Attention and Social Situatedness for Skill Acquisition

Yuval Marom and Gillian M. Hayes
Division of Informatics, The University of Edinburgh
5 Forrest Hill, Edinburgh EH1 2QL, UK
{yuvalm, gmh}@dai.ed.ac.uk

Abstract

We present an attention system that models the dynamics that occur in memory in response to stimuli, which includes habituation, novelty detection, and forgetting. We demonstrate how such an attention system can be used as a trigger for learning perception-action mappings. We discuss the value of social situatedness in the form demonstrator-learner interactions, and show results from both simulations and robot-human experiments of a simple wall-following task.

1 Introduction

Attention

In our approach to attention, we are mainly inspired by the *Habituation Hypothesis of Selective Attention* (Cowan, 1988), which claims that rather than attention being a simple filtering mechanism that selects certain inputs and disposes of others, it is a system that continually inspects all its input channels, compares them to descriptions stored in memory, habituates to familiar stimuli, and has the ability to dishabituate if needed.

The role of habituation is to inhibit what is known as the *Orienting Response*, which is a combination of neural, physiological, and behavioural changes that an organism undergoes when a novel or significant stimulus is detected (Sokolov, 1963; Kahneman, 1973). What is interesting is how the orienting response is reinstated (dishabituation), and this can occur due to a number of factors (Balkenius, 2000). The work we report here directly models two of these factors: the passage of time (forgetting), and the presentation of a novel stimulus.

Attention in general involves orienting cognitive resources to target locations likely to improve signal detection (Posner et al., 1980). In most of the computational implementations of attention the target locations are spatial in nature, and usually involve the computation of a saliency map, where spatial locations compete in a winner-take-all manner (for example (Itti and Koch, 2001)).

We are not concerned with spatial attention, *i.e.* orienting to salient spatial locations where a location's saliency is calculated with respect to its neighbouring spatial locations. Rather, we are looking at what could be thought of as temporal attention: orienting to salient perceptual *instances*, where saliency is calculated with respect to previous experiences. We do not deal with spatial selection — our system has to deal with all the perceptual sensors of the robot.

In other words, rather than asking *where* are the interesting locations given a snap-shot of the environment, we are asking *when* should a snap-shot be taken in the first place.

Learning in a social context

We believe that by placing a robot in a social context, one can achieve an implicit transfer of information between an experienced demonstrator (human or robot), and the learner robot. The social context provides a form of joint attention, where the two agents can share similar experiences. There are many phenomena that make up the field of Social Learning, and exact definitions and ideas vary between researchers (for good overviews, see Galef (1988) and Whiten (2000)). The one that describes our work best is *Stimulus Enhancement* (Spence, 1937), where the demonstrator merely acts in ways to increase the chances that the observer perceives given events — no information about goals is transmitted.

There are different forms of interactions possible in this demonstrator-learner scenario. The simplest one is when the learner follows the demonstrator, copying its actions — in this case the demonstrator wants the learner to go through certain sensori-motor experiences relevant to some task. This paper, and most of our work so far, deals with this kind of interaction. However, we believe the social context can also provide external stimulus rewards, which could be used as another factor for dishabituation (Balkenius, 2000), and we are currently investigating this idea (see more on this in Section 4).

Physical and social situatedness

By being situated in a physical environment, a robot learner builds up representations of its perceptual history, habituating to what it sees often; it begins to ignore certain information based on the properties of the particular environment it is in. This governs how it reacts to stimuli, and further — how it bootstraps new sensori-motor skills from existing ones. Being socially situated allows the robot to do the above more efficiently and with more relevance to a particular task, by implicitly utilising the knowledge of another agent in the environment. The particular environment and demonstrator play an important role here: if any of those were to change, the learner's memory and task capabilities would develop differently.

The next section describes a computational implementation of an attention system with habituation. Section 3 presents an example of how attention can be used to trigger learning necessary for the acquisition of new skills. Section 4 briefly discusses the value of the social context, and is followed by a concluding section.

2 Computational Framework for Attention

As mentioned in the previous section, by 'attention' we are referring to the interaction of structures in memory in response to stimuli, rather than one specific mechanism. A computational model should reflect how perceptual information activates stored structures for comparisons; how structures are added, updated, and deleted; and how structures habituate to familiar stimuli, but are able to reinstate the orientation response if required.

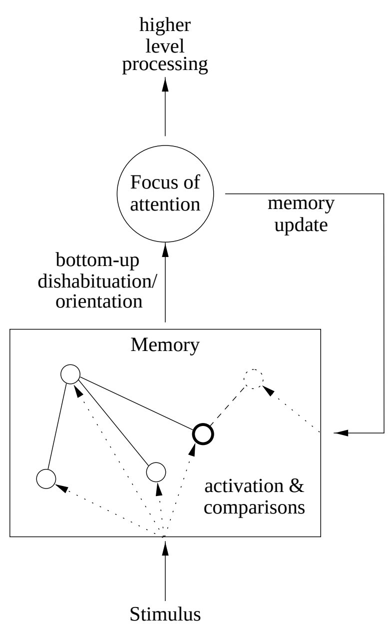

Figure 1: The attention model.

In our model, inspired by Cowan's model of attention (Cowan, 1988) and depicted in Figure 1, the activation and comparisons of structures in memory occur outside of attention, in a passive, automatic manner. When the orienting response is reinstated (due to novelty or forgetting), the information goes

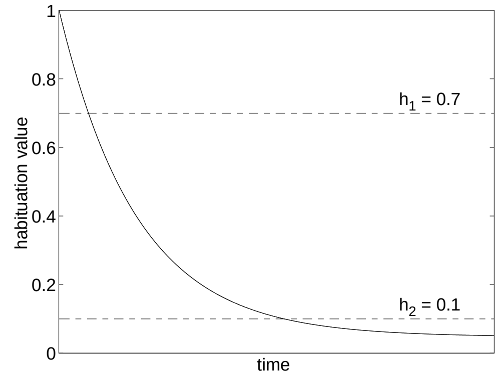

Figure 2: Habituation measure.

through to the focus of attention, which can cause the creation of new structures in memory, and trigger some higher-level processing, such as learning.

Several unsupervised learning approaches are suitable for modelling these kinds of dynamics in memory. We believe the Self Organising Feature Map (SOFM) is an appropriate tool, and it has been proven to be very useful in robotic implementations (Nehmzow, 1999). The SOFM attempts to cover the sensory input space with a network of nodes, and edges connecting neighbouring nodes determined by some distance measure, in our case an Euclidean distance. This has the effect of preserving the topology inherent in the space. We are interested in a variation of the SOFM, where structures (nodes in the network) grow from experience as required, rather than being specified a-priori.

We have adopted and suited to our purposes an algorithm developed by Marsland et al. (2001), which incorporates notions of habituation, novelty detection, and forgetting. This implementation reflects our biologically-inspired view of attention (as described in Section 1) only in the way that nodes of the SOFM react to stimuli; the notion of edges in the SOFM does not explicitly stem from any biological motivations, but rather is an inherent part of the algorithm, necessary for maintaining the relationships between nodes.

Habituation

The main asset of this algorithm is that it keeps a *habituation* measure for each node in the SOFM. This measure gives an indication of familiarity, *i.e.* the frequency of that node's activation, which provides a useful heuristic. Each time a node is active, its habituation value decreases exponentially, as shown in Figure 2, according to:

$$\tau \frac{dy(t)}{dt} = \alpha[y_0 - y(t)] - 1 \qquad (1)$$

where y is the current habituation value, y_0 is the initial habituation value, τ determines the habituation

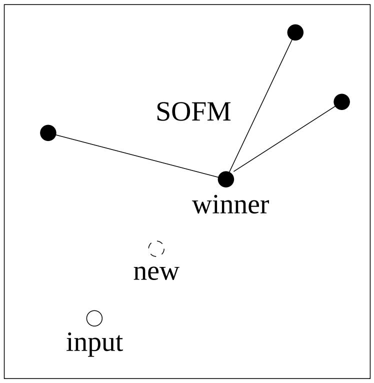

Figure 3: The SOFM network. Each node in the network has a habituation value that decreases whenever that node is active, or it is connected to an active node. A node is active when it matches the current input best. A familiar input moves the nodes towards it, whereas a novel one initiates the creation of a new node. Topology is preserved through the edges connecting the nodes.

rate, and α determines the habituation asymptote. This equation is used by Marsland et al. (2001), and is biologically inspired.

In order to eliminate the need to tune many parameters, we set up a typical scenario, as depicted in Figure 2: habituation decreases from 1 to 0 (y_0 always 1), and always converges on the same asymptote (fix the value of α, in our case to 1.05). The only remaining parameter is the 'time' parameter τ, and will depend on the time-scale of the particular application. Further, in addition to the active node, its neighbours also habituate, but slower, and therefore use a different (higher) value of τ (we have used values in the range 50–600, depending on the application).

For the algorithm presented below, two further parameters are required: a 'minimal' habituation threshold (h_1 in Figure 2), used to determine if a stimulus is completely unfamiliar to a node, and a 'full' habituation threshold (h_2), used to determined if a stimulus is very familiar.

By fixing these thresholds ($h_1 = 0.7$ and $h_2 = 0.1$), the only free parameter is still the time parameter, τ, and we can use it to control how fast a stimulus becomes more and more familiar. We can leave the remaining parameters fixed, and thus easily adapt the algorithm below to different applications.

The algorithm

Below is a general description of the algorithm, see also Figure 3. Note that the nodes and edges of the SOFM form the the memory module of Figure 1.

- For each input, decide which of the existing nodes in the network best matches it, in which case that node 'fires' and is referred to as the 'winning' node.
- Decide whether the input matches the winning node well. This is where novelty detection occurs: a threshold (on the Euclidean distance) is used to judge similarities.
- If the input matches the winning node well, the winning node and its neighbours habituate, and move towards the input by some fraction of the distance between them (update rate).
- If the input doesn't match the winning node well, it is *potentially* novel — the habituation value of the node is used to decide whether it is novel or not, as follows:
  - if the node has only recently been added (its habituation value is higher than h_1), then it is still being positioned in the input space, so we don't add a new node, but rather update it as described above.
  - otherwise, the node has fired a number of times, and has probably settled in the right place of the input space, so a mismatch means novelty is detected, and a new node is needed; we insert it half-way between the input and the winning node (see Figure 3).
- If a node is completely habituated (its habituation value is less than h_2), we 'freeze' the node: the node does not move from where it is, and cannot be deleted. Once a node has been frozen for a specific length of time, we 'un-freeze' it by setting its habituation value to half the starting value, thus introducing 'forgetting'.
- Finally, a brief note about the construction of edges. As well as the best-matching node, the algorithm finds the second best-matching node. These two nodes are then connected with an edge (if not already connected), and this edge has an 'age' value set to 0. The rest of the edges emanating from the winning node have their age increased, and when they get old enough they are deleted; further, any disconnected nodes are deleted. This has the effect of constructing clusters.

To summarise, the system handles attention as follows. Nodes in the network respond and habituate to their respective stimuli. When fully habituated, nodes ignore further stimulation and hence do not get updated. The orienting response is reinstated either due to novelty detection, when a new node is created, or due to forgetting, when a node is dishabituated — in both situations the stimulus is (re-) attended to.

Notice that there are a few more parameters involved in the algorithm (novelty threshold, update rate, full-habituation time, maximum edge-age) in

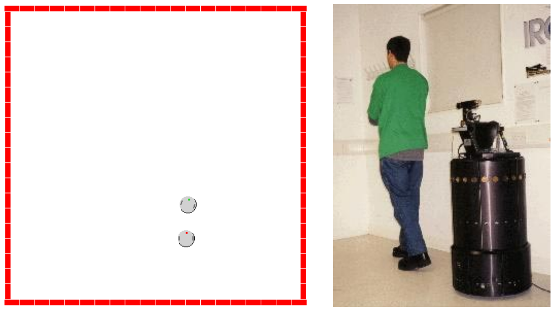

Figure 4: The simulated (left) and physical (right) environments.

addition to those discussed already. Altogether the algorithm is quite heuristic and involves many parameters, some of which arise from the choice to use the SOFM as a learning tool, and others added to explicitly model the characteristics of an attention system.

Some of the parameters are more sensitive (important) and hence interesting, such as the novelty threshold, and the length of full-habituation time, and we regard these as free parameters that have a significant effect on the performance of the algorithm (the latter is tested explicitly in Section 3). We have tried to fix the less sensitive and hence less interesting parameters to values that would work across different implementations, and have done so by experimenting with small toy problems. We have implemented the algorithm on three different implementations, two of which, reported in the next section, are on mobile robots, the third on a simulated humanoid robot (Maistros et al., 2001). For each implementation the free parameters are re-tuned to fit the time-scale and the characteristics of the data for the particular application.

Experimental setup

Throughout this paper we will be looking at experiments involving one task — wall-following, in both a simulated and physical environment. The simulated experiments are performed using a Khepera mobile robot simulator with a learner agent following behind a teacher agent, both using infra-red sensors (see left of Figure 4). The input here comes from 6 sensors around the front of the learner. The physical experiments are performed using a Real World Interface B21 robot, and a human demonstrator; the robot can detect and follow the human using its on-board video camera (see right of Figure 4). The input here comes from 20 sonar sensors around the top of the robot. The size of the physical arena is approximately a 4.8m $\times$ 6.5m square.

In both experiments, the learner uses its built-in following behaviour to keep behind the demonstrator; the demonstrator executes the wall-following task which involves moving parallel to a wall on either side for a fixed length of time, after which an 'interrupt' makes it turn towards the middle of the arena and adopt a wandering behaviour until a wall is found again. Since the learner can sometimes lose the demonstrator (in both experiments), it only inspects its perceptual input when the demonstrator is in sight, that is, when it is in a social context. Otherwise, the attention system would encounter situations not relevant to the task (see Marom and Hayes (2001) for more details). This is a form of stimulus enhancement as described in Section 1.

The algorithm in use

We want to see how well our attention system handles the perceptual data. First we'll show what a full perceptual dataset from a complete run looks like. Since in both types of experiments the dimensionality of the sensor space is quite high (6 and 20), we have used a dimensionality reduction technique called Principal Component Analysis (PCA), for display and analysis purposes. PCA finds the most statistically significant dimensions, called Principal Components, in a multivariate dataset (see Afifi and Clark (1996) for more information on PCA).

As the learner is led through the environment, we save its sensory input at each step into a dataset; the plots in Figure 5 are projections of the final dataset onto the first 2 principal components found by PCA.[1] In both experiments we expect to see 3 main clusters, and they are indeed apparent in the plots: one cluster is the intersection of the 2 apparent lines, which is the area of low (weak) wall-detection, *i.e.* 'no wall', and as we move away from this intersection in either direction, we reach clusters corresponding to 'right wall' and 'left wall' (or vice versa), at different distances from the wall. Note that in the simulation, at very close distances from the wall there are few data points (faint clusters). This is because the robot is usually slightly away from the wall, and this is captured by more data points (darker clusters).

Next we want to see how the attention system handles the perceptual data, noting that it receives them sequentially on-line, not as a complete dataset. Figure 6 shows the construction of the SOFM network in response to incoming stimuli, at 4 arbitrary stages. Note that here too we have had to use PCA to reduce the dimensionality of the SOFM for display purposes; at each stage there is a different SOFM, for which PCA finds different principal components, and so the axes differ (in the physical experiments)

[1] It's important to check how much of the variance is actually accounted for by the first 2 principal components, and this is given by the sum of their eigenvalues. This gives us an indication of how representative a 2-dimensional plot is of the true complexity in the data. In the simulation 2 dimensions account for approximately 95% of the total variance, compared to approximately 65% in the physical experiment. This is to be expected as there are many more physical sensors, and they have much more noise.

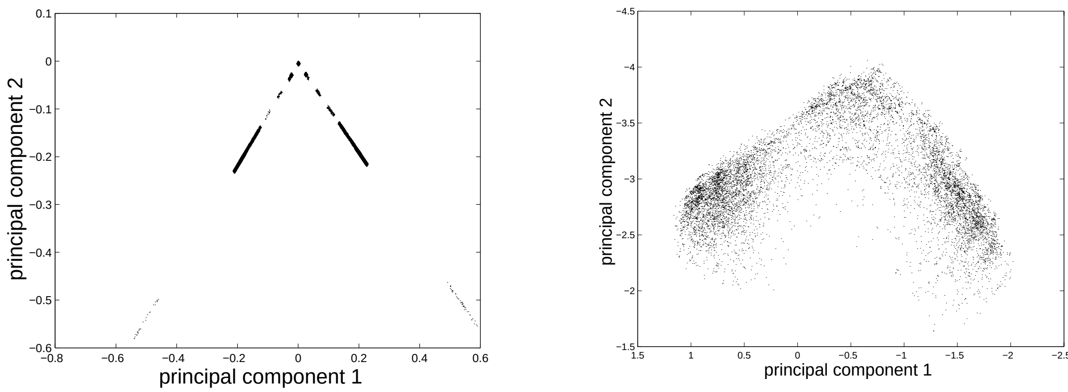

Figure 5: The full perceptual datasets from the simulated (left) and physical (right) experiments, projected for visual purposes onto 2 dimensions determined by PCA.

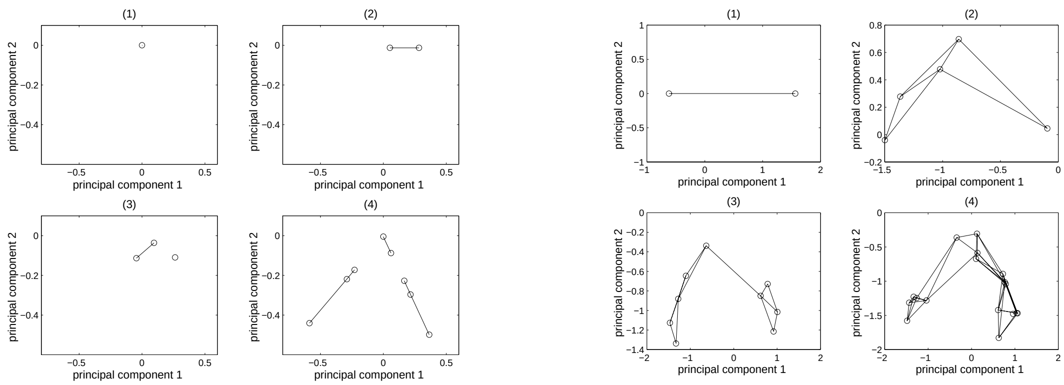

Figure 6: The construction of the SOFM network in the simulated (left) and physical (right) experiments, displayed at 4 arbitrary stages. These are also projection onto 2 dimensions.

— what is important is the structure of the SOFM rather than the axes that PCA decides on.

The SOFM starts with 2 random nodes, and creates new nodes and clusters as it receives the data. In both experiments the SOFM seems to capture the structure inherent in the data, suggesting that our attention system handles the perceptual data as desired.

3 Attention-Triggered Learning

The attention system described above is only responsible for analysing and structuring perceptual information, a problem motivated and explained in (Marom and Hayes, 2001). No learning has yet taken place in terms of the acquisition of new skills. We identify two possible ways to utilise the attention system, for learning perception-action mappings:

1. since the SOFM algorithm produces distinct representations of the perceptual space, one can directly associate (link) them with corresponding motor representations (grouped similarly or otherwise). We have implemented this approach and report it elsewhere (Maistros et al., 2001).

2. use the attention system purely as a trigger, and perform the actual learning on the raw sensori-motor information; while the system is paying attention, pass the information to a separate learning system.

In this paper we will investigate the usefulness of the attention system in the second scenario mentioned above. We can draw inspiration for choosing this approach from dual-process theories of memory, which claim that the process of familiarity detection is distinct from actual storage and recall (O'Reilly et al., 1998). We present here results mainly from learning in the simulated experiment, and briefly mention results from the physical experiment, which we are currently analysing.

The learning of perception-action mappings is achieved using a feed-forward neural network with back-propagation (backprop). The input layer consists of units representing the perceptual sensors (6 units), the hidden layer consists of 2 units, and the output layer consists of 1 unit, which is a preprocessed representation of the motor outputs, corresponding to a tendency to turn[2]. We are going to

[2]computed using a moving average of motor commands as follows: the motor commands are 0, -1, and 1 corresponding to a forward move, right turn, and left turn, respectively; these values are saved into a short moving window on which the average is calculated; the value used in the output neuron is

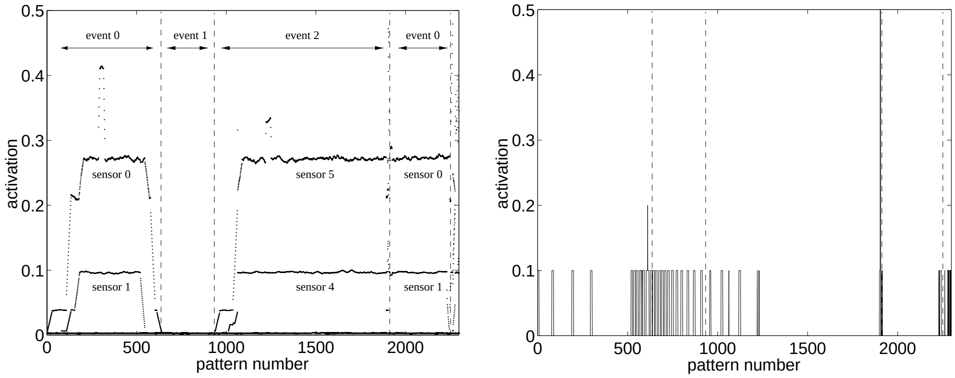

Figure 7: The patterns passed on to the backprop network form learning events. Left: activation of perceptual sensors. Right: activation of pre-processed motor values, representing tendency to turn. Emergent perceptual events are marked by the vertical lines.

train this network to distinguish between moving forward (low turn-tendency) when the robot is parallel to a wall, and moving around randomly with turning (high turn-tendency), when not next to a wall. We are not teaching the robot how to turn towards or away from a wall, and rely on low-level coded behaviours for these purposes.

Whenever the learner is attentive, its raw sensor values and (processed) motor values are used to make up a supervised learning-pattern for the backprop network. The network then performs the usual back-propagation of the error in the output unit, resulting in weight updates.

Perceptual events

We want to see how well the learning system copes with and without attention. We recall that the attention system is attentive whenever a node responds to the input, which occurs when nodes are not fully habituated. We can examine the learning patterns that are passed on to the backprop network, as a result of the attention triggering. For the most simplified case, we do not allow habituated nodes to dishabituate (forget), so the learning network is only exposed to stimulus 'events' once, where by 'event' we mean a sequence of similar stimuli, grouped by one or more nodes (a cluster) of the SOFM.

This situation is depicted in the left plot of Figure 7. It shows the perceptual activations of the learner's 6 sensors as it is led through the environment (0-1 on the left of the robot, 4-5 on the right, and 2-3 in front; see Figure 8), although only the 4 side sensors are active . What is clearly visible is the emergence of the events, and we have superimposed vertical lines and labels to mark them.

Initially the learner is exposed to the wall on the left (event 0); then it is exposed to a new event (event 1), which corresponds to not sensing the wall any-

the absolute value of this average, since we are not interested in direction of turn, only in 'tendency to turn'.

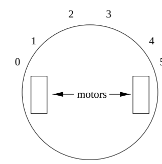

Figure 8: A diagram of the simulated Khepera robot.

where (recall that the demonstrator turns away from the wall at regular intervals and wanders in the middle of the arena); another new event is then encountered (event 2) — the wall being sensed on the right; and finally the first event is sensed again, because the SOFM nodes for this event did not fully habituate the first time; no further attention is given, as all the nodes are fully habituated, and no novel stimuli are encountered.

On the right of Figure 7 we see the pre-processed values from the motors, used to train the backprop network. We use the perceptual event-markings from the left plot to show how motoric events coincide with perceptual ones. Frequent activations correspond to a higher tendency to turn (hence 'wandering'), and this appears to occur when there is no wall stimulation; low frequencies correspond to low tendency to turn, or high tendency to move straight forward, which appears to occur when a wall is sensed on either side (hence 'wall-following'). The two plots in Figure 7 therefore suggest that the backprop network would learn the correct mappings.

Full exposure vs. high selectivity

Figure 7 clearly shows the benefit of attention-triggered learning: the attention system has reduced the exposure of the learning system to 3 'events', or approximately 2400 learning patterns (out of 50000 — the length of a single run). If the learning network can form the correct perception-action mapping, this is of great value.

Unfortunately, this is not always the case: sin-

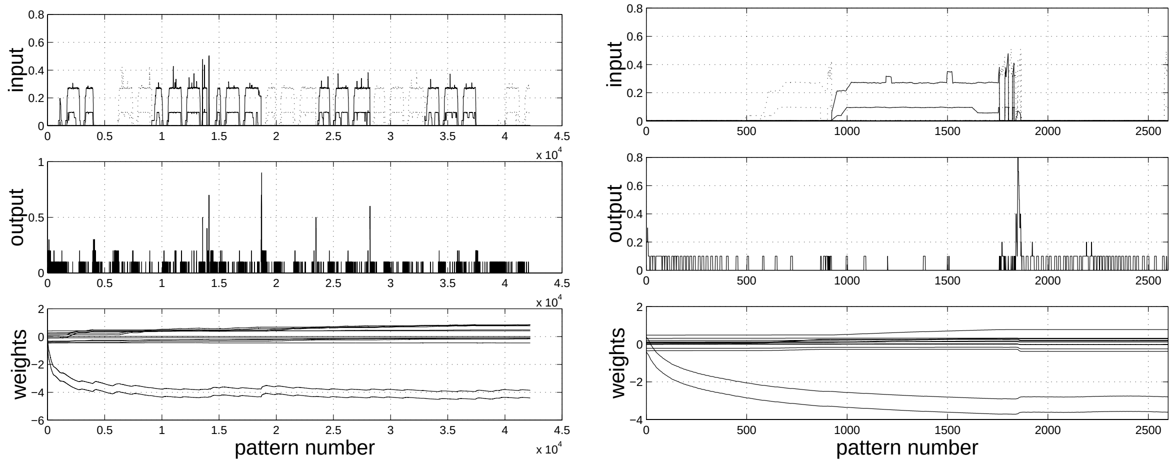

Figure 9: Convergence of the backprop weights of the fully exposed (left) and highly selective (right) networks at the learning phase, as input-output patterns are passed through from the attention system. Perceptual events are distinguished by different line types.

gle presentation of events is not always sufficient for weight convergence. Also, the learner can be 'unlucky' if at the particular times that the attention system triggers learning, the demonstrator is doing something distracting from the task (such as turning away from the wall because of the interrupt, or even turning away because the learner is in the way!)

However, if we use the forgetting mechanism as described in Section 2, the learning system can be re-exposed to events. The actual number of times the learning system is exposed to the same event is governed by how long we let a node in the SOFM stay fully habituated before we dishabituate it. If we make this length 0, this is equivalent to no attention-triggering at all, since the attention system is always attentive. In contrast, if we make the full-habituation length very high, the network is exposed to very little information, as we have seen in Figure 7.

We compare the performance of these extremes in both the learning and recall phases. Figure 9 shows how patterns, passed through from the attention system, affect the convergence of the network weights. Perceptual (input) and motoric (output) events are shown as in Figure 7 (perceptual events are distinguished by two different line types), together with the weights: 12 input-to-hidden weights (6×2), and 2 hidden to output weights (2×1). The left plot is for a full-habituation length of 0, and the right plot for 40000 (henceforth referred to as the 'fully exposed' and 'highly selective' networks, respectively).

Although the highly selective network receives very few patterns, these patterns form a range of experiences as representative as the fully exposed network, where in the latter there is a lot of repetition. The weights of the highly selective network are therefore able to converge, almost to the point of convergence of the fully exposed network. In fact, less selective networks would reach that convergence point because they would be exposed to more patterns. Note that the fully exposed network converges quite early.

To see the importance of the final convergence point of the weights we look at the output recalled by the networks after learning is completed. To do this we let the robot wander around on its own randomly in the environment (with an obstacle-avoidance competence) such that it picks up random perceptions that trigger the different output values learned through the weights (no attention is used). Figure 10 shows the distributions of the output recalled by the fully exposed and highly selective networks.

The two peaks in the output recalled by the fully exposed network show that this network learned to distinguish between the two types of perceptual experience: one requiring very low turn-tendency (values close to 0), corresponding to moving parallel to a wall, and the other requiring a higher turn-tendency, corresponding to moving randomly when not near a wall. We also see two peaks in the output recalled by the highly selective network, which suggests that it too has learned to distinguish the experiences; however, in absolute value the output is higher than from the fully exposed system, and the left peak is perhaps not close enough to 0. This is not surprising, and is due to the weights converging at different points, as discussed above.

To conclude, a fully exposed network converges quite early, suggesting that exposure to so many patterns is not needed. Further, such a network obviously does not make any use of the attention system. On the other extreme, we see from a very selective network that single presentations are perhaps not quite enough, although the network learns reasonably well.

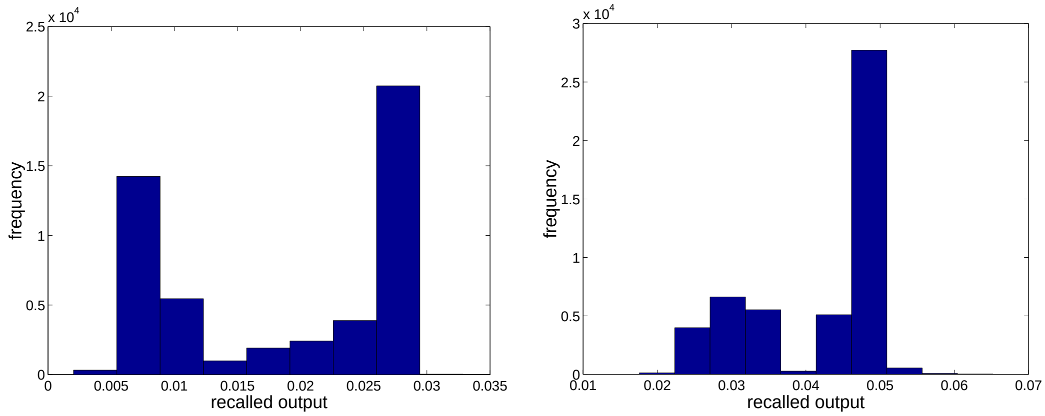

Figure 10: Distributions of the output recalled by the fully exposed (left) and highly selective (right) networks after learning is complete; the recall is triggered by input from a wandering behaviour in order to see all the possible values learned by the network.

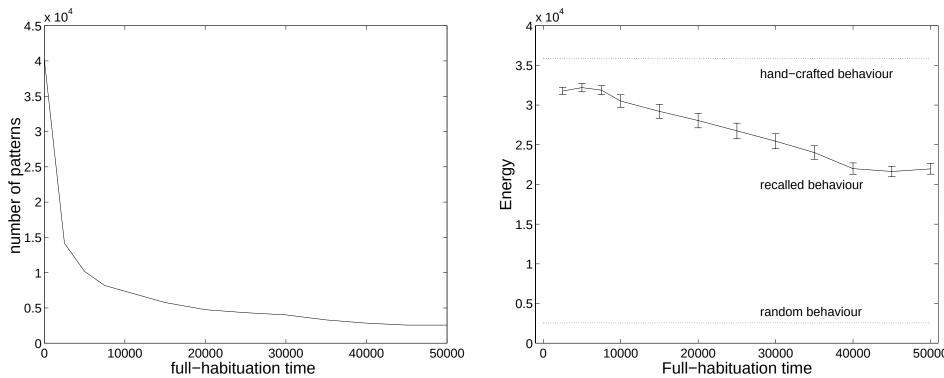

Figure 11: The number of patterns passed through from the attention system to the backprop network (left), and the energies acquired at the testing phase (right), as a result of different full-habituation lengths of the attention system at the learning phase. Each point is an average of 100 runs (the error bars in the left plot are of negligible lengths and therefore not shown); the length of a single run is 50000.

The effect of habituation

A good compromise must exist between these two extremes. In what follows, we look at the performance of a range of networks by modifying the length of the full-habituation time. In the plots of Figure 11, the independent variable is this length of time, where the experiment was repeated 100 times for each value shown. First, to show the effect of habituation on the exposure length, we plot the number of patterns presented to the network through attention-triggering, in the left of Figure 11.

As expected, when there is no attention triggering, the learning is exposed almost all the time (the only times it is not are when the learner loses the demonstrator). Attention triggering provides a substantial reduction in exposure, and the less forgetting we allow (*i.e.* longer time before dishabituation), the more reduction we get.

We have seen what the recalled output looks like (Figure 10), but how does this translate to the ability of the robot to reproduce the task of wall-following? To test this we place a threshold on the output neuron to determine whether to adopt a 'move-forward' (low output), or a 'wander' behaviour (high output). If we consider the fully exposed system as producing the most 'correct' learning, we can decide on such a threshold using the left of Figure 10 — a suitable threshold would be approx. 0.015.

The learner is placed in the environment on its own, its perceptual input is passed to the backprop network (no attention is used), where an output is computed. Further, a built-in 'obstacle-avoidance' is used to turn the robot when it is facing a wall, and prevent it from hitting obstacles.[3]

Equipped with this combination of built-in behaviours and recall from the backprop network, we calculate an energy measure which is the accumulation of the robot's side sensors sensing the wall. We use this as a measure of the robot's ability to perform the task (higher energies correspond to bet-

[3]We have only taught the robot what to do when there is a wall parallel to it, on its side; any other competencies would require a higher representational complexity, *i.e.* more output units, which we leave for future work.

ter wall-following). The right of Figure 11 shows the (95% confidence intervals of the) different energies acquired. In addition, the energies acquired by a hand-crafted wall-following behaviour, and a random-wander behaviour are shown.

We see that the energy only starts to drop when nodes stay fully-habituated for longer than 10000 steps, which is a fifth of the total run length. Below this time, the networks are re-exposed to events just enough times to ensure that the weights converge to a 'desired' point, and hence recall the 'desired' output.

To summarise, there is a substantial reduction in the exposure of the learning system due to attention-triggering (left of Figure 11), *and* this does not cause a significant decrease in the performance provided that forgetting is allowed reasonably frequently. In these cases the performance is almost as good as a hand-crafted behaviour. When forgetting is less frequent, the performance drops, but is still better than a random behaviour.

Learning on a real robot

We are currently implementing the learning back-prop system discussed in this section on the physical robot mentioned in the previous section (see Figure 4). 'Attention' on a real robot is much harder, as one is dealing with real, noisy data. Furthermore, our robot uses 20 perceptual sensors, compared with 6 in the Khepera mobile robot simulation, making the problem even harder.

Experimentation is also much harder as it is less practically possible to perform many experiments and control for all environmental conditions affecting the sensors (lighting, air-moisture etc.) We will discuss some early results here, but mention that we do not have enough data yet, and require more experimentation.

We have identified the emergence of perceptual and motoric events as in Figures 7 and 9 (it is hard to show, graphically, the activation of 20 sensors without grouping them!), but have not yet been able to achieve satisfactory weight convergence. This could be due to not having enough data — we are currently running more, longer experiments.

However, looking at the output recalled from the network, as in Figure 10, is quite encouraging. We have had to do this slightly differently: the robot's wander behaviour sees much less of the wall than in the simulated experiment, so by feeding the back-prop network input from a wander behaviour alone we would not see two peaks as in Figure 10. Therefore in addition to inspecting the output from a wander behaviour, we also inspect the output resulting from a hand-crafted wall-following behaviour. If the network has learnt correctly we expect the distributions of the two cases to be different, with lower values being recalled by the wall-following behaviour as it involves the robot being parallel to a wall most of the time (but not all the time).

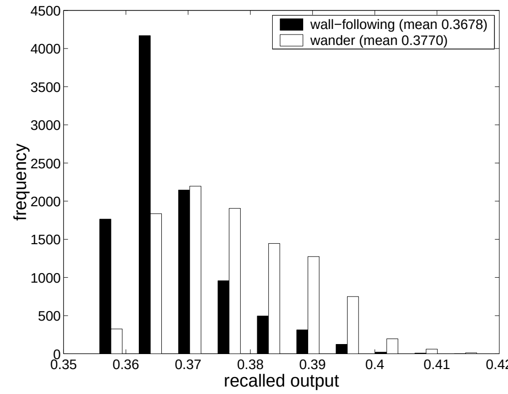

Figure 12: Distributions of the output recalled by a fully exposed network on the physical robot, after learning is complete; the recall is triggered by input firstly from a wandering behaviour and secondly from a hand-crafted wall-following behaviour.

The output recalled by a fully exposed network, learning approx. 6000 patterns out of a possible 10000 (10000 steps correspond to approx. 40 minutes. of experimentation), is shown for wander and wall-following behaviours, in Figure 12.

We see that on average the wall-following behaviour triggers lower outputs than the random behaviour (and these differences are statistically significant). We cannot expect the distributions to be completely separated because each behaviour shares some experiences of the other, but it is encouraging to see the differences mentioned.

4 Social Situatedness

The value of the social context has not been explicitly stressed in the paper, but is implicitly evident. Through a 'following' behaviour, the learner is exposed to the parts of the environment deemed important by the demonstrator. This helps the learner in the perceptual 'analysis' of the environment, in terms of the task (Marom and Hayes, 2001), and therefore in the construction of an attention network.

An interesting comparison would be between our attention-triggered learning system, and a reinforcement learning system, where the robot learns on its own. We would expect the socially-situated robot to learn faster, as it would not take it as long to discover the critical (rewarding) parts of the environment. We leave this for future work.

Finally, we are interested in implementing another form of social interaction, which could serve the role of an additional factor of dishabituation. When a neutral stimulus is followed by a rewarding one, the organism responds to the neutral stimulus, and this is a form of classical conditioning (Balkenius, 2000).

In the physical experiment reported above we have also used an additional stimulus: the human demonstrator waves a red glove to signal to the robot at critical parts of the task. The rewarding stimulus in this case is rather artificially implanted to draw attention, so strictly speaking this is not classical conditioning in the sense that the rewarding stimulus does not naturally occur in the environment without the presence of the demonstrator. Nevertheless, for a robot learning noisy data on-line, this could provide another useful enhancement — we are currently analysing the results.

5 Conclusion

We have presented an attention system, motivated from psychology and neurophysiology. With this system, a robot, situated in a physical environment, is able to build up perceptual representations of its experiences, and habituate to stimuli present in the environment. Additionally this system is capable of reinstating a reaction to the environment if this is needed, either due to novelty or to the passage of time (forgetting). In future work the system will also be supplemented with a mechanism for utilising stimulus rewards.

Social situatedness is a source of implicit information transfer. The input from an external source, in the form of a demonstrator, is of significant value for this system, as it provides it with relevant and well-structured experiences in terms of a particular task. We have discussed experiments from both a simulated environment, and a physical one involving a human demonstrator.

Lastly, we have also seen how the attention system is useful as a triggering mechanism for learning new reactive skills, and since the system determines *when* learning of perception-action mappings is triggered, the level of exposure provided by the attention system is important.

References

Afifi, A. A. and Clark, V. A. (1996). *Computer-Aided Multivariate Analysis*. Chapman & Hall, London. ISBN 041273060x.

Balkenius, C. (2000). Attention, habituation and conditioning: Toward a computational model. *Cognitive Science Quarterly*, 1(2).

Cowan, N. (1988). Evolving conceptions of memory storage, selective attention, and their mutual constraints within the human information processing system. *Psychological Bulletin*, 104:163–191.

Galef, Jr., B. G. (1988). Imitation in animals: history, definition, and interpretation of data from the psychological laboratory. In *Social Learning: Psychological and Biological Perspectives*, pages 3–28. Lawrence Erlbaum Associates, Hillsdale, N.J.

Itti, L. and Koch, C. (2001). Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203.

Kahneman, D. (1973). *Attention and Effort*. Prentice Hall, Englewood-Cliffs, New Jersey.

Maistros, G., Marom, Y., and Hayes, G. M. (2001). Perception-action coupling via imitation and attention. To appear in *AAAI Fall Symposium on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems*, North Falmouth, November 2001.

Marom, Y. and Hayes, G. M. (2001). Interacting with a robot to enhance its perceptual attention. In Nehmzow, U. and Melhuish, C., editors, *Proceedings of Towards Intelligent Mobile Robots (TIMR) 2001*, Department of Computer Science, Manchester University.

Marsland, S., Nehmzow, U., and Shapiro, J. (2001). Novelty detection in large environments. In Nehmzow, U. and Melhuish, C., editors, *Proceedings of Towards Intelligent Mobile Robots (TIMR) 2001*, Department of Computer Science, Manchester University.

Nehmzow, U. (1999). *Mobile Robotics: A Practical Introduction*. Springer Verlag. ISBN 1-85233-173-9.

O'Reilly, R., Norman, K., and McClelland, J. (1998). A hippocampal model of recognition memory. In *Advances in Neural Information Processing Systems*, volume 10, pages 73–79. MIT Press.

Posner, M., Snyder, C., and Davidson, B. (1980). Attention and the detection of signals. *Journal of Experimental Psychology General*, 109:160–174.

Sokolov, E. (1963). A neuronal model of the stimulus and the orienting reflex. In Sokolov, E., editor, *Perception and the Conditioned Reflex*, pages 282–294. Pergamon Press.

Spence, K. W. (1937). Experimental studies of learning and higher mental processes in infra-human primates. *Psychological Bulletin*, 34:806–850.

Whiten, A. (2000). Primate culture and social learning. *Cognitive Science special issue on Primate Cognition*, 24. In press.