Location Recognition in a Mobile Robot Using Self-Organising Feature Maps[1]

Ulrich Nehmzow, Tim Smithers and John Hallam

Department of Artificial Intelligence
University of Edinburgh
5 Forrest Hill
Edinburgh EH1 2QL
SCOTLAND

E-mail: ulrich@uk.ac.ed.edai

Abstract

Self-organising structures are a dominant feature of the experimental mobile robots built in our "Really Useful Robots" project. This paper continues where [Nehmzow & Smithers 90] finished. It explains some initial experiments using self-organising feature maps, and how those maps can be used by a mobile robot to recognise locations in its environment. This location recognition capability is achieved without using sensory information. Instead information derived from the motor actions of the robot is used, and shown to be sufficient.

1 Introduction

Self-organising feature maps (SOFMs), [Kohonen 88], can be used for simple navigation tasks. In earlier work, [Nehmzow & Smithers 90], we have shown that "Alder", the first of the "Really Useful Robots" (RUR) (see figure 1), is able to use a SOFM to recognize particular locations in a simple enclosure after it has had sufficient

[1]Reprint from G. Schmidt (ed.), Information Processing in Autonomous Mobile Robots, Springer Verlag 1991.

time to explore and to "learn" about this enclosure. In these earlier experiments the input that was presented to the SOFM was derived from sensor signals and contained explicit information about "landmarks" that the robot encountered, and the distance travelled between them. These landmarks were the concave and convex corners in the robot's enclosure.

Figure 1: Alder, the first of the "Really Useful Robots".

In the RUR project, we are investigating ways to make mobile robots more flexible, and in particular, to make them more able to cope with *unforeseen situations*. We believe that in order to achieve this as many decisions as possible should be left to the robot; in particular decisions about what and how information derived from sensors or other sources internal to the robot should be interpreted for the purposes of control in task achieving behaviour. For the same reason we aim to equip the robot with as little predefined knowledge about its environment as possible [Nehmzow *et al.* 89]. Information concerning landmarks is specialised information. Therefore, we decided to try to achieve similar results to those reported earlier, but this time using less specific input information. Whereas before we used processed sensor information (denoting corner types) to achieve location recognition, we now use *no sensor information* at all! Instead we use information about the history of the *motor action commands* of the robot controller. In this paper we describe how we achieved this, again using self-organising networks.

2 The Duality of Sensing and Acting

Sensing and acting are typically treated as separate functions in robotics. We believe, however, that sensing and acting are two aspects of the same function, and that they therefore cannot be successfully analysed in isolation. The actions of a robot, just like those of a person, determine to a large extent the sensory signals it will receive, which will in turn influence its actions. Breaking this tight interaction into two separate functions leads, we believe, to an incorrect decomposition of the robot control problem. While particular features of a robot's sensors and actuators do play an important role in determining its performance, these effects cannot be determined by their separate analysis. Acting and sensing have to be seen together; neither acting nor sensing alone will make the agent succeed.

The input vector we have chosen for the following experiments demonstrates this point: it contains no direct information about sensory input. The information it does contain is derived from the motor action commands of the robot controller, but these, as we have said, are themselves influenced by the sensory signals received by the robot as a result of its actions. We chose to use information derived from the motor action commands of the robot controller, rather than from sensor signals, because they form a smaller set of signal types, they are much less subject to noise, but they still adequately characterise the interactions between the robot and its environment as it seeks to achieve its task—wall follow, in our case.

For the location recognition experiments the robot is placed in an enclosure as shown in figure 2; it then follows the wall using a preprogrammed wall-following and obstacle-avoidance behaviour (alternatively these skills could be acquired by learning, see [Nehmzow *et al.* 89]). In other words, the robot is governed by its preprogrammed wall following behaviour which, of course, does use sensory information. The process of constructing the SOFM is, however, independent of the wall following behaviour, it simply "looks" at the motor action commands issued by the controller as the robot performs this wall following task.

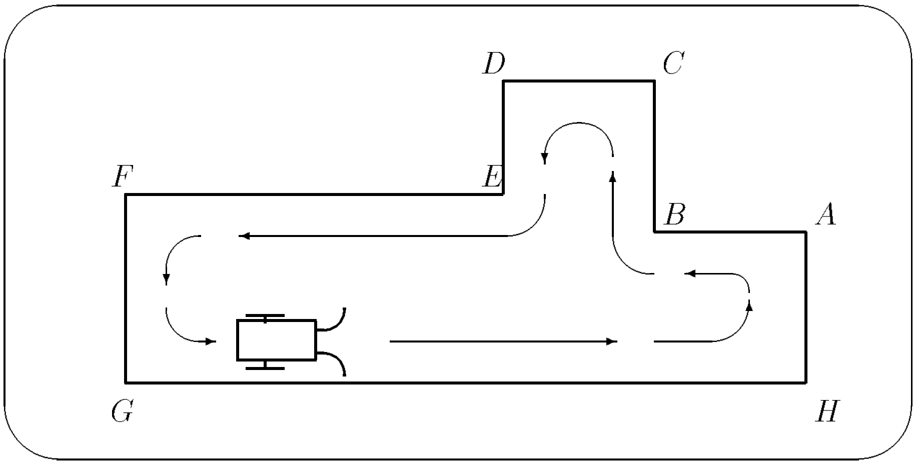

Figure 2: A typical environment for Alder

Every time a new motor action command is issued, that is, every time the wall-following or the obstacle-avoidance behaviour forces the robot to change direction a motor action vector is generated. This is a nine bit vector which contains information about the state the motors were in until this change, and thus the direction the robot has been travelling up to this moment (forward, left or right)[2], as well as information

[2]Strictly speaking, they contain information about the last command from the robot controller.

about how long it was in this state (see figure 3). This motor action vector forms the input to the SOFM.

Motor Action		Duration	
Forward	01 01	00000	less than 0.9s
Left	01 10	00001	0.9 - 1.3s
Right	10 01	00011	1.3 - 1.7s
		00111	1.7 - 2.1s
		01111	2.1 - 2.6s
		11111	over 2.6s

Figure 3: The motor action vector

Thus, from figure 3 we can see that no information concerning sensor signals is directly presented to the SOFM. The only information available to the network concerns motor action commands.

3 Early Experiments

3.1 Self-Organising Feature Maps

The mathematics for the kind of self-organising feature maps (SOFMs) that we use can be found in [Kohonen 88] or in [Nehmzow & Smithers 90]. Basically, they conform themselves to the structure of the space of input signals, clustering input signals in such a way that related inputs excite neighbouring areas of the network (they are *topology preserving*), and so that the relative sizes of these areas reflect the relative probabilities of the different inputs in the data presented. By making use of these two properties of SOFMs they can be used to perform a kind of unsupervised learning.

To investigate how this might be done using only data derived from motor commands we first carried out the following experiment in which we used a two-dimensional SOFM of ten by ten cells.

3.2 Experiments with a 10 x 10 SOFM

The robot was left to explore its enclosure (figure 2) by following the walls, generating input vectors as shown in figure 3 each time a new motor action command was issued by the robot controller for any reason. The following pictures show the response of a ten by ten cell SOFM to different input stimuli; the terms "right", "left" and "forward" describe the motor action command that was performed, followed by a number that indicates the length of time for which this action was performed (see figure 3).

Whether this command was actually obeyed by the robot or not is not sensed.

```
------------       ------------       ------------       ------------
|.       ..|       |  .... ...|       |  .....///|       |   ../////|
|..   .....|       |  /..///..|       |  ..//////|       |  ../+++//|
|//.  .....|       |++/..///..|       |//////////|       |/////+*+//|
|//.  ...  |       |*+/..///  |       |////////..|       |/////+++..|
|///.....  |       |+++/////  |       |//+++///..|       |///+++++..|
|++///..   |       |///////.  |       |//+*+///..|       |.//++++/..|
|*+//.     |       |/////...  |       |//+++//...|       |.//+++////|
|++/..     |       |/////.....|       |./+++/////|       |..////////|
|..        |       |  ........|       |  ..././//|       |  ...///..|
|..        |       |  ........|       |  ....////|       |  ..../...|
------------       ------------       ------------       ------------
Forward, zero      Forward, one       Forward, two       Forward, three
------------       ------------       ------------       ------------
|   ..//+++|       |    .//+++|       |          |       |          |
|  ...+++++|       |  ...//+++|       |          |       |          |
|//../+++++|       |....////+*|       |          |       |..    ..  |
|/////+++//|       |....//////|       |        ..|       |..    ....|
|/////+*+//|       |..///+++//|       |        ..|       |.....  ...|
|..///+++//|       |..///+++//|       |        ..|       |  ........|
|..////////|       |..////////|       |.    .....|       |  ...////.|
|..////////|       |..///./...|       |..........|       |..////////|
|     //...|       |     .....|       |++////..//|       |//+*+/////|
|     .....|       |     .....|       |*+////.///|       |//++++////|
------------       ------------       ------------       ------------
Forward, four      Forward, five       Left, zero         Left, one
------------       ------------       ------------       ------------
|       ...|       |     .....|       |     ..///|       |     ..///|
|     .....|       |     ///..|       |     /////|       |     .////|
|..  ......|       |.. ..///..|       |    ./////|       |    ...///|
|... ....//|       |...../////|       |    .///++|       |    .///++|
|..///...//|       |....//////|       |  ...///++|       |  ...///+*|
|  ///..///|       |  ..//////|       |  ...//+++|       |  ...///++|
|  ////////|       |  ../+++++|       |  .../++*+|       |  ...//+++|
| .///+++++|       |  ///+*+++|       |  /.//++++|       |  .../////|
|/////+++*+|       |..///+++//|       |..////+///|       |...../////|
|/////+++++|       |..////++//|       |..////////|       |...../////|
------------       ------------       ------------       ------------

  Left, two         Left, three        Left, four         Left, five


------------       ------------       ------------       ------------
|*++//..   |       |//+*+//.  |       |///++//...|       |..///++/..|
|++///     |       |//+++...  |       |//+++//...|       |../+++//..|
|  //.     |       |..+//...  |       |..+*+/....|       |../+*///..|
|  ...     |       |..///...  |       |..+++/..  |       |../++///..|
|          |       |.....     |       |..///...  |       |.../////..|
|          |       |   ...    |       |  ///...  |       |  ...//...|
|          |       |   ...    |       |  ///..   |       |  ........|
|          |       |   ...    |       |  ...     |       |  ........|
|          |       |          |       |          |       |     ...  |
|          |       |          |       |          |       |      .   |
------------       ------------       ------------       ------------

 Right, zero       Right, one          Right, two        Right, three
```

```
------------      ------------
|..///+*///|      |...//+*+//|
|..////////|      |..////////|
|  //+/////|      |  ////////|
|  //////..|      |  ////////|
|  ...///..|      |  .../////|
|  ...///..|      |  .../////|
|  ........|      |  ........|
|  ........|      |   .. ..  |
|      .   |      |      .   |
|          |      |          |
------------      ------------

 Right, four       Right, five
```

Figure 4: Excitation patterns of the SOFM for different types of motor action vector inputs. The cell with the largest excitation is indicated by "*", "+" means that the excitation of a cell is higher than 0.9, "/" denotes excitations between 0.9 and 0.7 and "." excitations between 0.7 and 0.5.

Two observations can be made from these pictures of the response of the SOFM to different types of input vector:

- The size of the excited area is roughly proportional to the frequency of occurence of the input signal that caused the excitation.
- Related inputs excite neighbouring areas. In this example, we can see that "forward" movements stimulate the central region of the network, "left" movements stimulate the lower region and "right" movements the top region of the net. Within these basic regions there are variations, depending on the duration of the movement.

Topographic mappings such as these are common in biological nervous systems. The striate cortex, for example (one part of the visual system of primates) is organized in topographic fashion, each half of the visual field (of a macaque monkey) is projected onto the striate cortex in a systematic map [Hubel 79]. [Churchland 86] gives a good survey of topographic mappings in neural systems (for topographic mappings of the visual cortex see, for example, [Allman 77]).

Similarly, somatotopic[3] maps can be found in the cortex. Very similar observations to those stated above can be made about such biological systems, ([Churchland 86]):

- Map distortion is based on the population of neurons representing a particular body area. Areas with a higher density of receptors are bigger. [A higher density of receptors will result in a greater number of sensory signals].

[3]From $\sigma\tilde{\omega}\mu\alpha$, body. What is meant here is that touch, pressure, vibration, temperature and pain sensors are mapped onto the cortex in an orderly fashion.

- A precise mapping of the body surface onto neurons of the somatosensory cortex is found. Neighbourhood relations of body parts are preserved.

If the nature of input stimuli is altered, for example by joining two fingers of a hand together, the mapping in the somatosensory cortex changes accordingly, see ([Clark *et al.* 88]).

The question that arose for us from these observations was: "Can we use this sort of SOFM to achieve location recognition?" In other words, can the mapping of different motor action types onto different regions of the SOFM network, as shown in figure 4, be used in a system for recognising locations? Our tentative answer is that it seems they can be. In the next sections we explain experiments we devised to do this, the system we finally used and the experimental results of the tests we have carried out so far.

3.3 Location Recognition, using one SOFM

Having done these initial experiments with a ten by ten cell SOFM, using the motor action vectors directly as an input to the network, we then tried to combine several of these vectors as input to a network of twelve by twelve cells. The idea was that the response of the network would be correlated with Alder's arrival at a physical location, thus allowing the robot to recognize particular locations. As it turned out, the number of motor action vectors within an input vector that gives best recognition results varies from location to location. Some locations are easy to identify using just a short history (i.e. small numbers of combined motor action vectors), others need a longer history (i.e. a large number of combined motor action vectors). There was no one length of input vector which gives equally good recognition performance with all the corners in the robot's enclosure.

In retrospect, this is not a particularly surprising result. The means by which locations are identified — their "signatures" — are patterns of actions extended over time. As the rate at which motor action commands are issued goes up at corners (obstacle avoidance), the temporal scale of these signatures is dependent upon the distances between corners. In order to be able to identify arbitrary locations it is therefore necessary to use input vectors of different lengths. A fixed size of input vector, whatever it is, will either include extraneous information (complicating the task of the SOFM) or miss possibly crucial signature components (eliminating necessary distinctions).

In some respects, the problem is similar to that encountered in the analysis of natural images in vision: the data is intrinsically multiscalar (though here it is temporal scale that is critical) and so must be subjected to a multiscalar analysis (for this analysis in images see [Marr & Hildreth 80] and [Canny 86]). This motivated our next experiments, in which different sized histories of actions were presented to separate SOFMs in parallel.

4 A SOFM Location Recognition System

As in our earlier work on location recognition (see [Nehmzow & Smithers 90]), the method used is one in which location recognition is based upon the robot recognising its arrival at some location. In other words, location recognition is not based upon recognizing some structural feature of the particular physical location, but on the sequence of actions that have taken place prior to arriving at the particular location. In our earlier work it was the sequence of previous features (corners) that had been detected immediately prior to the current feature (corner) that was used to determine recognition. In this work it is the sequence of motor action commands that have been issued prior to the arrival at the particular physical location that enable Alder to recognize the location. To achieve this we used a system of *seven* independent, two-dimensional SOFMs, working in parallel. Each SOFM consists, as in section 3.3, of a network of twelve by twelve cells. The input vectors to each of these networks are different, but all are built from motor action vectors as shown in figure 3. By combining 2, 4, 6, 8, 12, 16, and 24 of these basic motor action vectors we formed seven SOFM input vectors which correspond to increasingly longer histories of the robots motor action changes, (see figure 5).

The lengths of histories are chosen to cover adequately the expected spectrum of action periodicity. If we think of the sequence of actions generated as the robot circles its enclosure as a pseudo-periodic series, with period roughly equal to the average number of action-vectors generated in a single circuit, then the use of SOFMs tuned to different "frequency bands" allows us to sample the temporal structure of the series across its spectrum and associate these samples with the physical locations whose signatures they are.

In other words, the set of excitation patterns of the seven SOFMs produced as the robot arrives at a particular location (a corner, say) in its enclosure can be used to distinguish this location from all other locations the robot passes through as it wall-follows its way around the enclosure.

4.1 The Experiment

The robot was set to wall-follow its way around the enclosure. Every time a new motor action command was issued as a result of the built in wall-following or obstacle avoidance behaviour of the robot, a motor action vector as described in section 2 was generated. This vector, together with the respective number of previous motor action vectors was presented to each of the seven SOFMs. After a sufficient time, about five times round the enclosure, these feature maps had organized themselves into stable structures corresponding to the topological interrelationships and proportional densities of the input vectors.

After this "learning" period the excitation patterns of all seven networks at a particular location (the *target patterns*) were stored. All subsequent sets of seven excitation patterns generated by new input vectors (*object patterns*) were then compared to the set of seven target patterns. This was done by computing the Euclidean distance (or, alternatively, the city-block distance) between pairs of target and ob-

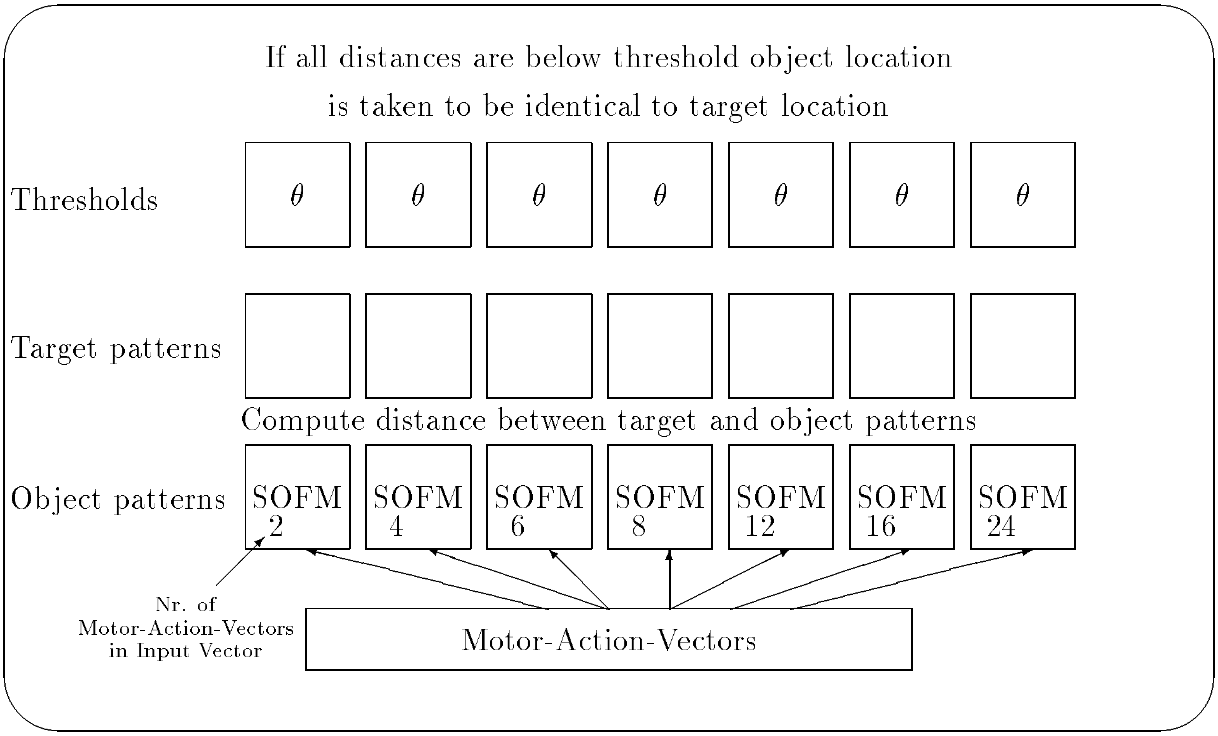

Figure 5: The System used for Location Recognition

ject patterns, see equation 2 in section 4.2. If the distance values between each of the seven pairs of object and target patterns are less than a threshold defined for each pair, the robot is taken to have arrived back at the target location and thus to have "recognized" the previously stored location.

4.2 The Mathematics

In mathematical terms, the system works as follows:

1. Compute the output o_{xyj} of each cell at position (x, y) of each network j:

$$o_{xyj} = \vec{w}_{xy} \cdot \vec{\imath}_j \quad j = 1, ..., 7, \tag{1}$$

   where $\vec{w}_{xy}$ is the individual weight vector of cell(x, y), and $\vec{\imath}_j$ is the input vector to network j.

2. Compute the distance between target pattern and corresponding object pattern:

   (a) Either the Euclidean distance is chosen,

$$d_j^e = \sum_{x=1}^{12}\sum_{y=1}^{12}(o_{xyj} - o_{xyj}^T)^2, \tag{2}$$

   where o_{xyj} is the output value of cell(x, y) of the object pattern j, and o_{xyj}^T is the output value of cell$_{xy}$ in target pattern j.

(b) Or, the city-block distance is computed

$$d_j^{cb} = \sum_{x=1}^{12} \sum_{y=1}^{12} |o_{xyj} - o_{xyj}^T|. \tag{3}$$

3. Determine whether object location and target location are identical:

   If $d_j < \Theta_j$ for all j=1 to 7 then object location and target location are taken to be identical. Θ_j is the threshold for network j. For the experiments reported here the Θ_j were chosen by the designer so that patterns obtained at target location and patterns obtained at other locations could be separated. Note that the actual values will, of course, depend on the distance metric chosen. Schemes can be devised that determine these thresholds automatically, but not without supervision.

4.3 Results

Data recorded from the robot was used to compute these results, however the actual computation was done off-line on a workstation.

In the experiments conducted so far the robot's task was to identify three particular corners in the enclosure shown in figure 2: corners H, E and F. To do this the set of seven excitation patterns of the seven SOFMs at the target corners were stored; all subsequent sets of excitation patterns were then compared to these by computing the Euclidean distances between respective pairs of patterns in the set of seven.

Provided a suitable set of thresholds was used (see paragraph 3 in section 4.2), the robot recognised corner H four times in the subsequent five rounds, and corners E and F in five out of the five times. At no time was a non-target corner erroneously "identified" as a target corner.

4.4 Changes in Parameters

Different Metrics

Instead of using the Euclidean distance to estimate similarity between excitation patterns, the "city-block" distance can be used with identical results (see equation 3 in section 4.2). City-block distance is computationally cheaper than Euclidean distance.

One-dimensional Networks

Instead of using two-dimensional networks, *one-dimensional networks* can also be used. This considerably reduces the computational cost of the scheme. We used one-dimensional networks of twenty-five cells to test this. In the case of corners H and E the results were the same as the ones presented above. In the case of corner F, one erroneous identification took place, as well as the five correct ones as before. This seems to indicate that one-dimensional networks can be used instead of two-dimensional ones in this task, except with slightly less reliability. A plausible conjecture for this is that the neighbourhood relationship plays an important role: each cell in a two-dimensional network has eight neighbours, through which excitation can spread, whereas a cell in a one-dimensional network has but two neighbours. The actual number of cells (twelve by twelve as opposed to one by twenty-five) will have some influence, too; but in our experience this is not as important as the neighbourhood relationships.

5 Summary and Conclusion

5.1 Summary

The robot's task was to recognize particular locations (for example corners) in its world—a simple enclosure. In an earlier publication ([Nehmzow & Smithers 90]) we showed that this task can be accomplished using self-organising feature maps. The input vector used then contained explicit information about landmarks encountered: that the robot was at a corner, what sort of corner (convex or concave) it was, and information about previous corners encountered.

In the subsequent experiments reported here we tried to reduce the explicit information content in the input vector. We also tried to generate input vectors to the network(s) that contain no direct information about sensor signals. The motivation for the first was to avoid using predefined knowledge, because we believe that predefinition of knowledge limits the robot's flexibility; the reason for the latter was to prove our claim that sensing and acting are in fact closely related.

The input vector we chose thus contained information only about the motor commands of the robot controller, and their duration. Vectors put together from varying numbers of these motor action vectors (2,4,6,8,12,16 and 24) were presented to seven separate self organising feature maps, each two-dimensional of size twelve by twelve cells. In order to identify a particular corner, all seven excitation patterns had to be close enough to a stored set of target patterns. If this was the case, the robot was said to have identified the target location.

The location recognition system performed well in this experiment, recognizing corner H (see figure 2) in four out of five times, corners E and F in five out of five times, with no erroneous identifications in either case. Using self-organising feature maps to recognize locations adds a high degree of freedom to the robot controller. The robot is able to build its own representations of the environment, independently of the designer.

5.2 Conclusion

There are two main conclusions we can draw from the work presented. First, our claim that sensing and acting are closely coupled is confirmed by the success of the robot in recognizing locations based upon the sequence of motor activity which leads to arrival there. The "sensor" being used to provide features on which the recognition is based is actually the behaviour of the robot.

Choosing an input vector that contains no direct information about sensory signals makes the system independent of the actual sensors used. Whether tactile, ultrasonic, infrared or other sensors are used: the location recognition system stays the same.

Second, with this approach the features to be identified by the SOFMs are spread out over time. This contrasts with our early work [Nehmzow *et al.* 89] in which the robot controller learns instantaneous reactions to sensor stimuli. The idea, successfully demonstrated here, of using multiple channels to avoid an assumption of a single natural timescale, merits considerable further investigation since, at first sight, it should be capable of wide application to problems of eliciting temporal structure.

5.3 Open Questions and Future Work

We intend to implement a whole system as described above on a mobile robot. This will show us how feasible it is to use this method in applications where real-time processing is important, and where signal noise can bring problems. Implementing such a system on a mobile robot will also enable us to investigate further the questions concerning network size and structure of input vector.

The question of identifying relevant structure over time merits further study and we intend to investigate the limitations of the method used here as well as test alternative approaches.

Acknowledgements

We thank Peter Forster, our colleague on RUR, for his constructive and helpful contributions to this work, and for his comments on earlier versions of this paper.

The work reported here is supported by a grant from the UK Science and Engineering Research Council (grant number GR/F/5852.3). Other facilities were provided by the University of Edinburgh.

References

[Allman 77] John Allman, *Evolution of the visual system in early primates*, Progress in Psychobiology and Physiological Psychology **7** pp.1–53, 1977

[Canny 86] John F. Canny, *A Computational Approach to Edge Detection*, IEEE PAMI **8(6)** pp. 679–698, 1986

[Churchland 86] Patricia Smith Churchland, *Neurophilosophy*, MIT Press Cambridge Mass. and London, England, 1986

[Clark *et al.* 88] Sharon A. Clark, Terry Allard, William M. Jenkins and Michael M. Merzenich, *Receptive fields in the body surface map in adult cortex defined by temporally correlated inputs*, Nature **332** 31 pp. 444–445, March 1988

[Hubel 79] David H. Hubel,*The visual cortex of normal and deprived monkeys*, American Scientist,**67** No 5 pp. 532–543, 1979

[Kohonen 88] Teuvo Kohonen, *Self-Organization and Associative Memory*, Springer Verlag Berlin, Heidelberg, New York, 1988

[Marr & Hildreth 80] David C. Marr and Ellen C. Hildreth, *Theory of Edge Detection*, Proc. R. Soc. London **207**, pp. 187–217, 1980

[Nehmzow & Smithers 90] Ulrich Nehmzow and Tim Smithers, *Mapbuilding using Self-Organising Networks*, in: Jean Arcady Meyer and Stewart Wilson (eds.), *From Animals to Animats*, Proceedings of SAB90 Paris, pp. 152–159, MIT Press Cambridge Mass. and London, England, 1991

[Nehmzow *et al.* 89] Ulrich Nehmzow, John Hallam and Tim Smithers, *Really Useful Robots*, in: T. Kanade, F.C.A. Groen and L.O. Hertzberger (eds.), *Intelligent Autonomous Systems*, Proceedings of IAS 2, pp. 284–293, ISBN 90-800410-1-7, Amsterdam 1989