A Tale of Two Filters - On-line Novelty Detection

Paul A.Crook[1], Stephen Marsland[2], Gillian Hayes[1] and Ulrich Nehmzow[3]

[1]Division of Informatics, University of Edinburgh, 5 Forrest Hill, Edinburgh EH1 2QL, {paulc,gmh}@dai.ed.ac.uk

[2]Department of Computer Science, University of Manchester, Oxford Road, Manchester M13 9PL, smarsland@cs.man.ac.uk

[3]Department of Computer Science, University of Essex, Colchester CO4 3SQ, udfn@essex.ac.uk

Abstract— **For mobile robots, as well as other learning systems, the ability to highlight unexpected features of their environment – novelty detection – is very useful. One particularly important application for a robot equipped with novelty detection is inspection, highlighting potential problems in an environment. In this paper two novelty filters, both of which are capable of on-line and off-line novelty detection, are compared for two robot inspection tasks, one using sonar and the other camera images. The benefits and problems of using each of the filters are discussed and demonstrated.**

I. Introduction

Inspection tasks, where a mobile robot is trained to recognise potential faults in an environment and then highlight them, is a very useful mobile robot application. However, it is difficult to ensure that the robot recognises every potential fault, since the appearance of a fault will vary widely in different circumstances. One way to avoid this problem is to train the robot to recognise 'normality', that is, those situations where there are no faults, and then require the robot to highlight those perceptions that do not fit the acquired model. This is a question of novelty detection. This approach is more likely to make errors in highlighting places that do not have any faults, rather than missing potential faults. While this bias towards false negatives means that some time is wasted in inspecting places that do not turn out to have faults, it also means that faults are not missed.

This paper describes and compares two novelty filters that are capable of on-line learning. On-line learning has the benefit that perceptions that were left out of the robots training, or were not detected by the robot's sensors during training, can be learnt later without presenting all the data to the network again. On-line learning also allows staged training of the novelty filters, with testing after each stage of learning to see which features are still found to be novel, and further training concentrating on those features.

Two robot inspection tasks are used to compare and contrast the filters, the first considers a robot travelling through a series of corridors and sampling them via its sonar sensors, while the second is made of a simple 'image gallery'. Both filters reliably learn models of the robot's perceptions and detect novelty with respect to that model.

II. The Novelty Filters

A. A Growing Novelty Filter (The GWR Network)

A novelty filter should only pass through inputs that have not been seen before, or seen only rarely. The filter should respond strongly the first time that a stimulus is seen, when it is novel, but as the stimulus is seen more often the filter should stop highlighting that feature. One way that animals do this is a process known as habituation. Habituation is a decrement in response to a stimulus that is seen repeatedly without ill effects [1]. Synapses that habituate respond strongly to a stimulus when it is first seen, but reduce their response as the synapse is repeatedly stimulated. As the level of habituation is a continuously varying quantity, the amount of novelty in a perception is quantified as a number between 0 and 1, with 1 meaning that the perception has never been seen before, and 0 signifying that similar perceptions have been seen frequently.

A novelty filter can therefore be made up of a set of habituating synapses and an unsupervised neural network that classifies the current input and acquires a model of typical perceptions. The choice of neural network is application-specific; one network that has been used in previous work [2] is the Self-Organising Map. However, in on-line inspection of middle-scale environments (containing more perceptions than those described in this paper, see [3]) it became apparent that the network used to classify the input needed to be capable of continuous learning. An investigation of networks in the literature suggested that none were particularly suitable. For this reason a new unsupervised growing network was developed, termed the 'Grow When Required' (GWR) network. It is this network that is used in the work reported here.

The network, the algorithm for which is given in algorithm 1, and which is described in more detail in [3], starts off with a very small network and adds nodes into the map space whenever the current input is found to be novel, as evaluated by the activity of nodes in the network. A threshold to define this novelty, the insertion threshold a_T, is used for this purpose. Neighbourhood connections between nodes that represent similar perceptions are maintained, which means that the network is perfectly topology-preserving.

Algorithm 1 Algorithm of the GWR-based novelty filter

1. Generate a data sample $\boldsymbol{\xi}$ for input to the network.
2. Select the two best-matching nodes; $s = \arg\min_{c \in A} \|\boldsymbol{\xi} - \mathbf{w_c}\|$ and $t = \arg\min_{c \in A \setminus \{s\}} \|\boldsymbol{\xi} - \mathbf{w_c}\|$; where A is the set of nodes in the network, and $\mathbf{w_c}$ is the cth node's weight vector.
3. Calculate activity of the selected nodes $a_i = \exp(-\|\boldsymbol{\xi} - \mathbf{w_i}\|)$.
4. If there is not an edge between s and t, create it, otherwise set the age of the connection to 0.
5. If the activity of node s is $a_s < a_T$ and the habituation is $h_s(t) > h_T$ (for pre-defined thresholds a_T, h_T) insert new node:
   - Add the new node, r, with weights $\mathbf{w_r} = (\mathbf{w_s} + \boldsymbol{\xi})/2$.
   - Insert edges between r and s and between r and t.
   - Remove the edge between s and t.
6. Adapt weights of the winning node, s, and neighbours, i, using learning rates $0 < \epsilon_n < \epsilon_b < 1$; $\Delta\mathbf{w_s} = \epsilon_b(\boldsymbol{\xi} - \mathbf{w_s})$; $\Delta\mathbf{w_i} = \epsilon_n(\boldsymbol{\xi} - \mathbf{w_i})$.
7. Age edges with an end at s.
8. Habituate the winning node and its neighbours using $\tau \frac{dh_i(t)}{dt} = \alpha \left[h(0) - h_i(t) \right] - S(t)$; initial habituation level $h(0)$, stimulus strength $S(t) = 1$, parameters α, τ.
9. Check if there are any nodes or edges to delete.
10. Novelty output = habituation of the winning node, $h_s(t)$.

B. A Novelty Filter based on Hopfield Network Energy

Hopfield [4] recognised that, as well as providing content addressable memory, auto-associative networks could indicate the familiarity of patterns, i.e., their similarity to patterns stored in the network. He proposed that this could be achieved by monitoring the rate at which neurons change state as the network was allowed to relax. This work was extended [5] by showing that the 'energy' of a Hopfield auto-associative network indicates the familiarity of a pattern – patterns with lower energies are generally familiar, those with higher energies are generally novel.

Calculating the energy of a Hopfield network has advantages over the method outlined by [4] as it is a simple algorithm whose execution time is fixed irrespective of the number of patterns stored. New patterns can be learnt by a Hopfield network in a fixed execution time and without presenting previous data again. Thus, a novelty filter based on the energy of a Hopfield network is ideal for on-line operation. The ability of the Hopfield network to retrieve previously learnt patterns from partial cues is sacrificed, but the benefit of this tradeoff is that it can classify significantly more patterns than an equivalent Hopfield network can typically recall. The algorithm is shown in algorithm 2, and further details are given in [5] and [6].

Novelty detection is achieved by checking the energy (E) of a pattern against some threshold. It is possible to show [5] that the energy for a pattern which has been learnt by the Hopfield network is $-\frac{N}{2}$ plus a noise term (where N is the number of neurons in the network) and the energy for a novel random pattern is zero plus a similar noise term. Based on this a threshold of $E < -\frac{N}{4}$ is typically used for classification of patterns.

Algorithm 2 Algorithm of the Hopfield-based novelty filter

1. Initiate network with weights $w_{ij} = 0$, where w_{ij} is the weight between the ith and j^{th} neurons; $1 \leq i \leq N$, $1 \leq j \leq N$, network is a fully connected network of N neurons.
2. Generate a data sample $\boldsymbol{\xi}$ which is a binary vector N bits long for input to the network. The ith element of $\boldsymbol{\xi}$ is $\xi_i \in \{-1, +1\}$.
3. Compute energy of network; $E = -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} \xi_i \xi_j$
4. Compare with threshold for familiarity; if $E < -\frac{N}{4}$ classify pattern as familiar, otherwise novel.
5. If learning enabled and pattern classified as novel then update network weights; $w_{ij} \leftarrow w_{ij} + \frac{1}{N} \xi_i \xi_j$ if $i \neq j$.

B.1 Encoding

In general, the function of any novelty filter is determined by the *representation* of the data and the *metric* used by the novelty filter to determine the distance (or similarity) between data points. If the metric used by the novelty filter is predetermined then the selection of representation becomes critical to the measurement of similarity.

The Hopfield network uses binary encoded patterns, which is a problem for continuous valued data, such as the sonar data used in section III-A. Such data has to be encoded into patterns of binary strings. The similarity metric used by the Hopfield novelty filter is the Hamming distance between two binary strings. Thus, the choice of encoding has to be carefully selected so that the required relationship between the data points is represented in terms of Hamming distance[1]. For sonar data the main relationship between sonar readings is that each reading is a measure of distance, thus one possible sensible encoding is one in which an increase in the distance measured by one sonar sensor is represented as a monotonic increase in the Hamming distance of that reading. Details of the encodings tried are given in section III-A.

III. EXPERIMENTS

Two different sets of robot experiments were used to compare the two novelty filters described in the previous section. The first set considered detecting environmental novelty in a set of corridor environments using sonar scans, while the second attempted to detect novelty in simple images taken by a camera as the robot travelled along an 'image gallery' of sheets of paper mounted on a wall. In both sets of experiments the results for each of the filters were collected independently. Both tasks are relatively simple, the reasons for this being to ensure that the comparison is between the filters and not any other experimental issues, such as the differing path of the robot as it travels round corners, or issues associated with com-

---

[1]The determination of whether the similarity measure is sensible depends upon the human observers' understanding of the relationships present in the data.

puter vision techniques for preprocessing of images. In the case of the vision experiments it also allowed straightforward control over the amount of change in the scene. Both of the experiments show simplified parts of a robotic inspection task where the robot must learn about an environment and then detect deviations from the acquired model.

The two sets of experiments employ a similar methodology. Initially both filters are untrained. Alternating 'learning' and 'non-learning' runs are then made through the environment. During a learning run a perception that is regarded as novel is added immediately to that filter's model of the environment. During a non-learning run each filter highlights the stimuli that are still perceived as novel, but without learning them. Once the filters have learnt a model of the environment, so that the majority of the perceptions are no longer classified as novel, the environment is modified or the robot is moved to a new environment. Each perception from this new environment is then evaluated for novelty with respect to the model of the previous environment. The concept behind this methodology is to demonstrate the ability of the novelty filters to: (i) initially learn an environment, (ii) recognise either changes in an environment or differences between the two environments, (iii) learn the differences found so that they are no longer regarded as novel.

A. Environmental Novelty in Sonar Scans

In these first experiments, the robot travels through 10 m sections of corridor using a wall-following behaviour. It takes continuous sonar scans of its surroundings using the 16 sonar sensors that are arranged in a ring around the robot. Every 10 cm the average of the scans over the last 10 cm is presented to the novelty filter, which evaluates how novel that perception is with respect to the model acquired so far.

The two novelty filters require different encodings of the inputs. Inputs to the GWR-based novelty filter consist of a vector of continuously valued real numbers. The filter classifies the input vector according to how similar it is to previously seen inputs, as measured by the Euclidean distance between the input and the weights of the network nodes. Each of the 16 sonar readings was used as one element of the input vector, and this vector was normalised.

In contrast, the Hopfield novelty filter requires the input to be encoded as a binary string (see section II-B.1). Several encoding schemes were tried, a binary code, Gray code and thermometer code, with variation of both the number of bits used and the rescaling of the sonar data values. Experiments showed that the thermometer encoding has the best performance[2].

---

[2]The level of performance determined by the human operator

Thermometer coding perfectly preserves the relationship between Euclidean distance and Hamming distance. Values are encoded by the length of a continuous string of '1's, e.g., 2 is encoded as 00011, 3 encoded as 00111, 4 as 01111 and so on (so that the length of the 'bar' of 1s shows how large the number is, as a column of mercury does in a thermometer). Thus, the Hamming distance between two inputs represented as thermometer codes will be identical to the Euclidean distance between the same two readings.

The use of thermometer encoding raises problems with the selection of a suitable threshold for the detection of novelty, because the typical threshold assumes that the range of patterns to be classified are distributed evenly through the binary input space. However, thermometer encoding is a very spatially inefficient coding, as a 256 bit string can represent only 257 possible values (0 to 256), out of the total 2^{256} possibilities. This means that a significantly lower threshold than usual is needed. The threshold was determined empirically based on the best obtainable performance on the sonar data.

For the novelty filter based on the GWR network, an insertion threshold (see section II-A) of $a_T = 0.7$ was used. Values of this threshold between 0.6 and 0.8 produced similar results.

A.1 Results

Figure 1 demonstrates the two filters learning about an environment without any prior information. A schematic of the environment is shown at the top of the figure, and then the output of both the novelty filters are shown below, with the GWR-based novelty filter being shown as the continuous curve (peaks are novel) above the spikes of the Hopfield novelty filter, for which each perception is either novel or not-novel, depending upon whether or not it exceeds the threshold. Five runs are shown. During the first, third and fifth, the filters are learning, and during the second and fourth learning is disabled so that the areas that require further attention can be clearly seen. It can be seen that the area around the doorway on the right of the robot is found to be novel, and that in general the Hopfield novelty filter has more problems with noise, as is shown particularly by the extra novelty in the second and third runs. Novelty observed at the start of each of the runs is due to the robot adjusting its orientation and position relative to the wall.

Figure 2 shows what happens when the novelty filter that has been trained in the first environment explores the same environment, but with the door in the wall to the right of the robot opened. It would be expected

---

examining the environment and deciding if there are any novel features at the point that is flagged novel by the filter.
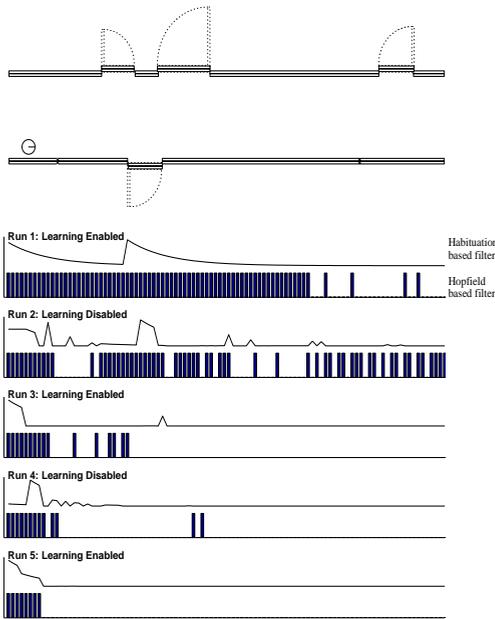
Fig. 1. The two filters learning about an environment, initially without any prior training. On each of the five plots above the output of the GWR-based novelty filter is shown above the output of the Hopfield novelty filter. Both filters start off finding every perception novel, but learn an accurate model fairly quickly.
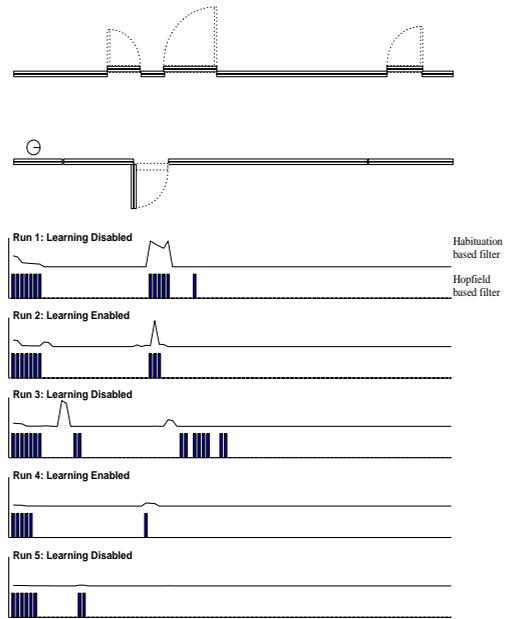

Fig. 2. After learning about the environment shown in figure 1 the filters perceive the same environment, but with a change made, a door being opened. Only the area around the now-open door is found to be novel.

that the perceptions of the rest of the environment would not be found to be novel, but that of the open doorway would. The spikes in the graphs demonstrate that this is indeed what happens, although on the third run it can be seen that there is some novelty early on in the run, which is detected by both filters. This is caused by some perceptions of a crack in the wall at this point, which is spotted only rarely, but causes a large jump in sonar readings when it is seen. The novelty detected by the Hopfield novelty filter just beyond the open doorway on this third run is surprising.

Figure 3 demonstrates the effects of putting the novelty filters trained in the first environment into a new environment, a very similar corridor environment. It can be seen that the areas around the doorways are found to be novel by both filters in figure 3, and that the GWR-based novelty filter also finds the perceptions at the end of the environment to be novel. The doorways in this environment are more deeply inset than those in the first, and at the end of the environment a number of boxes project from the wall, which is not seen anywhere else. It appears that this is not a sufficiently different perception for the Hopfield novelty filter.

B. An Image Gallery

Similar to the sonar experiment above, a robot travels the length of a wall using a simple wall-following algorithm. Along this wall a 'gallery' of orange rectangular sheets of card are placed. Mounted on the robot is a colour camera that looks directly at this gallery. As the robot travels along the wall images are captured from the video camera and processed to produce a 48×48 bit binary image (or 2304 bit binary string), where each bit signifies the presence or absence of orange in that pixel. This input is then presented to the two novelty detection filters.

As the images produced are binary no further encoding of the data is required for either novelty detection model. The 2304 bit binary string is regarded as a vector with 2304 entries for input to the GWR-based novelty filter.

The input patterns are theoretically evenly distributed through the input space, thus the threshold level of $E < -\frac{N}{4}$ suggested in section II-B is used for the Hopfield novelty filter. An insertion threshold of $a_T = 0.9$ was used for the GWR-based novelty filter.

B.1 Results

The novelty filters start with no prior information about the environment. Learning was enabled in run one in figure 4, and during the second runs in both figures 5 and 6. It was disabled during the remainder.
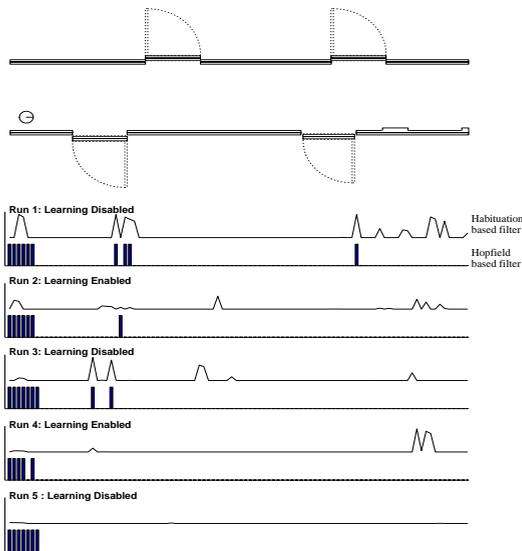
Fig. 3. The filter trained in figure 1 is used to evaluate another, similar, environment. Only a few features are found to be novel, the doorways, which are more deeply inset, and, for the GWR-based novelty filter, some boxes on the wall at the end of the environment.
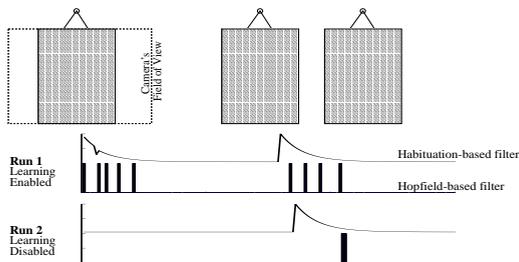

Fig. 4. The novelty filters (GWR-based novelty filter above, Hopfield novelty filter below) explore an image gallery. The dotted box shows the extent of the camera image, the x-axis of the graph corresponds to the middle of the box. Both filters find initial perceptions of the gallery novel and then the appearance of the narrow gap between the two rectangles on the right.

The results of the first exploration are shown in figure 4, again with the output of the GWR-based novelty filter above that of the Hopfield novelty filter. The perceptions found novel are the initial perceptions of the gallery, and the appearance in the camera's field of view of the narrow gap between the last two squares of card. Both filters show similar results. In the second run along the gallery it can be seen that most of the perceptions are regarded as familiar by both models apart from the region where the narrow gap between the last two cards comes into the image, provoking a reaction from the GWR-based novelty filter just as it comes into view, and from the Hopfield novelty filter once it occupies the centre of the camera's field of view.

In figure 5 both filters having been trained on the first gallery, viewed a gallery where the second picture was modified. The change was found to be novel by both filters during run one, was learnt during a single pass (run two) and the gallery was then perceived as completely familiar during run three.
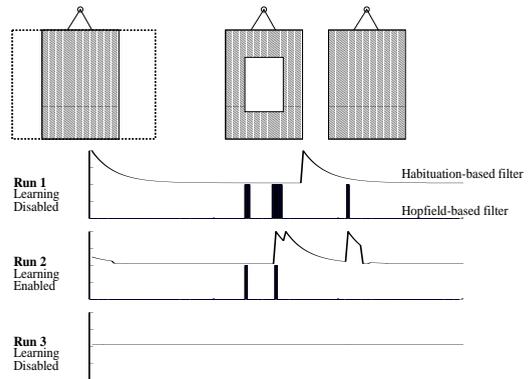

Fig. 5. After learning the first gallery, a change is made to one of the pictures. Both filters find this to be novel, and both learn about it during a training run.

Figure 6 shows the results with a further change to the gallery. The spikes in novelty start at the position where the leading edge of the final picture, which has been changed, are first seen by the camera. Only the perceptions relating to this change are found to be novel and it is learnt successfully during the second run. Finally, in figure 7, a further change is made to the gallery. The first of the three pictures is changed so that it contains a white square that is larger than the one that was added to the second picture. The Hopfield novelty filter does not find this to be novel because it is not sufficiently different from the second picture, but the GWR-based novelty filter does find it novel. Two spikes are seen; at the beginning of the run when the changed picture is first seen, and at the place where the edge of the second card comes into view.

IV. DISCUSSION AND CONCLUSIONS

The previous section has shown that both the GWR-based and Hopfield novelty filters can reliably learn a model of an environment during exploration by a robot and then detect novel features of further environments. It has also been demonstrated that the two filters are capable of continuous learning.

The problem of input encoding has been discussed in detail. The GWR-based novelty filter represents data as a vector of continuously valued entries, which is more flexible than the binary representation required by the Hopfield network. Experience in representing continuously valued data using binary strings has emphasised the importance of the correct selection of rep-
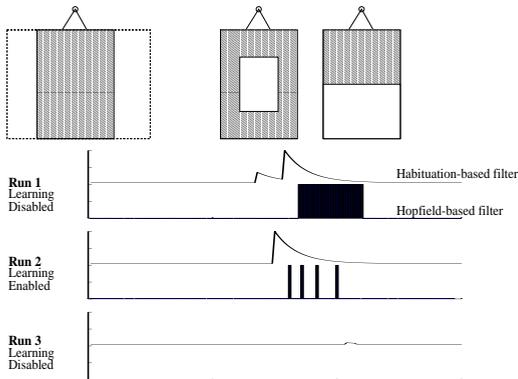
Fig. 6. The filters trained in figures 4 and 5 when a further change is made to the gallery. Both detect the change correctly.
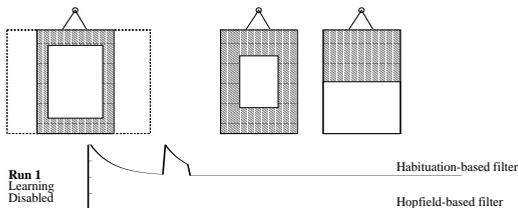

Fig. 7. A final change, which only the GWR-based novelty filter finds to be novel. The perceptions are too similar for the Hopfield novelty filter to be able to differentiate between the two.

resentation and matching this to the metric used to determine similarity, in order to obtain correct function of a novelty filter.

The only important parameter that the user needs to set for the GWR-based novelty filter is the insertion threshold a_T, which has a strong control over the amount of novelty that is found in a set of perceptions. While there are other adjustable parameters built into the model, in practice these parameters are not crucial. When the novelty filter is based on the GWR network the size of the network is not a parameter, because new nodes are added into the network whenever they are required.

The main parameter that can be varied with the Hopfield novelty filter is the threshold energy used to judge novelty. Theoretically, provided that the data is spread uniformly through the binary pattern space this threshold should be set at $\frac{N}{4}$ (where N is the number of bits in each input pattern). However, if this assumption is broken in the encoding of the data, it is necessary to use a much lower level to obtain reasonable performance of the network. The network showed limitations in its sensitivity compared to the GWR-based model in the above experiments, even in the experiments with sonar scans, where the threshold has

been tailored by hand. Network size and learning capacity of the Hopfield novelty filter are linked to the representation of the input data. Input data with N bits requires a network with N nodes and the number of examples that can be learnt is proportional to N^2 [5]. Although this proved no problem in the above experiments, it could prove a limiting constraint on the Hopfield filter's usefulness as the number of familiar patterns grows.

Overall, both novelty filters appear to behave reasonably on the experiments conducted in this paper. The GWR-based novelty filter appears to have some advantages over the Hopfield novelty filter in these tasks because of the ease with which the data can be represented, its slightly higher level of sensitivity being coupled with a more robust performance on noise. One reason for this is the different way that the data is represented by the two networks. The Hopfield network stores memories in such a way that parts of separately learnt perceptions can be recalled together, creating a spurious memory. A perception matching this spurious combination will be classified as familiar even though it is not. In contrast, the GWR network compares the distance between the input and network nodes that represent separate clusters of the previously seen perceptions, thus a novel perception that is a combination of parts of previously perceived perceptions would still be found novel.

REFERENCES

[1] R.F. Thompson and W.A. Spencer, "Habituation: A model phenomenon for the study of neuronal substrates of behaviour," *Psychological Review*, vol. 73, no. 1, pp. 16–43, 1966.

[2] Stephen Marsland, Ulrich Nehmzow, and Jonathan Shapiro, "Novelty detection on a mobile robot using habituation," in *From Animals to Animats: Proceedings of the 6th International Conference on Simulation of Adaptive Behaviour (SAB'00)*. 2000, pp. 189 – 198, MIT Press.

[3] Stephen Marsland, *On-line Novelty Detection Through Self-Organisation, With Application to Inspection Robotics*, Ph.D. thesis, Department of Computer Science, University of Manchester, 2001.

[4] J.J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," in *Proceedings of the National Academy of Sciences*, USA, 1982, vol. 79, pp. 2554–2558.

[5] Rafal Bogacz, Malcolm W. Brown, and Christophe Giraud-Carrier, "High capacity neural networks for familiarity discrimination," in *Proceedings of the International Conference on Artificial Neural Networks (ICANN'99)*, 1999, pp. 773 – 778.

[6] Paul Crook and Gillian Hayes, "A robot implementation of a biologically inspired method for novelty detection," in *Proceedings of Towards Intelligent Mobile Robots (TIMR'01)*, 2001.