Available online at www.sciencedirect.com
SCIENCE DIRECT®
Robotics and Autonomous Systems 51 (2005) 191–206
Robotics and Autonomous Systems
www.elsevier.com/locate/robot

On-line novelty detection for autonomous mobile robots

Stephen Marsland[a,*], Ulrich Nehmzow[b], Jonathan Shapiro[c]

[a] *Institute of Information Sciences and Technology, Massey University, Private Bag 11222, Palmerston North, New Zealand*
[b] *Department of Computer Science, University of Essex, Colchester CO4 3SQ, UK*
[c] *Department of Computer Science, University of Manchester, Manchester M13 9PL, UK*

Received 1 November 2003; received in revised form 27 July 2004; accepted 20 October 2004
Available online 17 February 2005

Abstract

The use of mobile robots for inspection tasks is an attractive idea. A robot can travel through environments that humans cannot, and can be trained to identify sensor perceptions that signify potential or actual problems without requiring human intervention. However, in many cases, the appearance of a problem can vary widely, and ensuring that the robot does not miss any possible appearance of the problem (false negatives) is virtually impossible using conventional methods.

This paper presents an alternative methodology using novelty detection. A neural network is trained to ignore normal perceptions that do not suggest any problems, so that anything that the robot has not sensed before is highlighted as a possible fault. This makes the incidence of false negatives less likely.

We propose a novelty filter that can operate on-line, so that each new input is evaluated for novelty with respect to the data seen so far. The novelty filter learns to ignore inputs that have been sensed previously, or where similar inputs have been perceived. We demonstrate the use of the novelty filter on a series of simple inspection tasks using a mobile robot. The robot highlights those parts of an environment that are novel in some way, that is they are not part of the model acquired during exploration of a different environment. We show the effectiveness of the method using inputs from both sonar sensors and a monochrome camera.

Keywords: Novelty detection; On-line learning; Mobile robot inspection

1. Introduction

The ability to detect novelty, that is, to recognise and respond to stimuli that do not fit into the class of expected perceptions, is very useful for animals and robots alike. For animals, novelty detection is an important survival trait — the unexpected perception could signify a potential predator, while for robots a novel stimulus could be some important feature of an environment, a potential problem, or something that has to be learnt.

In this paper, we consider the problem of training a robot to act as an inspection agent. Robots can examine environments that are not safe for humans,

* Corresponding author. Tel.: +64 6 350 5799; fax: +64 6 350 2259.
E-mail addresses: s.r.marsland@massey.ac.nz (S. Marsland), udfn@essex.ac.uk (U. Nehmzow), jls@cs.man.ac.uk (J. Shapiro).

0921-8890/$ – see front matter
doi:10.1016/j.robot.2004.10.006

and will not get bored examining similar features over and over again. Suitable applications include sewer inspection, or remote sensing, although such applications are not considered in detail here. Unfortunately, in such applications, the appearance of a fault can vary widely, and it will not be possible to guarantee that every different appearance has been seen, so that false negatives are likely. We approach the problem as one of novelty detection, so that the robot is trained to ignore perceptions that are similar to those seen during training, but to highlight anything different. In this sense, novelty detection can be seen as a form of negative learning — examples are provided of those features that should not be detected, and the novelty filter aims to highlight anything that differs from the inputs used in training. Provided that a good training environment can be found, which does not contain any faults, the novelty detection approach should then significantly reduce the incidence of false negatives.

In essence, as a robot travels within an environment, it generates a stream of data from its sensors. The robot should learn about the environment as it travels through it, in something close to real-time, which means that for a novelty filter to be useful in robots tasks, it needs to be able to operate on-line, so that the robot can deal with changing environments. From a practical point of view, it also needs to be able to deal with noisy data, as robot sensors are notoriously noisy, and to be able to deal robustly with seeing some novel perceptions during training, as it is difficult to tailor an environment perfectly. We therefore propose such an adaptive on-line novelty filter.

Novelty detection is also useful for other machine learning problems. Any learning system is only as good as the data that it is trained on, and often the data is noisy and subject to outliers. Detecting outliers in multivariate data is a difficult, but very useful task. Typical applications of novelty detection are in diagnosing medical problems or detecting machine faults. Typically for these problems there is a lot of data about the normal classes (healthy people or machines without problems), but very little data displaying the features that should be detected. For this reason, it is common to learn a model of the normal dataset and then attempt to detect deviations from this model for further processing. Note that in this paper the model that is learnt is of the pattern of inputs. It is not a map of the environment that the robot is travelling though, but of the perceptions of the robot as it travels. Thus, while the novelty filter described here might be useful for a navigating robot (for example, by recognising new places because they provide different sensory perceptions), it does not learn a representation of an environment.

The novelty filter described in this paper uses the biological phenomenon of habituation. Habituation, is a decrease in an animal's response to a stimulus when the stimulus is presented several times without ill effect. Habituation enables an animal to ignore stimuli that it sees often, so that it can concentrate on other, potentially more important, stimuli. This is the behaviour that we desire of a novelty filter — highlighting novel stimuli, but otherwise not giving any response.

This paper describes a novelty filter based on a clustering neural network and habituation. The application of the novelty filter to mobile robot inspection is demonstrated through a series of robotic experiments where an autonomous mobile robot travels through a series of corridor environments using a wall-following behaviour, and presenting its sensor readings to the novelty filter, which highlights features in those environments that have not been perceived previously, thus acting as an inspection agent. A novelty filter is also useful for the related tasks of directing attention and filtering out commonly seen stimuli. The trained filter will not pass through information about inputs that have frequently been seen before, meaning that any inputs that do get through the filter are worthy of attention by the learning system. This can speed up learning.

Section 2 of this paper provides a brief discussion of some relevant papers in the literature. This is followed by a description of the novelty filter in Section 3. Section 4 describes a number of experimental results using the novelty filter as an inspection agent on an autonomous mobile robot. Sensory inputs from sonar sensors and a monochrome camera are captured as a robot explores a set of corridors and used as input to the novelty filter. The robot learns a model of an environment and then detects deviations from that model, highlighting these novel features. The experiments demonstrate the filter being used in both small (~10 m) and large (~300 m) environments.

2. Related work

There have been a number of novelty detection techniques proposed in the literature, we highlight here those approaches that are directly relevant to this work; for a comprehensive overview, see [1].

Some researchers have considered using Kohonen's Self-Organising Map (SOM) [2] as the basis for a novelty detector. Ypma and Duin [3] describe a number of measures by which the correctness of fit of a SOM with respect to a particular dataset can be evaluated. They measure the average quantization error over the dataset, as well as the distance between map units that respond to similar inputs. The SOM is trained off-line on a dataset of normal inputs, and then the measures are evaluated on a new dataset. This provides information about whether or not the new dataset fits the same distribution as the data that was used to train the SOM, but cannot be used to categorise individual inputs, nor can it be trained on-line.

An alternative method using the SOM was described by Taylor and MacIntyre [4]. In their approach, the training data was used to select a set of neighbourhoods of the SOM that described normality. After training, new inputs were introduced, and any input that caused other nodes in the network to fire (or at least, nodes that were not in some pre-defined neighbourhood of the nodes that fired in training) were highlighted as novel.

There have been a couple of examples of novelty filters being used in robotics. The utility of novelty detection for intelligent robotics was identified by Brooks [5], where 'monitor changes' was one of the eight behaviours that were required for a task achieving robot. Öğmen and Prakash [6] used a field of gated dipoles [7] as part of a system that 'explored' a workspace by moving the end effector of a robot arm to places that had not been visited before. This was done by quantizing the workspace and associating a dipole with each area. Their work also considered recognising that objects were novel, so that the robot arm would pick them up. This was done by taking the output of an ART network [8], which performs classification, and feeding it into a network of gated dipoles.

The FamE (Familiarity based on Energy) novelty detector [9], has also been applied to robot tasks [10]. This novelty filter is based on the Hopfield network, and evaluates the energy of the network for each input, with inputs that do not settle into stable states, and therefore have high energy, being thought to be novel. For the robot experiments, a mobile robot took photographs of a 'picture gallery' of orange rectangles on a white wall and the FamE model evaluated the novelty of the simple images produced. In [10], the FamE approach is compared to the novelty filter that is described in this paper.

In general, these approaches are not suited to on-line learning, an important feature for a novelty filter suitable for use on an autonomous mobile robot. On-line learning means that the robot can be partially trained, its learning tested, and then further training applied without having to start again. Nor, in general, do these novelty filters show much robustness to any features that should be found to be novel being seen in the training set. For robotic applications it is hard to control exactly what the robot will see in every environment that it explores, and so this feature is very important. A novelty filter that can learn on-line, and that has some robustness to accidental training (i.e., occasionally seeing features during training that should be identified as novel), is the subject of the next section.

3. A novelty filter based on habituation

We are considering how to make a robot learn to detect novelty in an environment by exploration. This requires on-line learning, so that each new feature that is seen is learnt, and some form of filtering to desensitise the algorithm to features that are seen repeatedly, i.e., novelty filtering. This section describes our method, which is based on a clustering network that classifies and learns about the current input, and a set of habituating synapses linking the nodes of the network to an output node. These habituating synapses decrease in strength as the nodes they are connected to fire.

Habituation provides a way of recording whether or not a network node has often fired before. If the strength of the synapse is high, then the node has not fired, while if the strength is low, then the node has fired often, with other values inbetween. We can therefore use habituation to describe how often a node has fired during the operation of the network. We use a simple model of habituation (h) as an exponential function (for

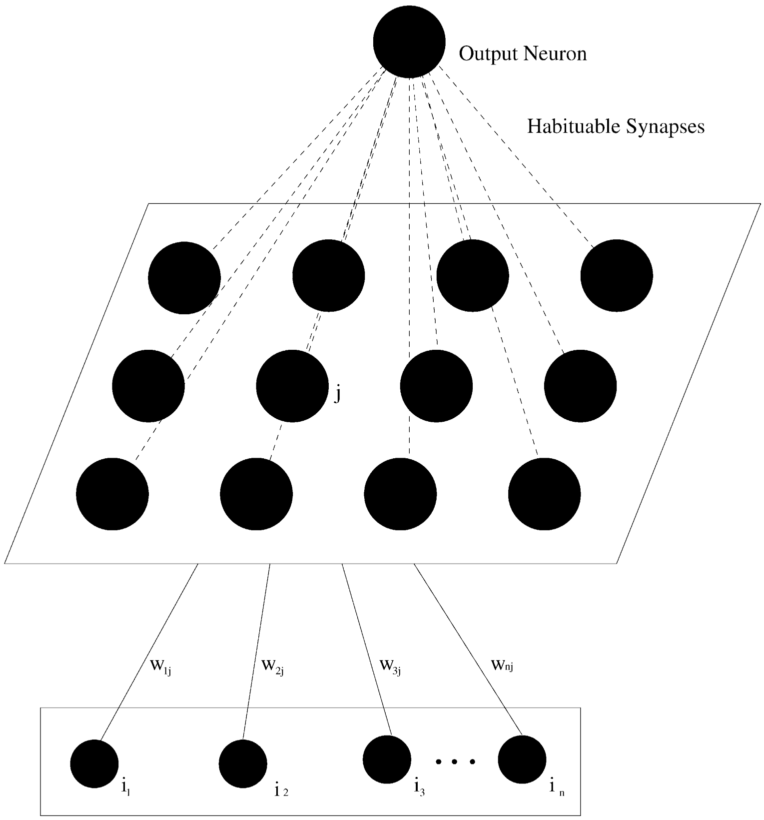

Fig. 1. A schematic of a novelty filter based on clustering and habituation. The input layer connects to a clustering layer that represents the feature space, the winning node (i.e., the one ‘closest’ to the input) passes its output along a habituable synapse to the output node, so that the output received from a node reduces as it fires more often.

more detail, see [11]):

$$h = h_0 - \frac{S}{\alpha}(1 - \mathrm{e}^{\frac{-\alpha t}{\tau}}), \tag{1}$$

where h_0 is the original strength of the synapse (usually 1), and S records whether or not a stimulus is applied ($S = 1$ or $S = 0$). The other parameters are discussed in Section 3.1.1.

By attaching an habituating synapse to each of the nodes in a clustering network performing a winner-takes-all learning strategy, a mechanism is produced that classifies each input and describes how unusual that input is with respect to previously seen inputs. A schematic of this general approach is shown in Fig. 1. The next section considers a suitable choice of clustering network to apply habituation to.

3.1. The Grow When Required (GWR) network

Earlier research (see [12] for details) used the SOM as the clustering basis of the novelty filter. However, for robotics inspection tasks, which require on-line novelty detection, the SOM is not suitable, because it is not designed for on-line learning. For optimal results, SOM training is performed in batch, and the weights of the network are initialised so that the nodes lie along the directions of the principal components of the data. Without these conditions, there is no guarantee of the convergence of the SOM, nor for the topography preservation of the input space [13].

Furthermore, with on-line learning it is possible to saturate a novelty filter based on the SOM so that any perception, even one that is completely novel, is considered to be normal. This happens if the habituation synapses of all the nodes in the map field habituate.

In this case, no matter what input was presented, the best matching node (which may still be a very poor match) will fire and, as the synapse has habituated, the output of the network will be that the perception is normal. This would be the case even if a completely novel stimulus was seen.

To overcome this saturation problem we developed a growing neural network that can dynamically grow to meet the demands of the particular data space that is being learnt. This has benefits in that there is no need to decide in advance how large the network should be, as nodes will be added until the network is large enough. This means that for small datasets the complexity of the network is significantly reduced. In addition, if the dataset changes at some time in the future, further nodes can be added to represent the new data without disturbing the network that has already been created.

The clustering network that was developed for this research, the 'Grow When Required' (GWR) network has these properties. The network is described below, and more details, together with experiments and analysis demonstrating the topology-preserving properties of the network, and comparisons to other algorithms, are given in [12,11]. Our algorithm has similarities to the Growing Neural Gas [14] and FOSART [15].

Fritzke's Growing Neural Gas (GNG) is not suitable for the task of novelty detection because it only adds new nodes every λ iterations, where λ is some predefined integer. This is because new nodes are added to support the node with the highest accumulated error from the data presented during those λ iterations. A novelty filter should highlight novel inputs immediately, which the GNG will not do, and furthermore, a novel input that occurred just once during the error accumulation stage may not be enough to cause a new node to be added to recognise it — there may also have been errors accumulated in other parts of the network.

The work of Baraldi et al., known as FOSART [16,15] and the Incremental Topology Preserving Map of Millan et al. [17] are conceptually similar to the novelty filter described in this paper. They consider the problem of growing neural networks through a merger of the class of simplified-ART (SART) networks and the GNG. SART networks preserve the concept of vigilance to detect mismatches between inputs and stored categories, but remove the resonance condition used for categorisation by ART. This makes them very similar to the unsupervised RCE network [18]. These networks do not have the neighbourhood relations that are useful in novelty detection tasks, so that similar categories can be linked together and, in the framework of this paper, habituate each other. Neighbourhood relations have other benefits to the operations of a self-organising network, too, particularly with the positioning of new nodes within the input space.

We now detail the precise steps of the algorithm. In order to facilitate comparisons, the new algorithm is described using the same notation as used in [14].

Let A be the set of map nodes, and $C \subset A \times A$ be the set of connections between nodes in the map field. Let the input distribution be $p(\boldsymbol{\xi})$, for inputs $\boldsymbol{\xi}$. Define $\boldsymbol{w_c}$ as the weight vector of node c.

Initialisation:

Create two nodes for the set A,

$$A = \{c_1, c_2\} \tag{2}$$

with c_1, c_2 initialised randomly from $p(\boldsymbol{\xi})$. Define C, the connection set, to be the empty set,

$$C = \emptyset \tag{3}$$

Then, each iteration of the algorithm looks like this:

(1) Generate a data sample for input to the network, $\boldsymbol{\xi}$.
(2) For each node in the network, calculate the distance from the input.
(3) Select the best matching node, and the second best, that is the nodes $s, t \in A$ such that

$$s = \arg\min_{c\in A} \|\boldsymbol{\xi} - \boldsymbol{w_c}\|, \tag{4}$$

and

$$t = \arg\min_{c\in A/\{s\}} \|\boldsymbol{\xi} - \boldsymbol{w_c}\|, \tag{5}$$

where $\boldsymbol{w_c}$ is the weight vector of node c.

(4) If there is not a connection between s and t, create it:

$$C = C \cup \{(s, t)\}, \tag{6}$$

otherwise, set the age of the connection to 0.

(5) Calculate the activity of the best matching unit,

$$a = \exp(-\|\boldsymbol{\xi} - \boldsymbol{w_s}\|). \tag{7}$$

(6) If we should add a node, i.e., if activity $a <$ activity threshold $a_{\mathrm{T}} = 0.7$ and habituation $<$ habituation threshold $h_{\mathrm{T}} = 0.15$:
   - Add the new node, r, between the two best matching nodes (s and t)

$$A = A \cup \{r\}. \tag{8}$$

   - Create the new weight vector, setting the weights to be the average of the weights for the best matching node and the input vector:

$$\boldsymbol{w}_r = \frac{\boldsymbol{w}_s + \boldsymbol{\xi}}{2}. \tag{9}$$

   - Insert edges between r and s and between r and t:

$$C = C \cup \{(r, s), (r, t)\}. \tag{10}$$

   - Remove the link between s and t:

$$C = \frac{C}{\{(s, t)\}}. \tag{11}$$

(7) Adapt the positions of the winning node and its neighbours, i, that is the nodes that it is connected to.

$$\Delta \boldsymbol{w}_s = \epsilon_b \times h_s \times (\boldsymbol{\xi} - \boldsymbol{w}_s), \tag{12}$$

$$\Delta \boldsymbol{w}_i = \epsilon_n \times h_i \times (\boldsymbol{\xi} - \boldsymbol{w}_i), \tag{13}$$

where $0 < \epsilon_n < \epsilon_b < 1$ and h_i is the value of the habituation counter for node i.

(8) Age edges with an end at s:

$$\mathrm{age}_{(s,i)} = \mathrm{age}_{(s,i)} + 1. \tag{14}$$

(9) Reduce the strength of the habituating synapses for the winning node and its neighbours using:

$$h = h_0 - \frac{S}{\alpha}(1 - \mathrm{e}^{\frac{-\alpha t}{\tau}}), \tag{15}$$

where $h(t)$ is the strength of the synapse, $h_0 = 1$ is the initial strength, and $S(t) = 1$ is the stimulus strength. $\alpha = 1.05$ and τ are constants controlling the behaviour of the curve, $\tau_b = 3.33$ for the winning node and $\tau_n = 14.33$ for the neighbouring nodes. This means that the winner habituates faster than its neighbours.

(10) Check if there are any nodes or edges to delete, i.e., if there are any nodes that no longer have any neighbours, or edges that are older than the greatest allowed age.

3.1.1. Parameters

There are several parameters that are critical to the algorithm. Of particular importance is the activity threshold $0 < a_{\mathrm{T}} < 1$, which controls when new nodes should be added according to the mismatch between the input and the best-matching node. The effects of varying this parameter are shown when the GWR algorithm learns about some simple two-dimensional datasets in [11], but in practice, coarse learning about a robot's environment occurs for values of $0.6 < a_{\mathrm{T}} < 0.8$, and more detailed learning (resulting in significantly larger networks being created) for higher values of a_{T}. The value of $a_{\mathrm{T}} = 0.7$ was used for all of the experiments described in this paper.

As well as the activity threshold, the habituation threshold h_{T} is also used to decide whether or not to add a new node. The idea here is that new nodes have not yet been trained, and therefore it is more effective to train the node than to add another new one. With the choice of parameters controlling the habituation curve (which were chosen so that it took about six presentations of a particular stimulus to fully habituate) given in step (9) of the algorithm, $h_{\mathrm{T}} = 0.15$ means that a node should be the best match four or five times before a new node is added nearby. It was observed experimentally that changing this parameter between $0.05 < h_{\mathrm{T}} < 0.4$ did not significantly affect the behaviour of the algorithm.

Finally, in common with other self-organising networks, there are the learning rates for the winning node and its neighbours. In the experiments reported in this paper $\epsilon_b = 0.3$ and $\epsilon_n = 0.15$, values that were determined experimentally on test data. It should be noted that these learning rates are multiplied by the habituation counter of the node, which means that nodes will move less after they have received previous training. This makes the network more stable.

3.1.2. Dishabituation and forgetting

In the system described above, only the habituation synapses of the winner and its topological neighbours were affected. In this case, the other synapses do not change value. However, if instead the synapses that are not in the topological neighbourhood are instead given an input of $S(t) = 0$, then those synapses recover some of their strength, in effect forgetting previous habituation — an effect known as dishabituation. This means that things that are seen only infrequently are considered novel, which can be useful for certain tasks. In addition, perceptions that are seen only once will be forgotten, so that transients in the training set are ignored.

Another effect can be created using forgetting. If two habituating synapses are attached to each node, with one forgetting and the other not, then the effect will be that the concept of ‘recency’ is encoded. If the synapse that does not forget is habituated, but the forgetting one is not, then the perception that is matched by the relevant node has been seen, but not recently, whereas if both synapses are habituated the perception has been seen in the recent past. This acts as a kind of memory, and would enable some temporal novelty to also be detected, although this is not investigated in this paper.

4. Experimental results

This section describes robotic experiments concerned with on-line environment inspection using the novelty filter. For examples of applications of the novelty filter to non-robotic tasks, and comparisons with other novelty filters, see [12,11]. The experiments presented here focus on a simplified inspection task, where the robot learns an internal representation of sensor perceptions of an environment through exploration, and then signals deviations from that model when exploring further environments. The robot explores an environment, and uses the novelty filter to build a model of the perceptions that it receives in that environment. Once the model is complete, the robot explores another environment, highlighting perceptions that are found to be novel, i.e., that were not seen in the first environment. This task was performed using the Nomad 200 robot shown in Fig. 2. This is a fully autonomous robot equipped with 16 sonar sensors and 16 short-range infrared sensors.

The aim of an inspection robot is not to learn maps of a set of environments, but to learn what perceptions (i.e., sensory inputs) are found in those environments, and then to detect different sensory perceptions. In the first set of experiments that are described here, the 16 sonar sensors are used as the inputs to the novelty filter, which results in a very coarse representation of the environment. Obviously, this would not be sufficient for a real inspection agent, but we offer a proof-of-concept for an inspection agent based on a novelty filter. A more substantial experiment, where inputs are based on camera images rather than sonar readings, is described in Section 4.4.

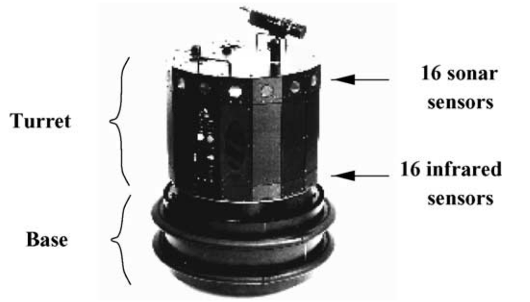

Fig. 2. The Nomad 200 robot.

4.1. Experimental procedure

We allowed the Nomad 200 robot (shown in Fig. 2) to explore a number of typical corridor environments such as are found in any office building, in this case the top floor of the Computer Science Department at the University of Manchester. The robot’s infrared sensors were used to perform a wall-following behaviour. In the first experiments the sonar sensors were used to sample the environment. It is the readings of the sonar sensors that were used as input to the novelty filter. The use of a wall-following behaviour meant that the robot took similar – although not identical – paths through the environment each time that it travelled through it, staying between 15 and 35 cm from the right-hand wall. If a different control program was used to control the path of the robot through the environments, then the perceptions of the robot would be different. However, provided that the same control program is used on each run, similar perceptions will be seen, and the novelty filter will operate effectively.

Each experiment consisted of the same steps, repeated a number of times. The robot was positioned at a starting point chosen arbitrarily in the environment. From this point the robot travelled along the corridor using a wall-following behaviour. Approximately every 10 cm along the way the smoothed values of the sonar perceptions over that 10 cm of travel were presented to the novelty filter, which produced a novelty value for the perception. At the end of the run the robot was returned to the starting location under manual control and the procedure was repeated, but starting with the weights learned during the previous run. Inbetween each pair of learning runs a test run, where the novelty filter evaluated the novelty of each input but did not learn about it, was performed, to monitor the learning process.

This procedure was repeated until the robot found nothing in the environment to be novel anymore, which typically took three learning passes. At that point the environment was changed, either by moving the robot manually to a new environment, or by making a change to an environment such as opening a door.

Three sets of experiments are described here. In the first, three different 10 m sections of corridor were used. Two of them were similar corridors in the same part of the building, while the third was very different, although still a corridor environment. In addition, the first of the environments was changeable, in that a door could be open or shut. These environments are shown in Fig. 3. In the second set of experiments,

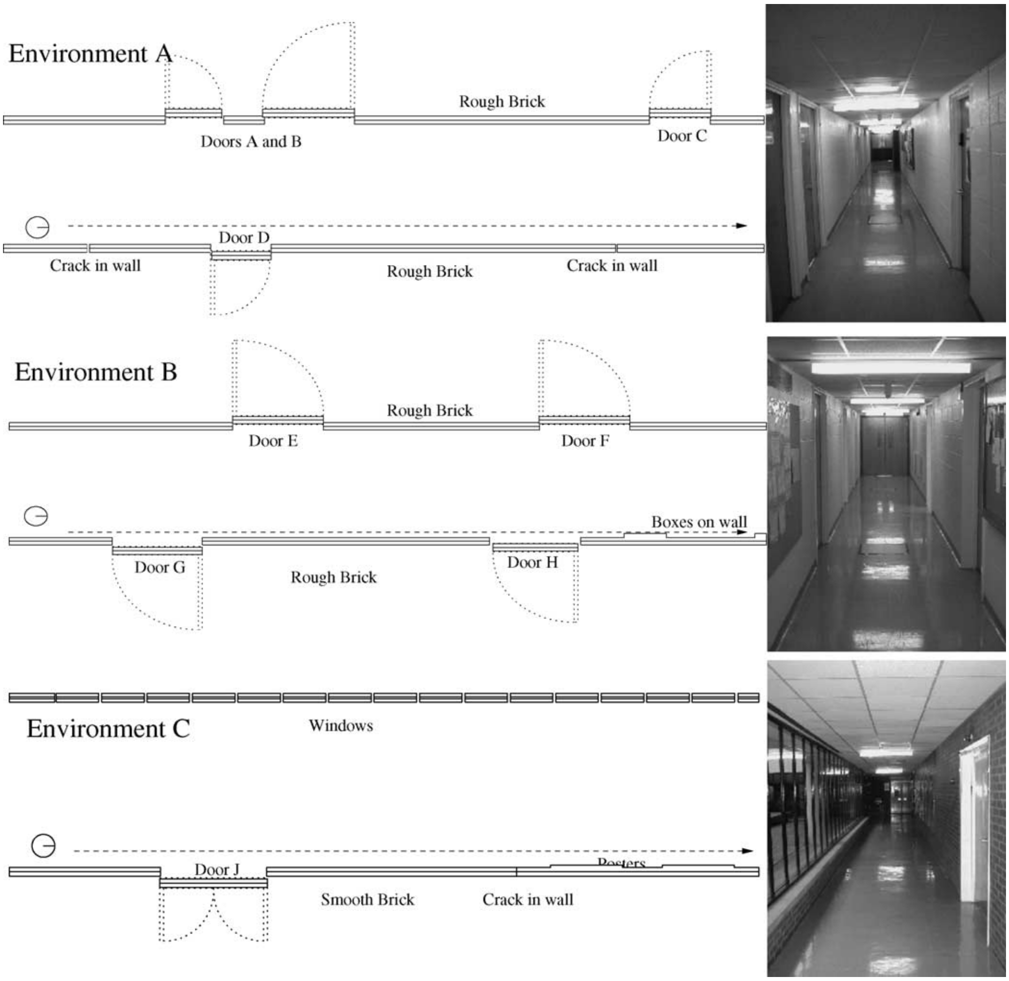

Fig. 3. Diagrams of the three environments used in the inspection task. The robot is shown facing in the direction of travel adjacent to the wall that it followed. Environments A and B are two similar sections of corridor, while environment C is wider and has walls made of different materials, glass and brick instead of breezeblock. The photographs show the environments as they appear from the starting position of the robot. The notice boards that are visible in environment B are above the height of the robot's sonar sensors, and are therefore not detected. Door D in environment A was opened for some of the experiments. When the door is open the environment is referred to in the text as environment A*.

described in Section 4.3, a 300 m long loop of corridor was used to show that the system scales effectively. Finally, in Section 4.4 some experiments using an input vector taken from camera images rather than the sonar sensors, are presented, to show how the filter deals with much richer information about the environment.

4.2. Short corridor experiments

4.2.1. Learning without forgetting

Fig. 4 shows how the novelty filter learns about the short environments. As the robot travels through the environment, the novelty of each perceptions was computed; this value is shown in the graphs in the figure – a peak in the graph means that the current perception is very novel, and no spike means that the current perception has often been seen before.

The top left of Fig. 4 shows the initial training when an empty novelty filter is used and the robot explores environment A (shown in Fig. 3). It can be seen the robot rapidly learns to recognise the wall that is seen most of the time, but that the doorway is found to be novel for several steps. However, after three learning runs in the environment, the network finds nothing to be novel. The top right section of Fig. 4 shows the effects when this trained network is used after a change has been made to the environment. A door to the right of the robot was then opened (door D in Fig. 3) so that the perceptions at this point changed, but otherwise the environment remained exactly the same. It can be seen that the robot finds the perceptions of the open doorway novel, but does not find any other perceptions novel. It then takes two learning trials for this change in the environment to be learned, after which nothing is found to be novel again.

Finally, the bottom of the figure shows that the model generalises to other, similar, environments. Environment B is a similar section of corridor, and the only highlighted differences are in areas around the doorways, which are significantly further inset. This is contrasted with a control trial shown in the bottom line of that figure, where the robot was trained in a completely different environment (a random walk around a large room), where every perception is found to be novel.

Fig. 5 shows how the amount of novelty found in an environment decreases as the robot explores it more. The amount of novelty is calculated by summing the output of the output node for each input. In each environment it took three passes through the environment before the robot stopped finding anything novel. For each figure, the robot explored environment A for three runs, until it stopped finding it novel, and was then moved into a new environment. The graphs show that environment A* is not much different to environment A (it is the same except that a door is opened), because the amount of novelty does not increase much. Again, when the robot first perceives environment B, which is a similar piece of corridor, there is not a large increase in novelty. However, when the robot explores environment C it finds a great deal of it novel. In fact, the amount of novelty is nearly as much as when the robot explores environment A without any previous training. This is to be expected as it is a very different environment. The growing novelty filter learns a compact representation of the robot's perceptions, during training in each of the environments an average of 4.5 new nodes were added.

4.2.2. Learning with forgetting

The effects of forgetting (dishabituation) were described in Section 3.1.2. Fig. 6 shows what happens to the evaluation of novelty in the network when forgetting is turned on, so that synapses can dishabituate as well as habituating. The two graphs show the effects of training the robot in environment A, and then letting the robot explore a second environment with forgetting turned on. This meant that the habituated synapses for features that are not seen in the second environment dishabituate, so that those perceptions are found novel again. In the graph on the left of the figure, the robot explores environment A* with forgetting turned on. After each training run in environment A*, the robot is again run in environment A, but with learning turned off. It can be seen that as the robot learns about environment A*, where the perceptions of the closed doorway on the right of the robot are never seen, the amount of novelty found in environment A increases, as the amount of novelty found in environment A* decreases.

Similar results can be seen in the graph on the right of the figure. Here, the robot was trained with forgetting in environment B after training in environment A. Again, after every learning pass in environment B the robot was returned to environment A, but with learning turned off, and again, the amount of novelty found in

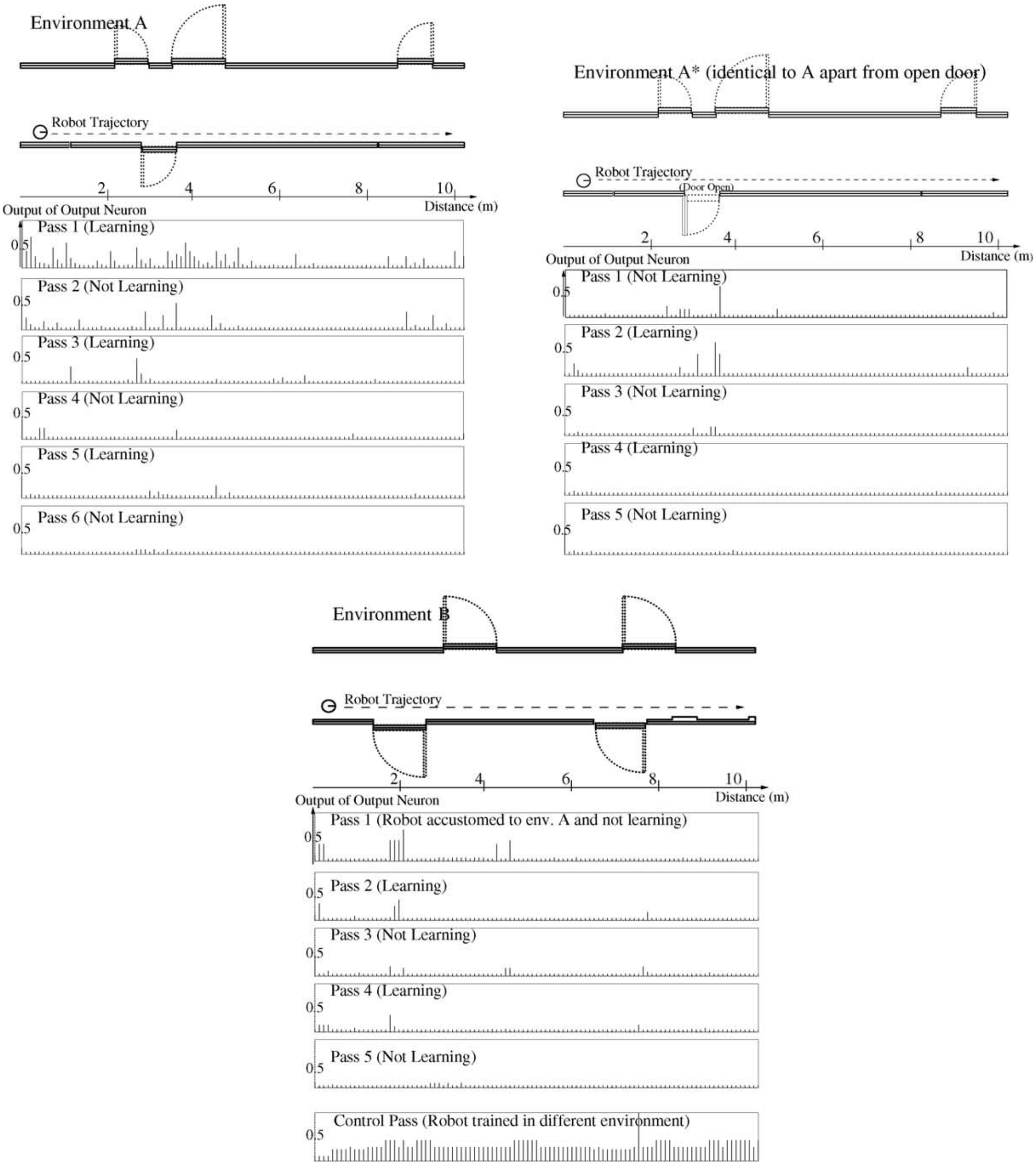

Fig. 4. The results of the first experiment. Top left: The output of the output node of the novelty filter as the robot moved within environment A when learning and not learning. Once it had stopped detecting novelty features (so that the activity of the output node is small), the environment was changed by opening a door (environment A* — top right). The only perceptions found to be novel were those around the doorway. Finally, bottom the trained filter is transferred to a novel, but similar, environment, where only things not seen in environment A were found to be novel.

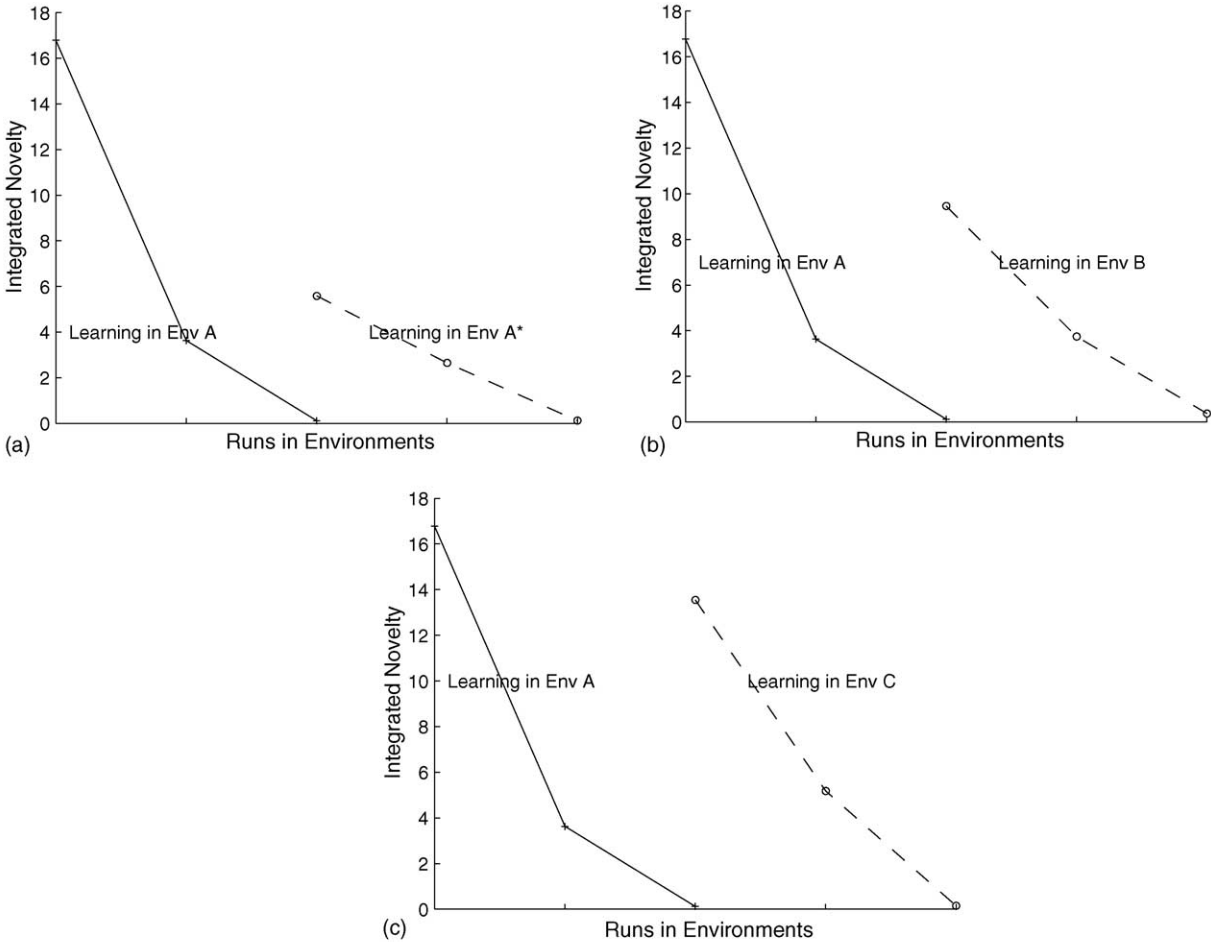

Fig. 5. Graphs showing how the amount of novelty in an environment decreases as the novelty filter learns over three runs in an environment. The graph on the top left shows the robot learning first in environment A, and then in environment A*. It can be seen that once the robot has learnt about environment A, very little in A* is novel. Fig. 4 shows that what is novel is the area around the (now open) door. The top right graph shows how the amount of novelty increases when the robot explores environment B after learning about environment A, and finally the graph on the bottom shows how the novelty increases when the robot begins to explore environment C. These results are discussed in Section 4.2.1.

environment A increased as the robot finds less novel in environment B. However, the amount of novelty found in environment A does not increase as much as it did for training in environment A*. This is because most of the features of environment A are also seen in environment B. In particular, the closed doorways are also seen in this environment.

4.3. A large environment

This experiment was designed to demonstrate that the growing novelty filter can deal with any size of environment. In this case, the environment that was explored was a loop of corridor on the top floor of the Computer Science Department at the University of Manchester. Overall, the loop is about 300 m long. The robot explored this environment using the wall-following behaviour described previously and, as before, used the novelty filter to compute the amount of novelty at each stage of exploration.

The output of the growing novelty filter is shown in Fig. 7. The graphs show the output of the novelty filter as the robot travels through the environment. At the end of the final (fifth) run a large number of card-

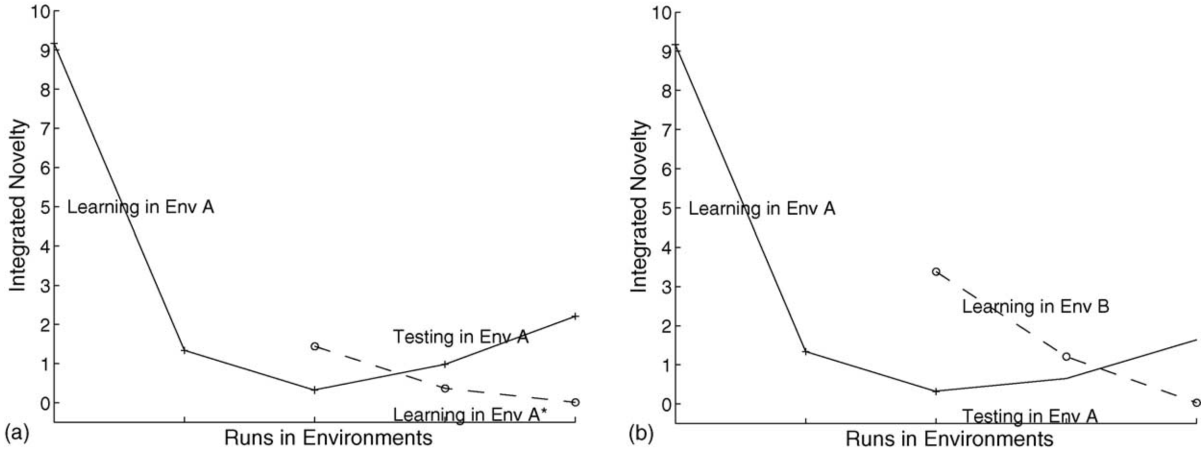

Fig. 6. The effects of forgetting on the amount of novelty found in an environment. In the left figure the robot explores environment A for three runs, as previously. Then the robot explores environment A*, with forgetting turned on. After each exploration of environment A*, the robot explores environment A. It can be seen that as the novelty filter finds environment A* less novel, so the amount of novelty found in environment A increases, as the robot forgets about these perceptions. The same is shown on the right for the robot learning about environment B. These experiments are discussed in Section 4.2.2.

board boxes were placed around the robot so that it reached a dead end in the corridor. This was done to give the robot a number of very novel perceptions. Otherwise, the robot was left to explore the environment without human intervention, although the experiments were performed at a time when the building was mostly empty, so that the environment was static.

As in the first experiment in Section 4.2.1, a non-learning trial followed each learning trial, in order to see how much of the environment had been learned. In the first trial a lot of novelty was found, especially at corners, where the motion of the robot was more unpredictable. The second run, a non-learning trial, shows a great deal less novelty, which is the same in the second learning trial (trial three). There is no real novelty found in the second non-learning trial (trial four). It was in the final trial that the environment was changed at the end. In this trial nothing was found to be novel, with the exception of the perceptions at the end where the environment was altered, nothing was found to be novel. It is interesting to note that in the second trial most of the places that are found to be novel are the corners. This may well be because the robot can take a large number of different paths around a corner, and so the perceptions vary considerably.

At the end of the final trial, where the environment was modified to produce some very novel perceptions, the novelty filter has responded as expected, and shown that these perceptions do not fit into the model that has been generated.

4.4. Using visual input

The sonar sensors used in the previous experiments provide only a coarse, noisy picture of the environment. To improve on this we also investigated using the output of a 480×200 pixel monochrome camera that was mounted on the robot. We experimented with several different methods of capturing and pre-processing images, and of generating an input vector for the novelty filter from the image. We present the most effective method only, for details of the other methods tried, see [12].

The first problem was to ensure that the robot saw the same image each time that it was in approximately the same place. The principal difficulty was that the angle between the robot and the wall could change over time, so that the camera could face in different directions. To correct for this, an initial image was taken, vertical edges in the image detected, and the robot then rotated its turret (on which the camera was mounted) in order to point at the position where the histogram of edges was centred.

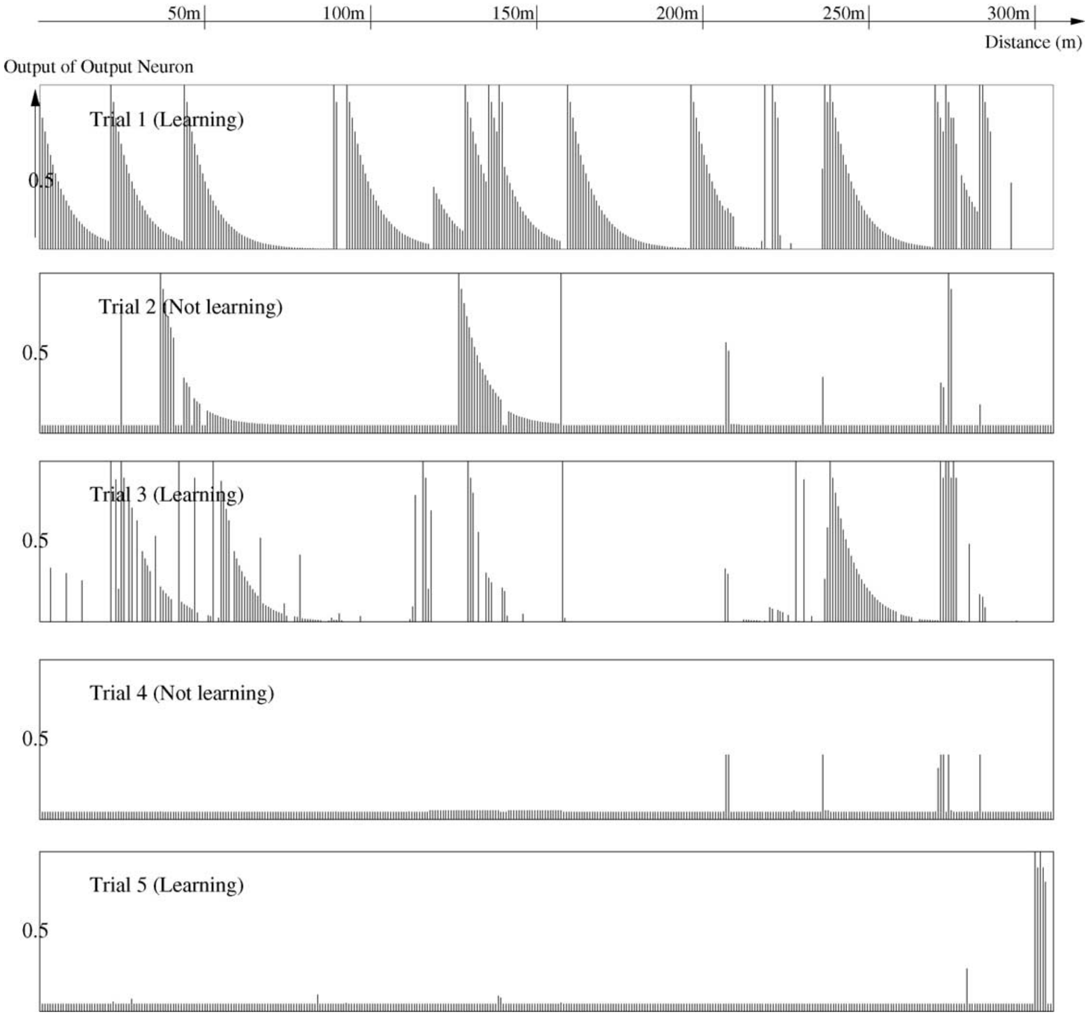

Fig. 7. Plots of the novelty found when the growing novelty filter is trained while the robot explores the entire 300 m loop of corridor around the Computer Science Building.

A new image was then captured, and the contrast of the image improved using histogram equalisation [19]. An input vector for the novelty filter was generated by taking only a subset of the pixels from the image and concatenating them into a vector. One hundred pixels were chosen along a spiral centred at the middle of the image (where the strongest edges are) and running out to the edges of the image.

The experiments then parallelled those described in Section 4.2. The robot travelled through the environment using wall following, stopping every 30 cm to take a picture and producing an input vector, which was fed into the novelty filter. Starting from an initially blank network the robot explored environment A for four learning runs until nothing was found to be novel — an extra learning run was needed, probably because there is significantly more information in the image vector than in the sonar vector. The door on the right of the robot was then opened (environment A*) and the robot again explored. As in the sonar experiments, the only perceptions that were highlighted were those around the doorway. The filter trained in environment A was also tested in environments B and C, and the amount of novelty found in each environment

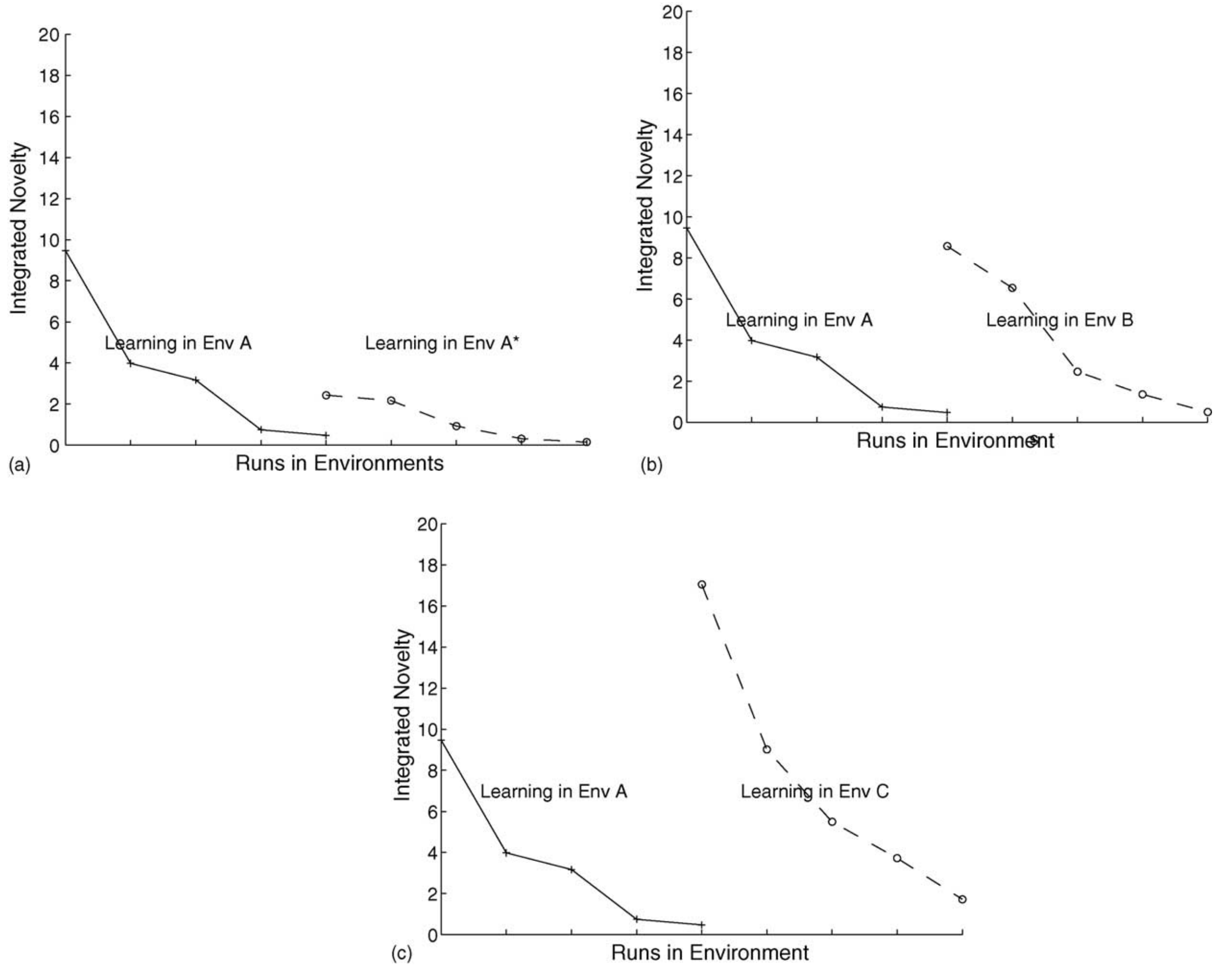

Fig. 8. Graphs showing the novelty filter learning about environments when using inputs from a camera mounted on the robot.

over the testing runs is shown in Fig. 8. It is particularly interesting to note that the novelty filter finds environment C, which is visually very different to the others, very novel, more so than the filter with sonar inputs. This shows that using the camera enables the filter to distinguish details at a much higher resolution.

5. Summary and conclusions

Novelty detection is concerned with recognising inputs that do not fit into the underlying model of a dataset. A novelty detecting robot can act as an inspection agent, examining the environment that it travels through and highlighting those features that it has not seen before, or seen only rarely — potential faults.

This paper presents a method for performing unsupervised on-line novelty detection, so that it is suitable for mobile robots. The method is based on learning to ignore perceptions that have been seen before, so that novel inputs are highlighted. This is achieved by using a neural network clustering algorithm with the addition of synapses connecting each node in the network to an output node. These synapses habituate so that the strength of firing decreases with repeated use.

The algorithm has been demonstrated on the task of mobile robot inspection of corridor environments. The robot performs several learning trips through an environment, and can then highlight deviations from the acquired model, either changes to that environment or differences between a test environment and those it was trained in. The algorithm has been demonstrated in a variety of environments, using inputs from sonar sensors and preprocessed images from a camera mounted on the robot. In all cases it has been shown to work reliably.

Future work will consider the question of how to fuse the information from the different sensors, and investigate how to deal with features that are spread out over time. For example, a robot travelling down a corridor may experience a number of doorways. In general, these will consist of a door jamb followed by the door and then another door jamb. If one of this group of perceptions is not found then this could be a novel feature.

This paper has shown that the concept of novelty detection is a useful methodology for robotic inspection. The experiments have been based in simple human environments. The extension of the work to environments of real application, for example, sewers, is an open one.

Acknowledgments

This work was performed at the Department of Computer Science at the University of Manchester as part of the first author's Ph.D., which was supported by an EPSRC studentship.

References

[1] S. Marsland, Novelty detection in learning systems, Neural Computing Surveys 3 (2003) 157–195.
[2] T. Kohonen, Self-Organization and Associative Memory, third ed., Springer, Berlin, 1993.
[3] A. Ypma, R.P. Duin, Novelty detection using self-organizing maps, in: Proceedings of International Conference on Neural Information Processing and Intelligent Information Systems (ICONIP'97), 1997, pp. 1322–1325.
[4] O. Taylor, J. MacIntyre, Adaptive local fusion systems for novelty detection and diagnostics in condition monitoring, in: SPIE International Symposium on Aerospace/Defense Sensing, 1998.
[5] R. Brooks, A robust layered control system for a mobile robot, IEEE J. Rob. Autom. 2 (1986) 14–23.
[6] H. Öğmen, R. Prakash, A developmental perspective to neural models of intelligence and learning, in: D.S. Levine, W.R. Elsberry (Eds.), Optimality in Biological and Artificial Networks? Lawrence Erlbaum Associates, Hillsdale, NJ, 1997, pp. 363–395, Chapter 18.
[7] S. Grossberg, A neural theory of punishment and avoidance. II. Quantitative theory, Math. Biosci. 15 (1972) 253–285.
[8] G.A. Carpenter, S. Grossberg, The ART of adaptive pattern recognition by a self-organising neural network, IEEE Computer 21 (1988) 77–88.
[9] R. Bogacz, M.W. Brown, C. Giraud-Carrier, High capacity neural networks for familiarity discrimination, in: Proceedings of the International Conferebnce on Artificial Neural Networks, 1999, pp. 773–778.
[10] P. Crook, S. Marsland, G. Hayes, U. Nehmzow, A tale of two filters-on-line novelty detection, in: Proceedings of the International Conference on Robotics and Automation (ICRA'02), 2002, pp. 3894–3900.
[11] S. Marsland, J. Shapiro, U. Nehmzow, A self-organising network that grows when required, Neural Networks 15 (8–9) (2002) 1041–1058.
[12] S. Marsland, On-line novelty detection through self-organisation, with application to inspection robotics, Ph.D. thesis, Department of Computer Science, University of Manchester, 2001.
[13] M. Cottrell, J. Fort, G. Pages, Theoretic aspects of the SOM algorithm, in: Proceedings of Workshop on Self-Organising Maps (WSOM'97), 1997, pp. 246–267.
[14] B. Fritzke, A growing neural gas network learns topologies, in: G. Tesauro, D.S. Touretzky, T.K. Leen (Eds.), Advances in Neural Information Processing Systems 7 (NIPS'94), MIT Press, Cambridge, MA, 1995, pp. 625–632.
[15] A. Baraldi, E. Alpaydin, Constructive feedforward ART clustering networks-part II, IEEE Trans. Neural Netw. 13 (3) (2002) 662–677.
[16] A. Baraldi, P. Blonda, A survey of fuzzy clustering algorithms for pattern recognition: Part II, IEEE Trans. Systems, Man Cybernetics-Part B: Cybernetics 29 (6) (1999) 786–801.
[17] J.D.R. Millan, D. Posenato, E. Dedieu, Continuous-action Q-learning, Machine Learning 49 (2–3) (2002) 247–266.
[18] A. Kurz, Constructing maps for mobile robot navigation based on ultrasonic range data, IEEE Trans. Syst. Man Cybernetics-Part B: Cybernetics 26 (2) (1996) 233–242.
[19] R.C. Gonzalez, P. Wintz, Digital Image Processing, second ed., Addison-Wesley, Reading, MA, 1987.

Stephen Marsland is a lecturer in the Institute of Information Sciences and Technology at Massey University, New Zealand. He has a degree in Mathematics from Somerville College, University of Oxford and obtained his PhD in Machine Learning from Manchester University in 2001. He worked as a postdoctoral researcher and lecturer at Manchester University before emigrating to New Zealand to take up his current post in 2004. Stephen's research interests

are in applying mathematical methods to problems in machine learning, image analysis, and related problems. He holds a Royal Society of New Zealand Marsden grant to study the non-rigid registration of groups of medical images using diffeomorphic warps, and is also carrying out work in complex systems modelling of healthcare choices, particle dynamics of the Euler equations, and bioinformatics.

Ulrich Nehmzow is reader in Analytical and Cognitive Robotics in the Department of Computer Science at the University of Essex. He obtained his Diploma in Electrical Engineering and Information Science from the University of Technology at Munich in 1988, and his PhD in Artificial Intelligence from the University of Edinburgh in 1992. After a postdoctoral position at Edinburgh University he became Lecturer in Artificial Intelligence at the University of Manchester in 1994, and Senior Lecturer in Robotics at the University of Essex in 2001. His research interests comprise autonomous mobile robotics, with emphases on robot learning, robot navigation, simulation and modelling and novelty detection for industrial applications of robotics. His current research focuses on scientific methods in mobile robotics, using chaos theory and dynamical systems theory to obtain quantitative measures of robot–environment interaction. He is the co-chair of the British conference on mobile robotics (TAROS), member of the editorial board of *Journal of Connection Science* and the *AISB Journal*, and member of the Engineering and Physical Sciences Research Council College for peer review.

Jonathan Shapiro got his PhD in Physics from UCLA in 1986. He is currently a senior lecturer in Computer Science of the University of Manchester, and has also been associated with the Santa Fe Institute, where he directed the Complex Systems Summer School in 2003 and co-directed of the Budapest Complex Systems Summer School in 2002. He co-developed the statistical mechanics approach for analysing evolutionary algorithms and has worked on the theoretical aspects of reinforcement learning, human verbal memory, genetic algorithms, artificial neural networks, robot navigation and animal learning.