Compound Effects of Top-down and Bottom-up Influences on Visual Attention During Action Recognition

Bassam Khadhouri and Yiannis Demiris
Department of Electrical and Electronic Engineering
Imperial College London
Exhibition Road, London SW7 2BT
Email: {bassam.khadhouri, y.demiris}@imperial.ac.uk

Abstract

The limited visual and computational resources available during the perception of a human action makes a visual attention mechanism essential. In this paper we propose an attention mechanism that combines the saliency of top-down (or goal-directed) elements, based on multiple hypotheses about the demonstrated action, with the saliency of bottom-up (or stimulus-driven) components. Furthermore, we use the bottom-up part to initialise the top-down, hence resulting in a selection of the behaviours that rightly require the limited computational resources. This attention mechanism is then combined with an action understanding model and implemented on a robot, where we examine its performance during the observation of object-directed human actions.

1 Introduction

In an attempt to arrive at a definition for *attention*, Tsotsos, in his review paper [Tsotsos, 2001], arrives at the following proposal: "*Attention is a set of strategies that attempts to reduce the computational cost of the search processes inherent in visual perception*". Our work aims at producing a model of *visual attention* for *dynamic scenes*, emphasising the importance of *top-down knowledge* in directing the attention during action recognition.

After introducing the two different (bottom-up and top-down) elements of attention, we will proceed to review a model of action understanding [Demiris and Johnson, 2003] that will make use of our attention model, to correctly allocate the limited resources available to it. Subsequently, we will proceed to describe how the multiple hypotheses generated by our model *while the human demonstration is unfolding* can feed top-down signals to the attention mechanism to direct attention to the important aspects of the demonstration.

Furthermore, we propose a method for *initialising* the top-down part, using the saliency of the bottom-up part in our visual attention mechanism. We have implemented our model on an ActivMedia Robot, running experiments observing a human acting upon various objects. These results will be presented and discussed, not only in terms of whether our attention model allocates the resources correctly, but also to see if it results in a faster recognition of the correct behaviour being demonstrated in the action understanding model.

2 Background

Work in cognitive science suggests that the control inputs to the attention mechanism can be divided into two categories: stimulus-driven (or "bottom-up") and goal-directed (or "top-down") [Van Essen *et al.*, 1991]. A number of bottom-up attention models follow Treisman's Feature Integration theory [Treisman and Gelade, 1980] by calculating the saliency for different low-level features of the object, e.g. colour, texture or movement. A winner-take-all approach is then used to decide on the most salient part of the scene (as in [Koch and Ullman, 1985], [Itti *et al.*, 1998], [Breazeal and Scassellati, 1999]). Top-down, on the other hand, covers the goal-directed factors, essentially the task-dependent part of the visual attention model. Wolfe [Wolfe and Gancarz, 1996] produced a biologically inspired Guided Search Model that controls the bottom-up features that are relevant to the current task by a top-down mechanism, *through varying the weighting of the feature maps*. However, it is not clear what the task relevant features are, particularly in the case of action recognition.

Our attention mechanism is inspired by Wolfe's model, and integrates bottom-up elements with a top-down mechanism using *behaviours*[1] and *forward models* that can guide the robot's attention according to the current task. A forward model is a function that, given the current state of the system and a control command to be applied on it as given by the behaviour, outputs the predicted next state. Our top-down part of the attention model, when observing a demonstration, will make a prediction of the next state for a number of different possible behaviours, producing a confidence value for each, based on the observed accuracy of the prediction. These confidence levels are important values that can be thought of as *saliencies* for the top-down mechanism of our attention model, hence producing a principled method of quantifying the top-down part of the attention mechanism.

In the experimental section, we will demonstrate that this attention mechanism improves performance on action understanding mechanisms that use multiple behaviours and for-

[1] Also known as controllers or inverse models in the control literature

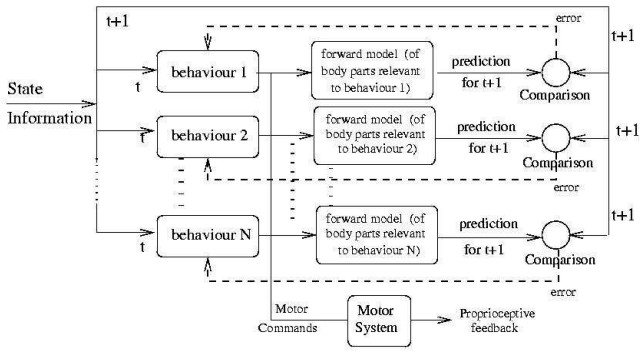


Figure 1: Action understanding model [Demiris and Johnson, 2003].

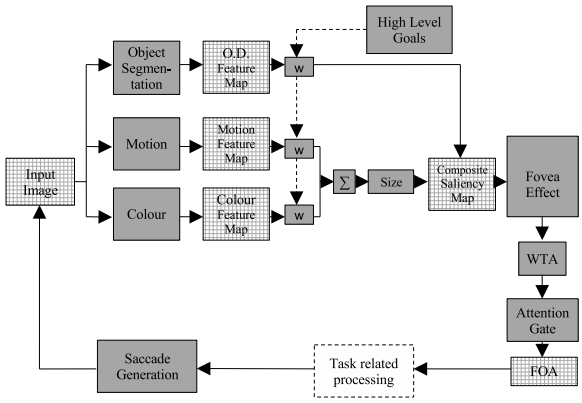


Figure 2: Our Bottom-up model of Visual Attention. The output is the Focus of Attention (FOA)

ward models such as [Demiris and Johnson, 2003]. This will be done in two ways: first by cutting down the number of computational cycles required for identifying the correct behaviour and by directing the limited computational resources to the relevant parts of the human demonstration, instead of the whole scene. Secondly, by using the saliency from the bottom-up part of our visual attention mechanism to initialise the top-down part, enabling it to select only the relevant behaviours to the demonstrated action, instead of activating and running all of them.

3 Action Understanding Model

Demiris's action understanding model [Demiris and Johnson, 2003], shown in figure 1, identifies the correct behaviour that is being observed by using forward models for a number of behaviours. By predicting what will happen next, and comparing it with what actually does happen next (from the behaviour that is being demonstrated), confidence levels are generated for each of the predicted behaviours. From these, a winner is selected by picking the predicted behaviour with the highest confidence level.

The attention mechanism we propose in this paper can be used to cut computational costs on this action understanding model [Demiris and Johnson, 2003]. It would be far too computationally expensive to direct the attention of the observer towards all the parts of the demonstration to satisfy all the possible behaviours [Tsotsos, 1989]. Hence, the attention mechanism is used to restrict this, giving only one of the behaviours at a time the information it requires. Using this attention mechanism, we managed to cut down substantially on the computational costs, yet it was also achieved without affecting the quality of the system to the extent of producing wrong outcomes.

The success of our model will be demonstrated through comparisons of the new results with the results from the original action understanding model for a number of different behaviours. If, after having cut down on all the computational costs using our visual attention model, the final behaviour chosen in each situation with the previous model [Demiris and Johnson, 2003] remains the same, then our attention model will be deemed to have succeeded in its task.

4 The Attention Mechanism

In this section we will describe the design of our attention mechanism, and the integration of both bottom-up and top-down elements.

4.1 Bottom-up

We have implemented a bottom-up model that is mainly based on Wolfe's biologically inspired Guided Search 3.0 model of human visual attention and visual search [Wolfe and Gancarz, 1996]. This model uses Treisman's Feature Integration theory [Treisman and Gelade, 1980] to construct a model of human visual attention. In this model, low-level filters are applied to various visual stimuli in order to produce individual feature maps in which high values indicate areas of interest.

All of the individual feature maps are weighted and then summed into a single activation map. Attention is guided to peaks in the activation map, because these represent the most salient areas in the scene. In our model, top-down task information can influence the bottom-up feature maps by changing the activation map through the modifying of the weights that are applied before the summation. Our model is shown in figure 2. There are certain features that can make objects salient. For example, brightly coloured objects are a most typical example, or if they are moving in a way that can attract attention, e.g a sudden, irregular and fast movement. Each of the bottom-up blocks in the model represents a certain feature that contributes towards the calculation of the saliency of an object. Our implementation focuses on three bottom-up blocks that are feature detectors. These are: *Motion*, *Colour* and the *Size* the object occupies in the image, which not only accounts for the actual size of an object, but also for the distance of the object from the camera, both of which are important in grabbing one's attention.

The remaining blocks in the model are:

- Fovea-effect – this models the decrease in resolution away from the centre of the image, because our eyes' resolution decreases dramatically with the distance from the fovea [Farid *et al.*, 2002].

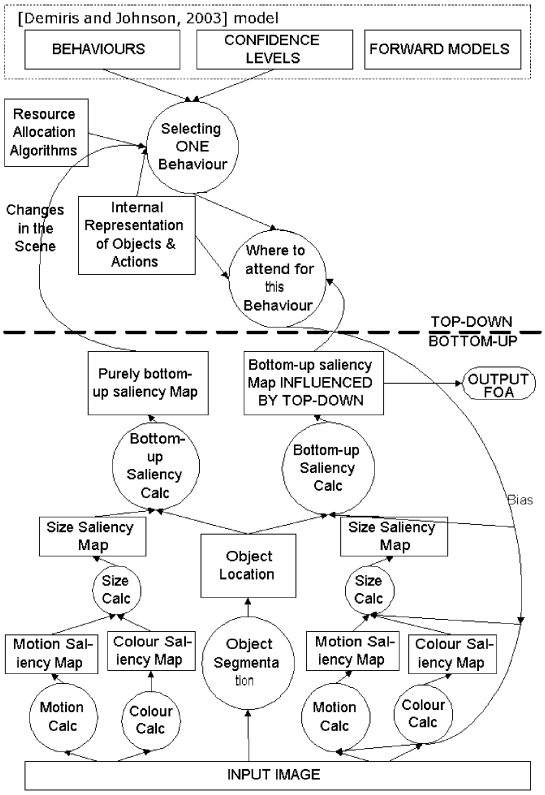


Figure 3: The architecture of the top-down part of the model and how it integrates with the bottom-up part

- Winner takes all (WTA) selection as in Itti's model [Itti *et al.*, 1998].
- An attention gate mechanism as in Wolfe's model [Wolfe and Gancarz, 1996] which keeps a record of the two most salient areas in the scene.
- A saccade generation system as in Wolfe's model [Wolfe and Gancarz, 1996].

4.2 Top-down

Figure 3 shows our complete visual attention model which includes the top-down part. This figure specifically illustrates how the top-down information could influence the bottom-up part of the model, and vice versa.

Our Top-down part of the model receives the list of behaviours from the action understanding model, described in the previous section, together with their confidence levels and their forward models. It must select only one behaviour out of the many behaviours to attend to at any given point in time. The block labelled "Selecting ONE Behaviour" in figure 3 has five inputs:

- Behaviours – A list of hypotheses (potential behaviours that explain the demonstration) is passed in, one of which must be selected.
- Confidence Levels – The current confidence levels for each behaviour.
- Resource Allocation Algorithms – we have performed experiments with different resource allocation algorithms [Stallings, 2000] that can be employed to decide on how to distribute resources between the behaviours.
- Internal Representations of objects and actions – this block gives information about objects and how they move, and interact with other objects etc.
- Purely bottom-up saliency map – this is the saliency map representing the most salient object in the scene.

From the above five inputs, one behaviour must be chosen. The block labelled "Where to attend for this Behaviour?" in figure 3 has three inputs:

- Output from "Selecting ONE Behaviour" – the winner behaviour that is selected from the previous stage is passed on.
- Internal Representations of objects and actions – this block gives information about objects and how they move, and interact with other objects etc.
- Bottom-Up Saliency map influenced by top-down – this is to give current information on where the attention of the model is.

The output of this block influences all the bottom-up calculation blocks in order to direct the attention of our model in such a way that it serves the current behaviour.

5 Experimental Setup

We implemented our model on an ActivMedia Peoplebot robot, equipped with a pan-tilt-zoom camera, as well as a two degrees of freedom gripper, sonar and infrared sensors. Only the camera was used in the following sets of experiments, and the saccade generation module was switched off as it was not needed.

For these experiments, three objects were chosen: A hand, a coke can, and an orange. Eight behaviours were then defined:

- Behaviour 1 - Pick coke can
- Behaviour 2 - Move coke can
- Behaviour 3 - Move hand away from coke can
- Behaviour 4 - Pick orange
- Behaviour 5 - Move orange
- Behaviour 6 - Move hand away from orange
- Behaviour 7 - Drop coke can
- Behaviour 8 - Drop orange

Each of these behaviours has a corresponding forward model. Figure 4 shows the arrangement for Behaviour 1.

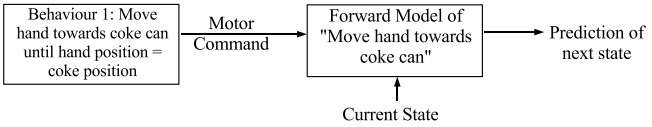


Figure 4: Behaviour1 - Picking a coke can

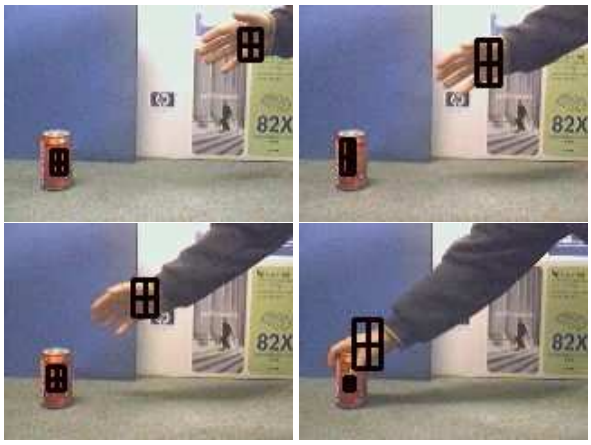


Figure 5: Images to show bottom-up block processing a scene of a hand picking a coke can

All the other behaviours are implemented in the same way. Forward models were hand coded using kinematic rules to output the prediction of the next state. The output from the forward model, which is a prediction of the actual next state, is compared with the next state. Based on this comparison, a confidence level is generated for this behaviour by either rewarding it with one confidence point if the prediction is successful, or otherwise punishing it by subtracting one confidence point.

Figure 5 shows an input example of what the robot sees when a behaviour is carried out (in this case, it is the demonstration of a hand picking a coke can). A background was chosen where these object's colours were minimally present. These are only snapshots of some frames.

The bottom-up block detects and tracks the presence of the coke can, the orange and/or the hand in the scene, depending on what the top-down part of the attention model requires. The output of this bottom-up block are the corresponding locations of where the hand, the coke can and/or the orange are in the scene. This information is then passed to the top-down part of the model for intelligent processing. The CAMShift algorithm [Bradski, 1998] was used to assist in doing this. We used a hue and saturation histogram back-projection of camera images taken at a pixel resolution of 160×120 and at 30 frames per second. The corresponding histograms of the three objects used in our experiments were pre-saved into the robot's memory, and used during the experiments as a simple method of object recognition.

Four different implementations were experimented with each of the eight behaviours:

- A pure implementation of the action understanding model without our attention model. Hence there was no layer of intelligence to cut down on computational costs, i.e. each behaviour gets to carry out all the computations it requires at *each* frame.
- Our Attention model is added to the action understanding model using a "round robin" scheduling algorithm (equal time sharing) [Stallings, 2000] to select between the behaviours. Therefore, each behaviour is processed every eighth frame, since there are eight behaviours in total.
- Our Attention model is added to the action understanding model using the strategy of "highest confidence level always wins", which means the behaviour with the previous highest confidence level gets the next computation.
- Our Attention model is added to the action understanding model using a combination of the "round robin" and the "highest confidence level always wins" strategies to select between the behaviours.

Finally, we also ran another set of experiments by adding to these implementations initialisation for the top-down part using bottom-up saliencies.

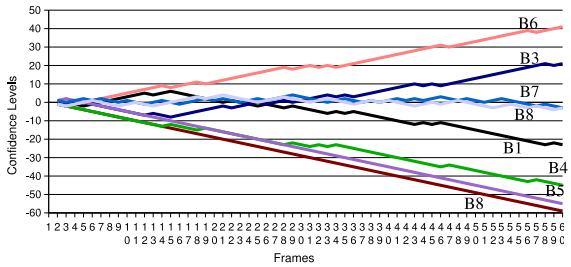
6 Experimental Results

We used 26 different videos, performing 130 experiments, using the different scheduling algorithms above, while varying some of the model's parameters. The results for behaviour 6 are shown in figure 6 as an example of a typical output to demonstrate how our system can work on top of the action understanding model, cutting down its computational costs by up to 87.5% (because every behaviour is now only being processed *once in every 8 frames*), and still producing the correct results to determine which behaviour is being demonstrated. But more importantly, it directs the limited computational resources to the relevant areas in the scene, instead of analysing the whole scene.

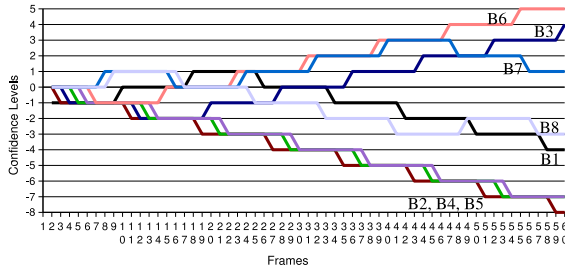
In addition to this, the above results were substantially improved by adding initialisation to the top-down part using the bottom-up saliencies to our attention model . Therefore, in the case of a scene where the orange does not exist, as shown in figure 5, our bottom-up block would detect the coke can and the hand as being salient objects. Using previously saved colour histograms, our system recognises that these salient objects are a coke can and a hand. This result is passed to the top-down part of our attention model, which in turn will only select the behaviours that are involved with these objects, as opposed to previously selecting every behaviour that exists in the database.

Results from behaviour 1, where no orange was present in the scene, are shown in figure 7 as an example for this initialisation process of only selecting the relevant behaviours using the bottom-up saliencies. The results are compared to the previous implementation without this initialisation. It can be seen that this initialisation process speeds up the correct recognition of the correct behaviour. Furthermore, it will also serve in allowing scalability to be added to our model.

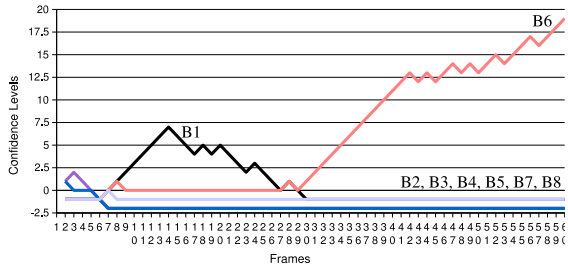
As can be seen from figure 6, our attention model not only still gives the correct results for the recognition of behaviour 6, but it does it with the saving of up to 87.5% of the total computational costs. Furthermore, it returns better results in recognising the correct behaviour by isolating it from the other wrong behaviours, due to the focusing on the correct areas of the scene only, instead of the entire image. These successful results were also seen for all of the other seven behaviours that were tested.



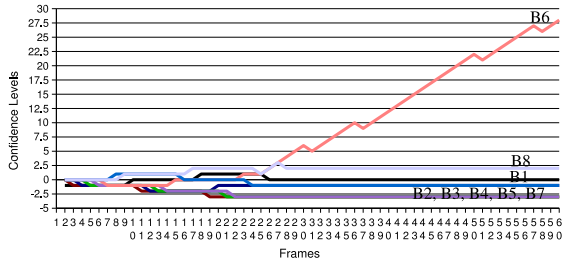
Each behaviour is processed every frame



Behaviours are processed according to Round Robin scheduling algorithm



Behaviours are processed according to the highest confidence level



Behaviours are processed according to Round Robin at the start, and then according to the highest confidence level

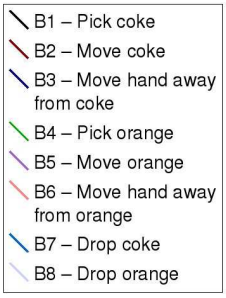
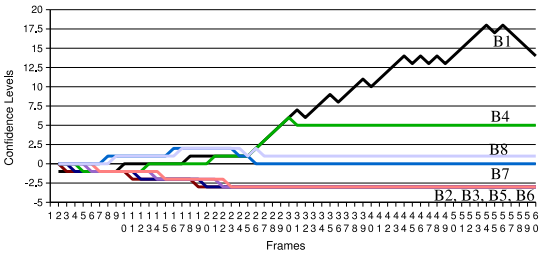
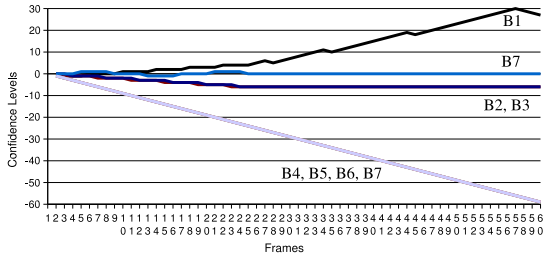


Figure 6: Behaviour 6 - Move hand away from orange



Behaviours are processed *without* the bottom-up block initialising the top-down



Behaviours are processed *with* the bottom-up block initialising the top-down

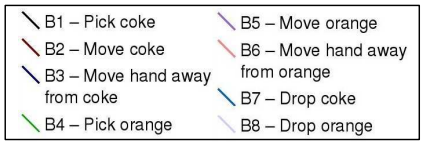


Figure 7: Behaviour 1 - Pick coke can

Behaviour 6 in figure 6 shows that when a "round robin" scheduling algorithm is applied, the correct behaviour is still picked, but it ends up with a lower confidence value that is quite close to the other resulting confidence values of the other behaviours. The reduction in computational costs using this algorithm has resulted in a decrease on the separability of the behaviours. This is because there are n frames in the demonstrated scene (in these experiments, n is 60), and each behaviour is only processed once every m frames, where m is the number of behaviours (in these experiments, m is 8) meaning a total number of n/m computations per behaviour (which is 7 in these experiments, hence, only a maximum score of 7 for the winning behaviour). Behaviour 6 in this case scores 5 out of 7, still enough to make it the winning behaviour, but much lower than the pure implementation of the action understanding model without the use of any attention mechanism.

When "highest confidence level always wins" is used, the opposite effect to "round robin" can be seen: attention acts like an accelerator to the winning behaviour once it recognises who is the winner, suppressing all the others. The problem with purely using the scheduling algorithm of "highest confidence level always wins", is that it may not always initialise correctly, as can be seen in figure 6, hence taking some time before converging on the correct behaviour.

To alleviate this problem, we used the "round robin" scheduling algorithm as an initialisation step for the "highest confidence level always wins" which then acts as an ac-

celerator for the winning behaviour. Hence, "round robin" is applied only for half the length of the demonstration (30 frames in these experiments). Then the "highest confidence level always wins" is applied as an accelerator to the winning behaviour. As can be seen from figure 6, it returns accurate and fast results for selecting the winning behaviour. This process may be seen as being equivalent to "playing it safe" at the beginning by looking everywhere, because we are not yet sure what is happening. But once we start to understand the scene better, we are more confident to focus on the relevant parts and ignoring the rest.

Results from figure 7 show significant improvements when using the bottom-up saliencies to initialise the top-down selection of the relevant behaviours. Hence as a result, only the relevant behaviours are picked for processing, instead of picking all the behaviours in the robot's database. In this experiment, the robot has 8 behaviours in its database, 4 of which involve a hand and a coke can, and the other 4 involve a hand and an orange. In the demonstrated video, the orange is not present, hence, all the 4 behaviours involving the orange, are immediately punished and not processed. The remaining 4 behaviours (instead of the total 8) are processed with the "round robin" scheduling algorithm at the beginning. Great improvement can be seen here in finding the correct behaviour sooner, at frame 10 instead of the previous frame 31. This can be thought of as enabling our model to recognise the correct behaviour being demonstrated easier and quicker in a less complicated scene with fewer objects.

7 Conclusion

Our attention model utilises both, resource scheduling algorithms, and initialisation of the top-down mechanism to cut down on the computational costs. We have found that a combination of the "round robin" and the "highest confidence level always wins" strategies, together with using the bottom-up saliencies for initialising the top-down selection of the relevant behaviours to the demonstrated scene, worked very well.

The computational costs needed to run our attention model are justified since the action-understanding model is aimed at having a large number of behaviours. As the number of behaviours increases therefore, the resultant search space makes the model indispensable. This is especially because the savings on the computational costs will also increase by $(n-1)/n$ for n behaviours.

We are working towards further optimisation of our model by considering optimal path algorithms. For example, if the sequence of planned allocations involves looking at the head, feet, and arm of a human demonstrator, the algorithm will try to minimise the saccade required and rearrange the allocations to accommodate that.

Optimisation based on predicting future requests will further enhance the performance of our model, and will be our next step in our investigation between attention and action understanding, with the ultimate goal of having robots that will efficiently understand our human actions.

Acknowledgments

This research was funded by the Donal W Morphy Trust Scholarship and by the UK Engineering and Physical Sciences Research Council and The Royal Society. The authors would like to thank the BioART team for their support and assistance, especially Matthew Johnson who assisted in the implementation of the camshift tracker.

References

[Bradski, 1998] G. R. Bradski. Computer vision face tracking for use in a perceptual user interface. *Intel Technology Journal*, Q2:1–15, 1998.

[Breazeal and Scassellati, 1999] Cynthia Breazeal and Brian Scassellati. A context-dependent attention system for a social robot. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1146–1153, San Francisco, CA, USA, 1999.

[Demiris and Johnson, 2003] Y. Demiris and M. Johnson. Distributed, predictive perception of actions: a biologically inspired robotics architecture for imitation and learning. *Connection Science Journal*, 5(4):231–243, 2003.

[Farid *et al.*, 2002] M. M. Farid, F. Kurugollu, and F. D. Murtagh. Adaptive wavelet eye-gaze-based video compression [4877-32]. *Proceedings- Spie the International Society for Optical Engineering*, (4877):255–263, 2002.

[Itti *et al.*, 1998] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *Ieee Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[Koch and Ullman, 1985] C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology*, 4:219–227, 1985.

[Stallings, 2000] William Stallings. *Operating systems : internals and design principles*. Prentice Hall, 4th ed edition, 2000.

[Treisman and Gelade, 1980] A. Treisman and G. Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980.

[Tsotsos, 1989] J. K. Tsotsos. Complexity of perceptual search tasks. In *Proceedings of the 11th Int. Joint Conf. on Artificial Intelligence*, pages 1571–1577, 1989.

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

[Van Essen *et al.*, 1991] D.C. Van Essen, B. Olshausen, C.H. Anderson, and J.L. Gallant. Pattern recognition, attention, and information bottlenecks in the primate visual system. *Proc. SPIE Conf. on Visual Information Processing: From Neurons to Chips*, 1991.

[Wolfe and Gancarz, 1996] J. M. Wolfe and G. Gancarz. Guided search 3.0. In *Basic and Clinical Applications of Vision Science*, pages 189–192. Kluwer Academic Publishers, Dordrecht, 1996.