonstrator. In this paper we review our computational architecture, HAMMER (Hierarchical Attentive Multiple Models for Execution and Recognition), where the motor control systems of a robot are organised in a hierarchical, distributed manner, and can be used in the dual role of (a) competitively selecting and executing an action, and (b) perceiving it when perfomed by a demonstrator. We subsequently demonstrate that such arrangement can provide a principled method for the top-down control of attention during action perception, resulting in significant performance gains. We assess these performance gains under a variety of resource allocation strategies.

Key words: Imitation, attention, hierarchical structures, action recognition

1 Introduction

Following the increased interest in mechanisms that will endow robots with the capability to imitate human action, several computational architectures have been proposed to match visual information from observing a demonstrator to motor plans that would achieve the corresponding action for the observer [1–3]. Internal models of the motor systems of the observer and the demonstrator

* Corresponding author.
Email address: y.demiris@imperial.ac.uk (Yiannis Demiris).
URL: http://www.iis.ee.ic.ac.uk/yiannis (Yiannis Demiris).

Preprint version; final version available at www.sciencedirect.com
Robotics and Autonomous Systems (2006), vol: 54(5), pp: 361-369
DOI: 10.1016/j.robot.2006.02.003

and their capabilities have been frequently suggested as useful tools aiding in this matching (e.g. [4], for a review, see [5]). The particular method we will employ in this paper relies on the concept of motor simulation. Mental simulation theories of cognitive function [6], 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 [7]. With the discovery of the mirror system, first in monkeys [8] and subsequently in humans [9], the idea that motor systems can be used for the perception of others' actions became increasingly popular [10]. In our previous work [2], we have performed a computational investigation of the mirror system using robotic devices as models, which resulted in plausible explanations and testable predictions regarding its behaviour, and in particular its sensitivity to the velocity profile of the demonstrator (for a review see [11]).

In this paper, we describe HAMMER, our computational architecture for recognizing and executing actions, utilising hierarchical, attentive, multiple models. After a review of related work, we will review the operation of the building blocks of the architecture. Subsequently we will perform a number of experiments, (a) illustrating that such architecture can provide a principled method for the top-down control of attention during the perception of actions of others, and (b) demonstrating that it results in significant computational performance increases.

2 Background

Equipping robots with the ability to imitate enables them to learn to perform tasks by observing a human demonstrator [12]. In the center of this ability lies a mechanism that matches demonstrated actions with motor actions available to the robot [2,1,5]. Several architectures have been proposed for implementing this mechanism (for reviews see [13,12,5,14]), including a number of proposals utilizing a shared substrate between execution, planning, and recognition of actions [2,5,15]. This methodological shift, compared to other successful approaches to learning by demonstration [16] was inspired by the discovery of the mirror system [17,18], which indicated that, at least in primates, there is indeed a shared neural substrate between the mechanisms of action execution and those of action recognition. Apart from being compatible with the motor theories of perception, from an engineering perspective this approach is also attractive since it allows reuse of subsystems in multiple roles.

An important issue that remains unresolved is where the attention of the observer should be focused when a demonstrator performs an action. Whilst there is little agreement on an universal definition of attention [19], from an

Preprint version; final version available at www.sciencedirect.com
Robotics and Autonomous Systems (2006), vol: 54(5), pp: 361-369
DOI: 10.1016/j.robot.2006.02.003

engineering point of view it can be defined as a mechanism for allocating the limited perceptual and motor resources 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 scence, 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 [20]. However, it is well known from human psychophysical experiments that top-down information can also influence bottom-up processing (e.g. [21,22]). Wolfe [21] 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 even 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. We will utilize our previous work implementing the motor theory of perception [2] for deriving a principled mechanism for the goal-directed control of attention during action perception, which we consider the key contribution of this paper.

3 HAMMER

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

- The basic building block involves a pair of inverse and forward models in the dual role of either executing or perceiving an action [2].
- These building blocks are arranged in a hierarchical, parallel manner [4].
- 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.

3.1 Building blocks

HAMMER makes extensive use of the concepts of inverse and forward models [23–25]. An inverse model (akin to the concepts of a controller, behavior, or

Preprint version; final version available at www.sciencedirect.com
Robotics and Autonomous Systems (2006), vol: 54(5), pp: 361-369
DOI: 10.1016/j.robot.2006.02.003

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). Related to this concept is that of a forward model of a controlled system: 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.

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 "believes" are necessary to achieve or maintain these implicit or explicit target goal(s). The forward model provides an estimate 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 (an example of this would be achieving different movement speeds [2]).

If HAMMER is 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.

3.2 Distribution and hierarchy

Multiple pairs of inverse and forward models can operate in parallel [2]. 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 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

Preprint version; final version available at www.sciencedirect.com
Robotics and Autonomous Systems (2006), vol: 54(5), pp: 361-369
DOI: 10.1016/j.robot.2006.02.003

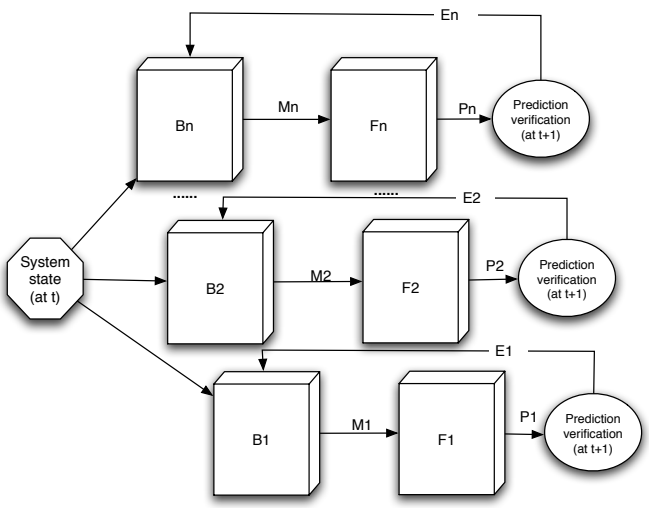

Fig. 1. The basic architecture, showing multiple inverse models (B1 to Bn) receiving the system state, suggesting motor commands (M1 to Mn), with which the corresponding forward models (F1 to Fn) form predictions regarding the system's next state (P1 to Pn); these predictions are verified at the next time state, resulting in a set of error signals (E1 to En)

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 [2], and robotic platforms [4,26] and has offered plausible explanations and testable predictions regarding the behaviour of biological imitation mechanisms in humans and monkeys (review in [11]). More recently we have designed and implemented a hierarchical extension [4] to this arrangement: primitive inverse models are combined to form higher more complex sequences, with the eventual goal of achieving increasingly more abstract inverse models [26], thus helping in dealing with the correspondence problem [27]. The hierarchical aspect of the architecture has been described in detail in [4,26], and since it is not used in the attention experiments that follow, is not elaborated further here. The main point of it is that nodes higher in the hierarchy encode increasingly abstract behavioural aspects, such as goal states, and can be used to simulate the demonstrated action not by following the demonstrated movements, but their effects on the environment instead [26]. We have also conducted experiments to learn forward and inverse models through motor babbling [28], and to integrate the results with the inverse and forward models learned through demonstration [29].

3.3 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,

Preprint version; final version available at www.sciencedirect.com
Robotics and Autonomous Systems (2006), vol: 54(5), pp: 361-369
DOI: 10.1016/j.robot.2006.02.003

the sensory and memory capacities of the observer are limited, so in order to increase the efficiency of the architecture, we do not feed all the state information to 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 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.

4 Experiments

4.1 Experimental setup

We implemented and tested this architecture on an experimental setup involving an ActivMedia Peoplebot; in these experiments the on-board camera was used as the only sensor. A human demonstrator performed an object oriented action (an example is shown in figure 3), and the robot, using HAMMER, attempted to match the action demonstrated (an example is shown in figure 4), 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 2 seconds. In the following sections, we will describe the implementation of the architecture, along with its performance with and without several implementations of the attention mechanism. We will compare its performance against previous implementations of this approach [2,4] to demonstrate the performance improvements that the attention subsystem of HAMMER brings.

Preprint version; final version available at www.sciencedirect.com
Robotics and Autonomous Systems (2006), vol: 54(5), pp: 361-369
DOI: 10.1016/j.robot.2006.02.003

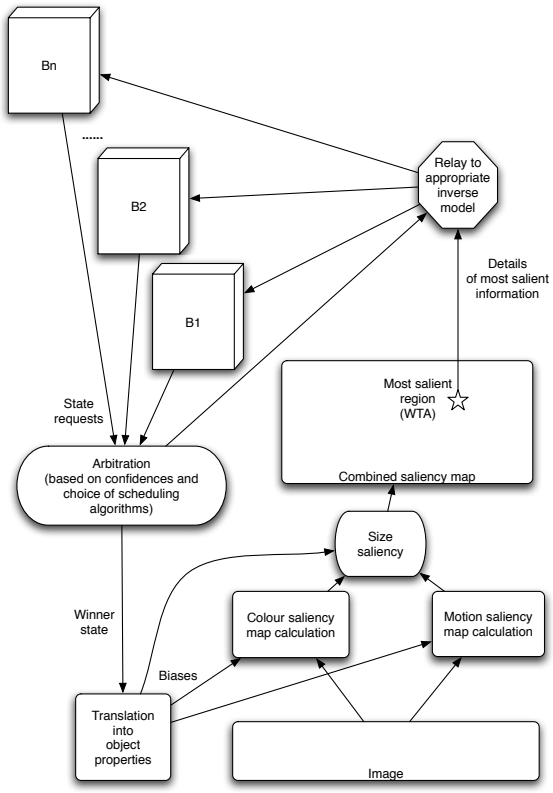

Fig. 2. The complete architecture, incorporating the attention systems; forward models have been omitted (but are shown in figure 1); B1 to Bn are the available inverse models, and the arbitration block has to decide which of their requests to satisfy.

Fig. 3. The experimental setup involves an ActivMedia Peoplebot observing a human demonstrator acting on objects

4.2 Implementation of HAMMER

4.2.1 From inverse models to biases - top down control

A number of inverse models were implemented including *move effector towards object* (in the experiments below objects included a soda can and an orange),

Preprint version; final version available at www.sciencedirect.com
Robotics and Autonomous Systems (2006), vol: 54(5), pp: 361-369
DOI: 10.1016/j.robot.2006.02.003

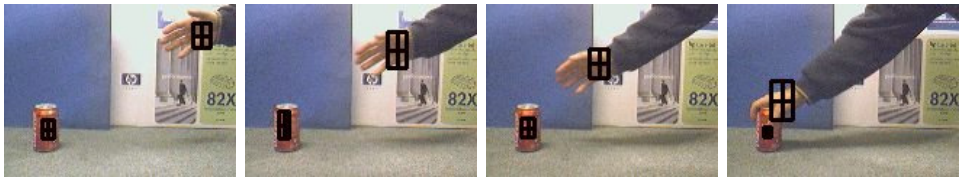

Fig. 4. Representative frames from a video sequence involving a human reaching for an object

pick object, *drop object*, and *move away from object* using the ARIA library of primitives provided with the ActivMedia Peoplebot, similarly to our previous experiments with these robots [4,26]. For each of the inverse models simple directed graphs with conditions that needed to be satisfied (for example, to pick an object, the robot's effector must move towards the object's location until that location is reached) were made available, and were used to monitor the progress of the inverse model. In total, eight inverse models were used (two instances of each of the above four inverse models, one for each object). We used separate inverse models for each object merely in order to increase the number of inverse models available given the limited motor capabilities of the Peoplebot; this way the effects of the architecture and the different test conditions are clearer. The inverse models could be parametrized with the object type rather than keeping a separate model for each object. Forward models were hand-coded for each of the inverse models, using kinematic rules to output a qualitative prediction of the next state of the system for each of these inverse models. For example, given the current position and speed of the hand and with a motor command to move the hand to a certain direction, the predicted next state would be "closer" or "further away" from an object.

The list of state requests of each of the inverse models is passed to the arbitration module of the attention mechanism (essentially the part of the architecture exerting the top down control) along with the current confidence of each of the inverse models. The arbitration module (figure 2) selects which inverse model will be awarded the attention of the robot. We have implemented two different selection algorithms:

- The first one is based on the "Round-Robin" scheduling algorithm [30], whereby a first-in first-out circular linked list is formed with all the requests, and equal resources are given to each. In the implementation of these particular experiments one frame is given to each.
- The second one is priority-based, which gives all the resources to the state requests of the inverse model with the highest confidence.

4.2.2 From biases to inverse models

Once the arbitration process is complete, and an inverse model has been selected, the state vector that this inverse model is requesting gets translated

Preprint version; final version available at www.sciencedirect.com
Robotics and Autonomous Systems (2006), vol: 54(5), pp: 361-369
DOI: 10.1016/j.robot.2006.02.003

into a set of object properties, including colour, motion and size information. The corresponding properties, including hue and saturation histograms for each of the objects (an orange, a soda can, and a human hand) used in our experiments were pre-calculated and saved in the robot's memory. These properties act as biases in the calculation of saliency maps [21,22], which are combined to form a composite saliency map (figure 2). The image location with the highest saliency is extracted and this information is sent to the corresponding inverse model that asked for it. 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:

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

where N is the number of times the inverse model has been rewarded in the past. If an inverse model has not been given the state information it requested, it is not processed in this frame, and its confidence remains constant. We adopted this update function, among others we tried (for example, without the N), since it reflects the desired behaviour of rewarding or punishing an inverse model more as the demonstration progresses: in plain terms, if the behaviour has been rewarded a lot in the past, and continues to get it right, it deserves to get more reward; if an inverse model has been rewarded a lot in the past and begins to get it wrong, it deserves to get a lot of negative reward for taking up too many resources. This update function proves capable of good performance, as described next.

4.3 Results

In these experiments, we captured eight video sequences of a demonstrator performing different tasks in front of the robot to ensure fair comparison between the different conditions, and performed 40 experiments on these videos under different combinations of scheduling algorithms. Figures 6 to 9 give representative results with respect to the performance of the architecture under five different experimental conditions (figure 5 serves as a legend for the other figures, and shows the assigned numbers (B1 to B8) for the different inverse models):

Preprint version; final version available at www.sciencedirect.com
Robotics and Autonomous Systems (2006), vol: 54(5), pp: 361-369
DOI: 10.1016/j.robot.2006.02.003

- No attention mechanism - at each frame all inverse models receive the state information they require, and all compute and update their confidences. This condition is the same used in previous experiments [2,4], and will serve as the reference condition for comparing the performance of the attention subsystem.
- Round-Robin implementation: a circular linked list is created and all requests are served in a first-in first-out fashion (one frame per request).
- HCAW (Highest Confidence Always Wins) condition, in which the inverse model with the current highest confidence is always given the state it is requesting.
- RR-HCAW condition, which combines conditions two and three above. Initially all inverse models 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 inverse models with positive confidences), the arbitration mechanism switches to HCAW.
- RR-HCAW-A condition, which repeats the last condition but performs an initialization step that removes from the candidate inverse models the ones that are not applicable – for example if an object is not present, the corresponding inverse models for handling it are not activated.

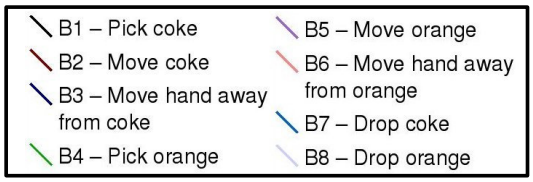

Fig. 5. The inverse models used in the experiments with their corresponding labels

Figure 6 gives the evolution of the confidences of all the inverse models when no attention mechanism is employed. The demonstrator is picking up an orange and the corresponding inverse model (B4) correctly receives the highest confidence score, with the "picking up the can" action (B1) receiving the second highest given its similarity to B4 for the initial part of the trajectory. Figure 7

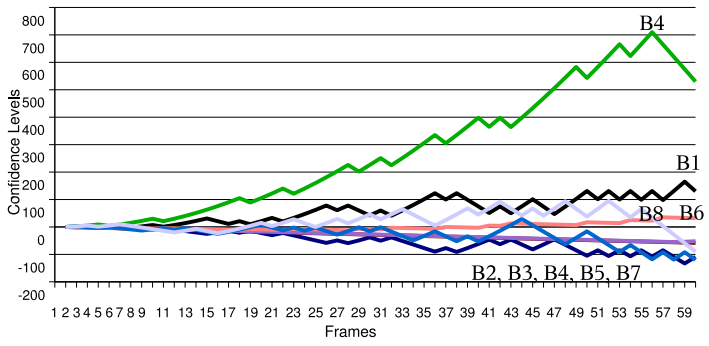

Fig. 6. Evolution of the confidences when no attention mechanism is employed

gives the evolution of the confidences of all the inverse models when attention

Preprint version; final version available at www.sciencedirect.com
Robotics and Autonomous Systems (2006), vol: 54(5), pp: 361-369
DOI: 10.1016/j.robot.2006.02.003

is given to each of the inverse models in an equal-share manner. The corresponding inverse model (B4) again correctly receives the highest confidence score, but the separability of B4 from B1 is now significantly smaller since the inverse models are not as frequently updated as in the first experimental condition. The corresponding computational savings however are significant (assuming equal computational resources are required for each inverse model, this is of the order of $\frac{n-1}{n}$ (87.5% in this case), with n being the number of inverse models - the arbitration mechanism's overheads are negligible), since under this condition (and the following three), only one of the eight inverse models gets processed at each frame. This point is further elaborated in the discussion section. Figure 8 gives the evolution of the confidences of all the

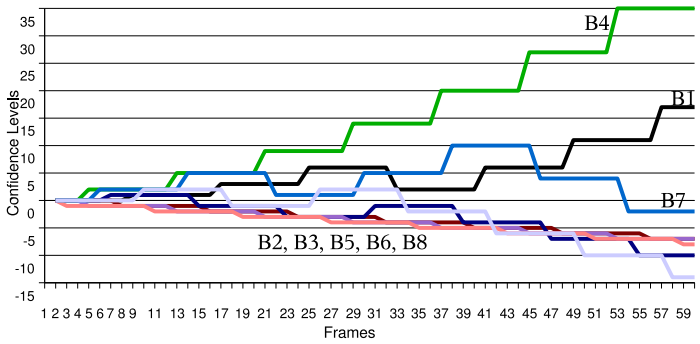

Fig. 7. Evolution of the confidence values with a Round Robin implementation

inverse models when attention is given only to the inverse model with the highest confidence. The corresponding inverse model (B4) still correctly receives the highest confidence score, and the separability of B4 from B1 (and all the rest) is now clearer. However this condition is more susceptible to noise with respect to the RR condition and the overall confidence portait of the inverse models is noisier than before. Figure 9 gives the evolution of the confidences

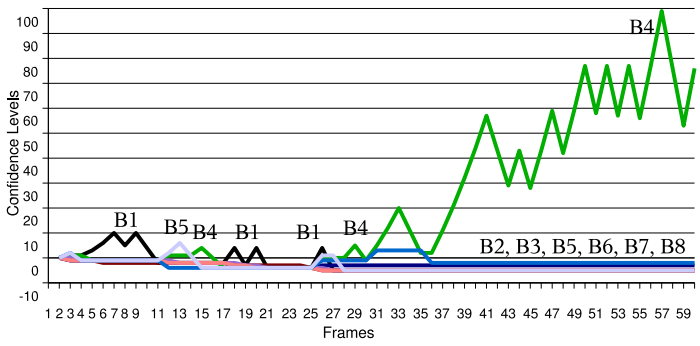

Fig. 8. Evolution of the confidence values with a Highest Confidence Always Wins (HCAW) implementation

of all the inverse models when attention is given to an inverse model using a combination of the two previous conditions. Initally all inverse models are receiving equal treatment (RR), but when the confidence of one of them in-

Preprint version; final version available at www.sciencedirect.com
Robotics and Autonomous Systems (2006), vol: 54(5), pp: 361-369
DOI: 10.1016/j.robot.2006.02.003

creases to more than 50% of the average of all inverse models with a positive score, the selection algorithm switches to one that favours the one with the highest confidence. This condition combines the benefits of the two previous conditions: it displays clear separability between the winner inverse model and the rest, while not displaying the initial noisy profile of the HCAW condition. This condition essentially amounts to an attention strategy of "playing it safe" at first, and then focusing the system's resources to the winner once a strong candidate is found.

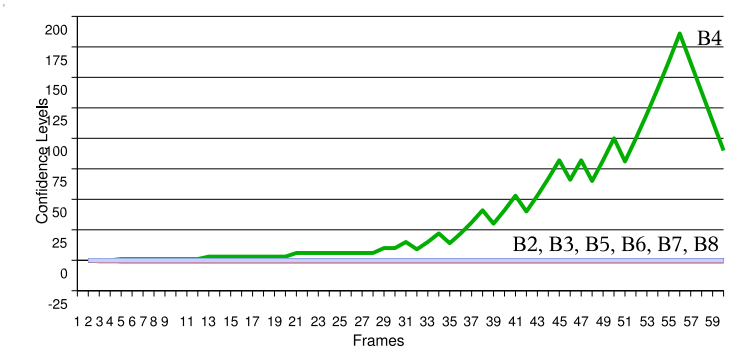

Fig. 9. Evolution of confidence values using the RR-HCAW condition

Finally, we repeat the last experiment, removing the soda can from the setup. Figure 10 shows the evolution of confidences of the inverse models with the initialisation step; the inverse models that have to deal with the soda can are not considered in the arbitration process, and (as a matter of convention) are constantly receiving negative confidence reinforcement of -1. This boosts the separability even further since there are fewer inverse models competing for attention. Figure 11 gives a summary of the performance of the attention

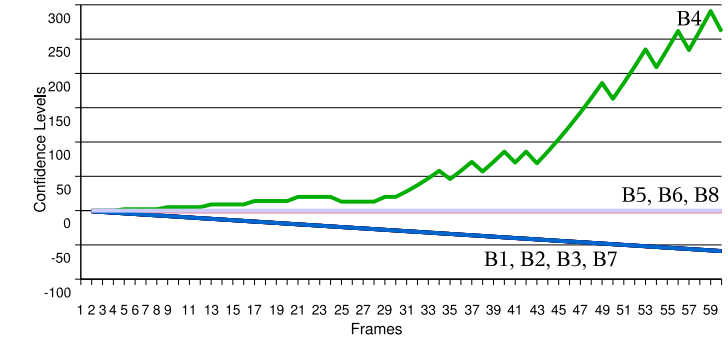

Fig. 10. Evolution of the confidence values using the RR-HCAW-A condition

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 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.

Preprint version; final version available at www.sciencedirect.com
Robotics and Autonomous Systems (2006), vol: 54(5), pp: 361-369
DOI: 10.1016/j.robot.2006.02.003

Demonstrated	No Attention		RR		HCAW		RR-HCAW		RR-HCAW-A	
Behaviour	Conf	%	Conf	%	Conf	%	Conf	%	Conf	%
1	503	100.0	35	7.0	84	16.7	145	28.8	365	72.6
2	721	100.0	35	4.9	402	55.8	414	57.4	491	68.1
3	681	100.0	28	4.1	107	15.7	366	53.7	524	76.9
4	527	100.0	35	6.6	77	14.6	91	17.3	263	49.9
5	395	100.0	35	8.9	113	28.6	200	50.6	279	70.6
6	642	100.0	44	6.9	412	64.2	260	40.5	452	70.4
7	739	100.0	35	4.7	175	23.7	238	32.2	447	60.5
8	628	100.0	28	4.5	227	36.1	337	53.7	452	72.0
Mean	604.5	100.0	34.4	5.9	199.6	31.9	256.4	41.8	409.1	67.6

Fig. 11. Confidence for each behaviour for each condition as a percentage of the confidence reached with no attention present

5 Discussion

The key contribution of this paper is the introduction of a novel way of exerting top-down control to the attention mechanism, using a competitive, parallel implementation of the simulation theory of mind. Our experiments demonstrate that the addition of an attention mechanism which arbitrates among the state requests of multiple inverse models results in a significant amount of computational savings, without significant loss of separability between the inverse models. This is particularly true with a combination of a conservative approach (initially equal-share, RR) switched to a more aggressive one (HCAW) when a strong candidate is found. As expected, taking into consideration the environmental context (condition RR-HCAW-A) further improves the results by removing inapplicable inverse models. We expect that the combination of our system with a joint attention mechanism [31] which extracts the gaze of the demonstrator, and through it, attempts to infer the focus of attention of the demonstrator will further reduce the number of applicable inverse models. We intend to pursue this in future work.

The savings in our current implementation, although significant, are only computational given that saccades within our system take place within the image and do not involve camera movements. However, should the situation require camera movements to locate the state information requested, these savings would increase given the additional cost associated with each camera movement. The fact that at the motor level, resources will always need to be shared, makes our architecture useful even for implementation on parallel machines that could afford to process, from a computational perspective, more than one inverse model at a time.

One point worth discussing is the "resource-slice" that the winning inverse model is given at each iteration by the attention mechanism. Although in this particular set of experiments we have used one single frame, and we perform one iteration of the arbitration mechanism at each frame, that does not imply it is the only solution. We have found that for some of the experimental conditions, it has the undesirable effect of linking the numbers of inverse models with the frame rate: for example, if the number of inverse models in-

Preprint version; final version available at www.sciencedirect.com
Robotics and Autonomous Systems (2006), vol: 54(5), pp: 361-369
DOI: 10.1016/j.robot.2006.02.003

creases and the attention system is working in the RR condition, the rate that each inverse model will be processed (and thus, the accuracy of its predictions) decreases. On the positive note, using the applicability initialization (RR-HCAW-A condition), the number of inverse models that are involved is reduced, reducing this problem, but it would be interesting to experiment with different resource-slices, and methods of allocating resources. Our current implementation utilizes an arbitration mechanism that takes into consideration the requests blindly, without considering what they represent; for example, more than one inverse model can be asking for the same information, the state of the arm. We are currently considering a further optimization method that allocates resources according to the current popularity of state requests across the inverse models. A "working memory" mechanism can also maintain a number of recent states and immediately return them when they are required. Furthermore, path planning algorithms can also be utilized so future requests are optimized topographically (for example, if three inverse models are requesting the state of the head, arm and leg of a human, these will be served in that order, irrespective as to whether the order that the requests were made, or their priorities) to minimize lengthy saccades.

6 Conclusions

We reviewed our approach to the development of architectures that incorporate distributed, hierarchical networks of inverse and forward models, and described how HAMMER can be used to perceive a demonstrated action. We performed computational experiments demonstrating how different resource allocation strategies in the control of attention can influence the performance of HAMMER.

The novelty of our approach lies in the idea that the features that the observer will choose to attend are the ones that (s)he would have to control if (s)he was in the same situation and would have to perform the same action. This is compatible with recent biological evidence [32] on the use of action plans in action recognition. This stems naturally from the simulation approach to action perception [11], which has been the main inspiration underlying our work, and provides a timely opportunity to study the interplay between the two important topics of attention and action perception.

References

[1] A. Billard, Learning motor skills by imitation: a biologically inspired robotic model, Cybernetics and Systems 32 (2000) 155–193.

Preprint version; final version available at www.sciencedirect.com
Robotics and Autonomous Systems (2006), vol: 54(5), pp: 361-369
DOI: 10.1016/j.robot.2006.02.003

[2] Y. Demiris, G. Hayes, Imitation as a dual route process featuring predictive and learning components: a biologically-plausible computational model, in: K. Dautenhahn, C. Nehaniv (Eds.), Imitation in Animals and Artifacts, MIT Press, 2002, Ch. 13, pp. 327–361.

[3] A. Alissandrakis, C. L. Nehaniv, K. Dautenhahn, Imitating with ALICE: Learning to imitate corresponding actions across dissimilar embodiments, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 32 (4) (2002) 482–496.

[4] Y. Demiris, M. Johnson, Distributed, prediction perception of actions: a biologically inspired architecture for imitation and learning, Connection Science 15(4) (2003) 231–243.

[5] S. Schaal, A. Ijspeert, A. Billard, Computational approaches to motor learning by imitation, Phil. Trans. R. Soc London B 358 (2003) 537–547.

[6] G. Hesslow, Conscious thought as simulation of behaviour and perception, Trends in Cognitive Sciences 6 (6) (2002) 242–247.

[7] M. Jeannerod, Neural simulation of actions: a unifying mechanism for motor cognition, NeuroImage 14 (2001) 103–109.

[8] G. Rizzolatti, L. Fadiga, V. Gallese, L. Fogassi, Premotor cortex and the recognition of motor actions, Cognitive Brain Research 3 (1996) 131–141.

[9] J. Grezes, J. L. Armony, J. Rowe, R. E. Passingham, Activations related to "mirror" and "canonical" neurones in the human brain: an fmri study, NeuroImage 18 (2003) 928–937.

[10] V. Gallese, A. Goldman, Mirror neurons and the simulation theory of mind-reading, Trends in Cognitive Sciences 2 (12) (1998) 493–501.

[11] Y. Demiris, M. Johnson, Simulation theory of understanding others: A robotics perspective, in: K. Dautenhahn, C. Nehaniv (Eds.), Imitation and Social Learning in Robots, Humans and Animals: Behavioural, Social and Communicative Dimensions, Cambridge University Press, 2005, p. to appear.

[12] S. Schaal, Is imitation learning the way to humanoid robots?, Trends in cognitive sciences 3 (6) (1999) 233–242.

[13] C. Breazeal, B. Scassellati, Robots that imitate humans, Trends in Cognitive Sciences 6 (11) (2002) 481–487.

[14] K. Dautenhahn, C. Nehaniv, Imitation in Animals and Artifacts, MIT Press, Cambridge, MA, USA, 2002.

[15] D. M. Wolpert, K. Doya, M. Kawato, A unifying computational framework for motor control and social interaction, Phil. Trans. of the Royal Society of London B 358 (2003) 593–602.

[16] Y. Kuniyoshi, M. Inaba, H. Inoue, Learning by watching: Extracting reusable task knowledge from visual observation of human performance, IEEE Transactions on Robotics and Automation 10 (6) (1994) 799–822.

Preprint version; final version available at www.sciencedirect.com
Robotics and Autonomous Systems (2006), vol: 54(5), pp: 361-369
DOI: 10.1016/j.robot.2006.02.003

[17] G. di Pellegrino, L. Fadiga, L. Fogassi, V. Gallese, G. Rizzolatti, Understanding motor events: a neurophysiological study, Experimental Brain Research 91 (1992) 176–180.

[18] J. Decety, J. Grezes, N. Costes, D. Perani, M. Jeannerod, E. Procyk, F. Grassi, F. Fazio, Brain activity during observation of actions: Influence of action content and subject's strategy, Brain 120 (1997) 1763–1777.

[19] J. Tsotsos, Motion understanding: Task-directed attention and representations that link perception with action, International Journal of Computer Vision 45 (3) (2001) 265–280.

[20] L. Itti, C. Koch, A saliency-based search mechanism for overt and covert shifts of visual attention, Vision Research 40 (10-12) (2000) 1489–1506.

[21] J. M. Wolfe, Visual search in continuous, naturalistic stimuli, Vision Research 34 (1994) 1187–1195.

[22] S. Treue, J. C. M. Trujillo, Feature-based attention influences motion processing gain in macaque visual cortex, Nature 399 (575-579).

[23] K. S. Narendra, J. Balakrishnan, Adaptive control using multiple models, IEEE Transactions on Automatic Control 42 (2) (1997) 171–187.

[24] A. Karniel, Three creatures named forward model, Neural Networks 15 (2002) 305–307.

[25] D. M. Wolpert, M. Kawato, Multiple paired forward and inverse models for motor control, Neural Networks 11 (1998) 1317–1329.

[26] M. Johnson, Y. Demiris, Abstraction in recognition to solve the correspondence problem for robot imitation, in: Proceedings of TAROS, Essex, 2004, pp. 63–70.

[27] C. Nehaniv, K. Dautenhahn, The correspondence problem, in: K. Dautenhahn, C. Nehaniv (Eds.), Imitation in animals and artifacts, MIT Press, 2002, Ch. 2, pp. 41–61.

[28] A. Dearden, Y. Demiris, Learning forward models for robotics, in: Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), 2005, pp. 1440–1445.

[29] Y. Demiris, A. Dearden, From motor babbling to hierarchical learning by imitation: a robot developmental pathway, in: Proceedings of the 5th International Workshop on Epigenetic Robotics, 2005, pp. 31–37.

[30] W. Stallings, Operating Systems: Internals and Design Principles, Prentice Hall, 2000.

[31] Y. Nagai, K. Hosoda, A. Morita, M. Asada, A constructive model for the development of joint attention, Connection Science 15 (4) (2003) 211–229.

[32] J. R. Flanagan, R. S. Johansson, Action plans used in action observation, Nature 424 (2003) 769–771.