Presented at the AAAI-2001 Spring Symposium on Learning Grounded Representations.

Toward Learning the Causal Layer of the Spatial Semantic Hierarchy using SOMs

Jefferson Provost and **Patrick Beeson** and **Benjamin J. Kuipers**

Artificial Intelligence Laboratory[*]
The University of Texas at Austin
Austin, TX 78712
{jp,pbeeson,kuipers}@cs.utexas.edu

Abstract

The Spatial Semantic Hierarchy (SSH) is a multi-level representation of the cognitive map used for navigation in large-scale space. We propose a method for learning a portion of this representation, specifically, the representation of *views* in the causal level of the SSH using self-organizing neural networks (SOMs). We describe the criteria that a good view representation should meet, and why SOMs are a promising view representation. Our preliminary experimental results indicate that SOMs show promise as a view representation, though there are still some problems to be resolved.

Introduction

For a mobile robot to be able to navigate, it must know where it is. More specifically, if a robot is to be able to navigate an environment larger than the robot's sensory horizon, it must store some representation of its large-scale environment, i.e. a cognitive map. Given some sensor input, the robot must be able to identify its position in the cognitive map in order to plan routes or monitor the progress of navigation.

Our model of the cognitive map is the Spatial Semantic Hierarchy (SSH) (Kuipers, 2000). The SSH is a system of layered representations, each abstracting the details of the layers below it. From the bottom up, the layers are: the *control layer*, dealing with raw sensor input and direct motor control of the robot; the *causal layer*, consisting of control actions and the effects they have on sensor views, the *topological layer* consisting of places connected by paths, and finally the *metrical layer* in which the places and paths of the topological layer are annotated with metrical information, to form a “patchwork metrical map.”

People and animals learn to navigate using data gathered from interacting with the world. We feel that a robot should be able to do the same. To this end, we and other members of our lab are working on systems which can learn all the levels of the SSH. This paper describes preliminary investigations into using self-organizing neural networks to represent sensor views in the causal layer of the SSH.

The SSH Causal Layer

The *causal layer* of the SSH deals with sensor views and control actions. It consists of a graph whose nodes are distinct sensor views, and whose edges are control actions which cause a transition from one sensor view to the next. The edges of the graph form causal triples of the form $\langle V, A, V' \rangle$, indicating that action A, taken in the context of sensor view V, can produce view V'.

In order for a robot to use the SSH causal layer, the view representation must have these qualities:

- Views must be *indexable* in some fashion using current sensor data. A robot must be able to recognize its current sensor input as one of its set of known views, or a new view.
- Views must *cluster sensory images*. Sensory images are affected by sensor noise and reasonably large amounts of positional noise from the execution of imprecise and uncertain control laws. Matching sensory images for equality provides no useful information because of this noise. Thus,
- Views should *maximize the prediction accuracy* of the causal $\langle V, A, V' \rangle$ relations. To navigate reliably, the robot must be able to predict with confidence the view it will see after executing an action at the current view. Of course, the highest confidence representation would be to cluster every sensory image into a single view; then all the $\langle V, A, V' \rangle$ relations would have 100% confidence. Such a degenerate causal layer, however, would be useless for navigation. Thus,
- Views should *maximize* the amount of *information about location* encoded in the causal representation. In order to build a topological map using the causal layer, the causal representation should distinguish between the views of different places in the environment as much as possible for the given levels of sensor noise and positional error.

Self-organizing Maps

We are investigating using a self-organizing map (SOM) (Kohonen, 1995) as a representation of the SSH view set. A standard SOM consists of a set of units or cells arranged in

[*] This work has taken place in the Intelligent Robotics Lab at the Artificial Intelligence Laboratory, The University of Texas at Austin. Research of the Intelligent Robotics lab is supported in part by NSF grants IRI-9504138 and CDA 9617327, and by funding from Tivoli Corporation.

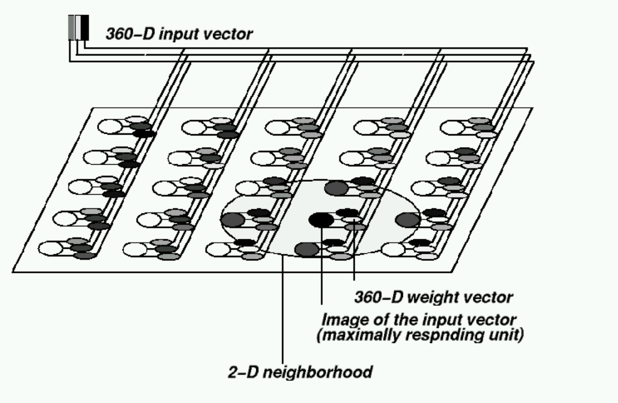


Figure 1: **5x5 Self-organizing Map.** In training, each sensor image is compared with the weight vector of each cell, and the weights are adapted so that over time each cell responds to a different portion of the input space.

a lattice.[1] The SOM takes a continuous-valued vector X as input and returns one of its units as the output. Each unit has a weight vector W_i of the same dimension as the input. On the presentation of an input, each weight vector is compared with the input using the Euclidean distance and a *winner* is selected as $\arg\min_i ||W_i - X||$.

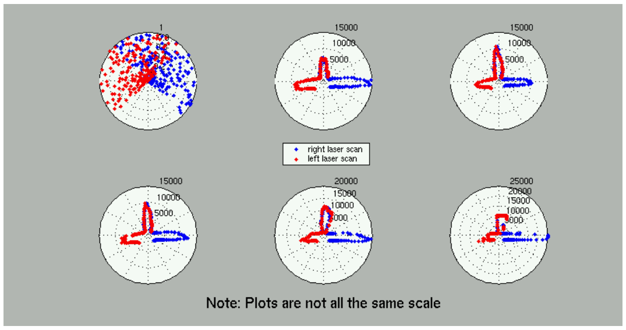
In training, the weight vectors in the SOM are initialized to random values. When an input vector is presented, the winning unit's weights are adjusted to move it closer to the input vector by some fraction of the distance between the input and the weights (the learning rate). In addition, the units in the neighborhood of the lattice near the winner are adjusted toward the input by a lesser amount.

Training begins with initially large values for both the learning rate and the neighborhood size. As training proceeds through numerous cycles, or *epochs*. In each epoch, the full set of training vectors is presented in a new random order, as the learning rate and neighborhood are gradually annealed to very small values. As a result, early training orients the map to cover the gross topology of the input space, and as the parameters are annealed, finer grained structure of the input space emerges.

In our implementation, the trained SOM is used as the view matcher: it takes a sensory image as input and the winning unit is taken to be the view at the current state.

Self-organizing maps have several properties which suggest that they will lend themselves well to representing views:

- *Clustering with topology preservation* – SOMs partition the input space into a set of clusters that preserves the topology of the original input space in reduced dimensions[2]. Thus similar views will be placed near one another in the SOM.
- *Data- and sensor- generality* – Because they operate on any input that can be expressed in the form of a vector, they are not specific to range-sensors or office environments as occupancy grids and segment/wall feature representations are.
- *Good intuitive fit to task* – As our results show below, a SOM generates views that look to the human eye like the sensor images they represent. Furthermore, the representation distinguishes between views of places that look different, while aggregating views from places that look similar.



Note: Plots are not all the same scale

Figure 2: **Development of a SOM unit.** Plot of six snapshots in the development of a single unit in the SOM over the course of training.

Related work

Many researchers have tried a number of unsupervised learning methods in robot localization, including SOMs (Nehmzow & Smithers, 1991), growing cell structures (Duckett & Nehmzow, 2000), nearest neighbor (Duckett & Nehmzow, 1998), and local occupancy grids (Yamauchi & Langley, 1997). Duckett & Nehmzow tested these methods in the same localization task, and compared localization accuracy against computational cost for all the algorithms.

The major differences between these efforts and ours is that they all use their respective methods to identify *locations*, while we are attempting only to distinguish and classify *views*. This frees us from the need to eliminate perceptual aliasing[3] entirely at the view-recognition level. The SSH model anticipates that some places may be indistinguishable at the view level, and mechanisms are built into the topological level to deal with this. (Kuipers & Byun, 1988, 1991)

Nevertheless, these works, particularly that of Duckett & Nehmzow, contain many good ideas which bear further investigation in the context of the SSH. For further discussion, see the future work section, below.

Experiment

We tested SOMs as a view-representation on a hand-selected set of 10 "distinctive states," chosen to represent possible

[1] The lattice is usually, but not necessarily, a 2D rectangular grid.

[2] In this context, topology refers to the topology of the space of sensor images, not the topology of places and paths in the SSH topological layer.

[3] The term for the situation when two or more distinct states in the environment are indistinguishable to the robot's perceptual apparatus.

termination points of control laws in the SSH. The states are shown in Figure 4. Each state is a pose in the environment defined by the robot's position and orientation (x, y, θ). To model positional uncertainty, several samples were taken from each state in which the canonical pose for that state was randomly perturbed in these ranges:

$$\text{-500mm} \leq \Delta x, \Delta y \leq \text{500mm}$$
$$-10^\circ \leq \Delta\theta \leq 10^\circ$$

All data were generated with the Flat robot simulator (Hewett *et al.*, 1999) designed to simulate Vulcan, one of the physical robots in our lab. The simulated robot is configured with two range-finders sampling 180 ranges at 1° intervals around a semi-circle. The range-finders are oriented at -45° and $+45^\circ$ from the robot's centerline, so that they cover a 270° arc around the robot, with 90° of overlap directly in front. The maximum range of the range-finders is 25m. The environment is a simulated layout of the fourth floor of Taylor Hall at the University of Texas at Austin, with walls, doorways, doors, etc. The noise model gives each point a 40% probability of a ±4cm error (20% +4, 20% -4)[4]. The resolution is sufficient to make out features as small as 100mm.[5]

We trained a 5x5 SOM for 5000 epochs on 10 images from each distinctive state. The sensory images were 360-dimensional vectors constructed by concatenating the output of the two range-finders.

Results and Discussion

Learned Views

We can see from the following figures that the trained map learned a view representation with a strong qualitative match to the sensor images. Figure 2 plots a single cell at six steps in the training.[6] The unit begins with random weights, and gradually grows to closely match the view of one of the states in the training data. Figure 5 plots the weight vector of each cell in the trained SOM as a range-finder image. Each cell's weight vector can be thought of as the sensor image to which that cell responds the most strongly. Many of the cells very closely resemble typical range scans that would be seen in the states in Figure 4, others resemble averages of many noisy scans from one state, or combinations of the scans of distinct but similar states.

Despite this visible similarity, the learned representation is not perfect. In particular, notice that many of the cells on the left side in Figure 5 are very similar to one another. These cells represent views learned from images of states 2, 3, 4, and 6. As you can see from Figure 4 these states are all in the long central corridor of the environment, facing walls or doorways with long open reaches of corridor to either side. One feature of the standard SOM learning algorithm is that it automatically assigns the cells to cover areas of the input space in proportion to their representation in the training data. Because scans 2, 3, 4, and 6 are close to one another in the input space, a large amount of the SOM representation is devoted to covering this space, posing problems for view representation which we address below.

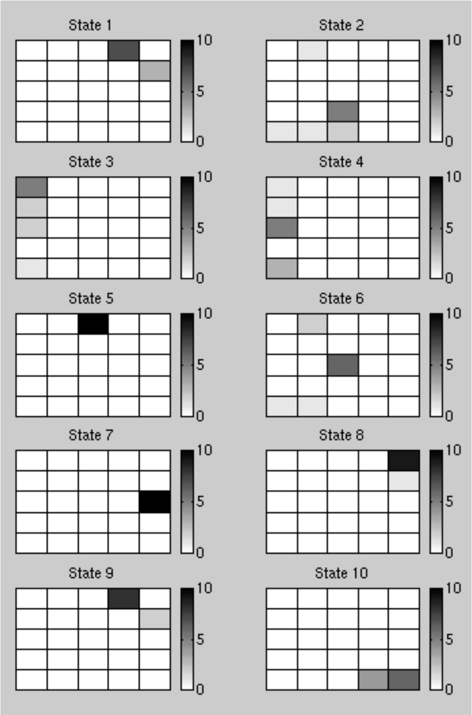


Figure 3: **Histograms of SOM output for each state.** Each grid is a histogram of the winning cell output of the 5x5 SOM for each sensor image from each of the 10 distinctive states. Darker shading means more images were mapped to that cell. States 1, 5, 7, 8, and 9 show strongly focused responses. While 2, 3, 4 and 6 show unfocused responses. The states correspond to the states in Figure 4. The weight vectors of the map cells are shown in Figure 5.

View Response to Distinctive States

To measure how well the trained SOM functions as a view representation, we presented the trained SOM with all the images from all 10 distinctive states, and created histograms of the winning cells for each state (Figure 3). The results are mixed. States 1, 5, 7, 8, 9, and, to a lesser degree, 10, show strongly focused responses, with all or nearly all images for a view activating the same cell, with the few "misses" activating nearby cells in the map. States 2, 3, 4, and 6, on the other hand, have broad unfocused responses. These are the states from the central corridor mentioned above. Because the images of these states are so much alike, the SOM has over-fit the data and created more views than are needed to represent the input data. This is a potential problem for use of the SOM as a view representation. We are investigating a number of possible solutions, which are discussed below.

Figures 6 and 7 illustrate the difference in SOM response

[4] This is a more realistic noise model for the SICK laser range scanners used on the physical robot than either Gaussian or uniformly distributed noise.

[5] The depth of the "dent" formed by a doorway with a closed door in our environment.

[6] In all the plots, the ranges from the two overlapping range finders, or the corresponding SOM weights, are plotted on a single polar graph with the centerline of the robot oriented vertically.

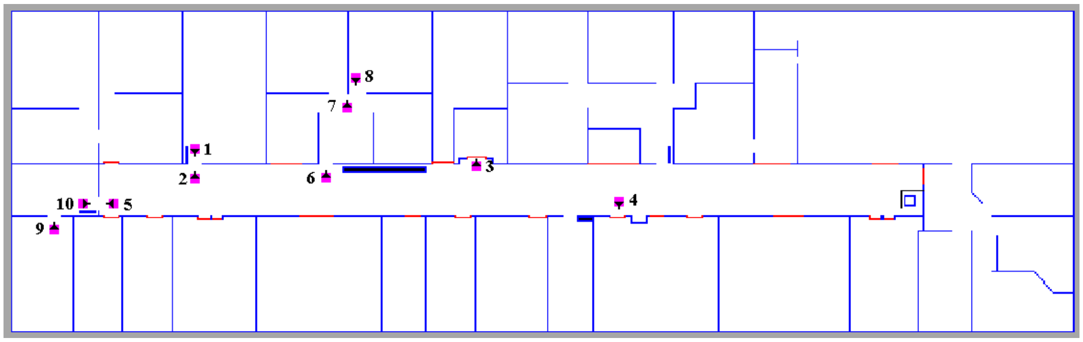


Figure 4: **Distinctive states in the environment.** The simulated robot environment used in our experiment, with the 10 distinctive states used for training marked.

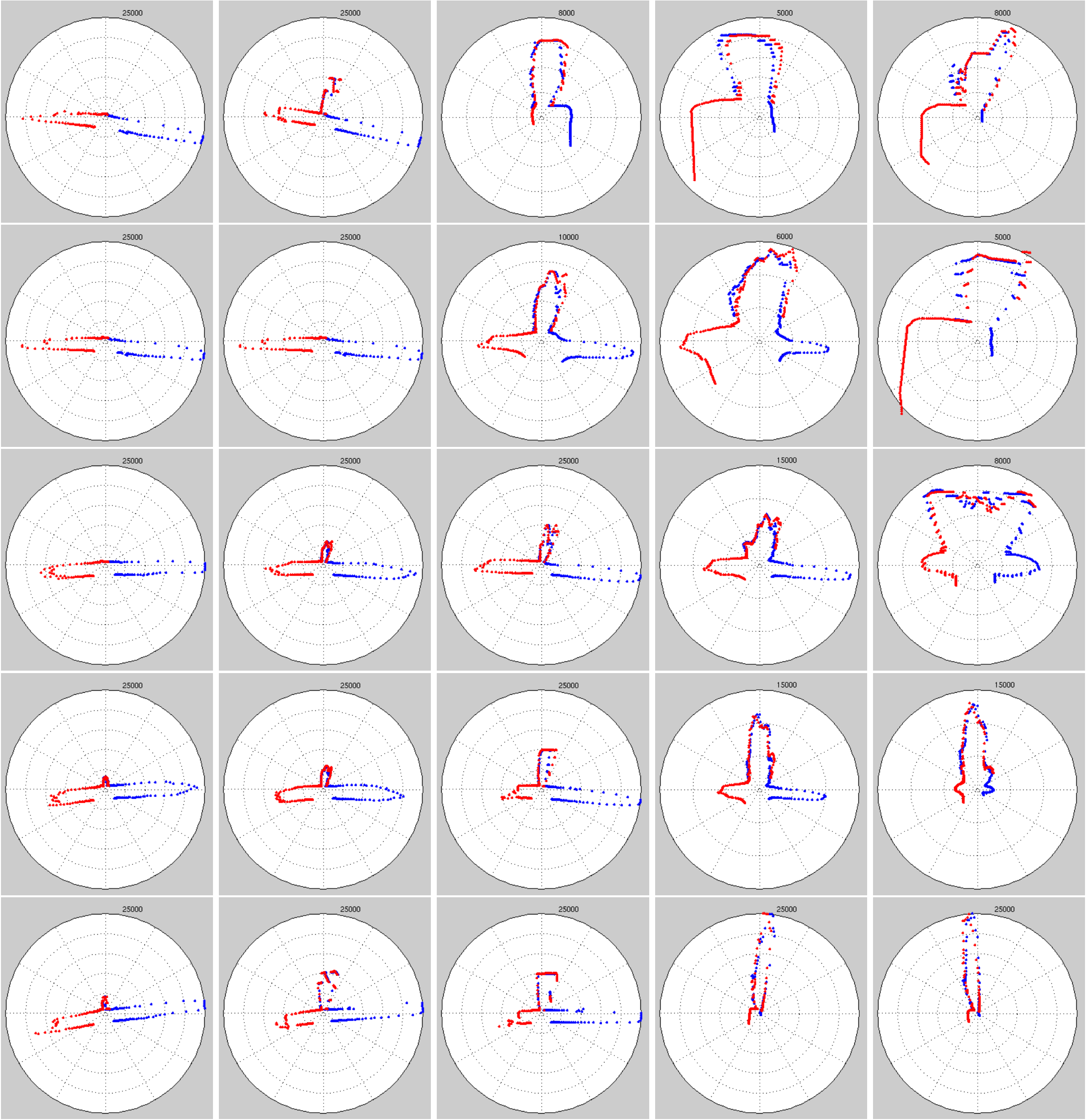


Figure 5: **SOM Weight Vectors as Views.** Here the weight vector of each cell in the 5x5 SOM is plotted as a range-finder image. Each cell's weight vector can be thought of as the sensor image, or view, to which that cell responds most strongly. These images correspond to the grid cells in Figures 1, 3, 6, and 7. Note the similar views represented along the left side of the map, contributing to the poor map responses in states 2, 3, 4, and 6 .

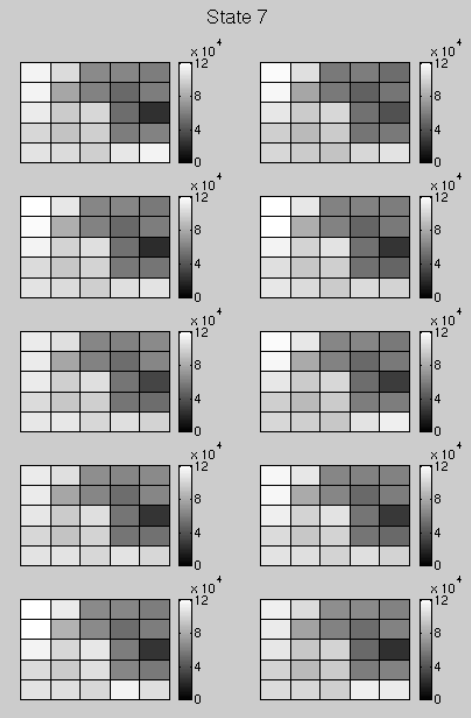


Figure 6: **SOM activation response for state 7.** The response activations of the SOM for all 10 sensory images of state 7, a state with a strongly focused histogram in Figure 3. Note that all the responses are nearly identical. (Darker shading means a closer match to the input.)

between the states that have been over-fit and those that have not. They plot the activation of the SOM to each individual view from states 7 and 2 respectively. State 7 had the most strongly focused response in the histograms, while state 2 had a very poorly focused response. Figure 6 clearly shows that the map response for state 7 is nearly identical for all sensor images. In Figure 7 the responses to the views of state 2 differ greatly, not only in the winning cell, but in the general pattern of activation. Matching the activated cells from the histograms of states 2 and 7 in Figure 3 with the view images in Figure 5, we see that the single view activated for state 7 is unlike any of the other views represented in the map. The five views activated for state 2, on the other hand, are all very similar to one another.

Perceptual Aliasing

The histograms of states 1 and 9 in Figure 3 display a classic case of perceptual aliasing. These two states have identical response histograms in the SOM. This is expected if the metrical differences between the images of the two rooms are within the bounds of the positional error and sensor noise in the data. Figure 8 shows two typical sensor images from these two states, along with the view of the most common matching cell for each. In these two, the robot was positioned inside a room facing the doorway, with similar geometry of walls and corners on all sides. These states are both represented by the same view in our SOM.

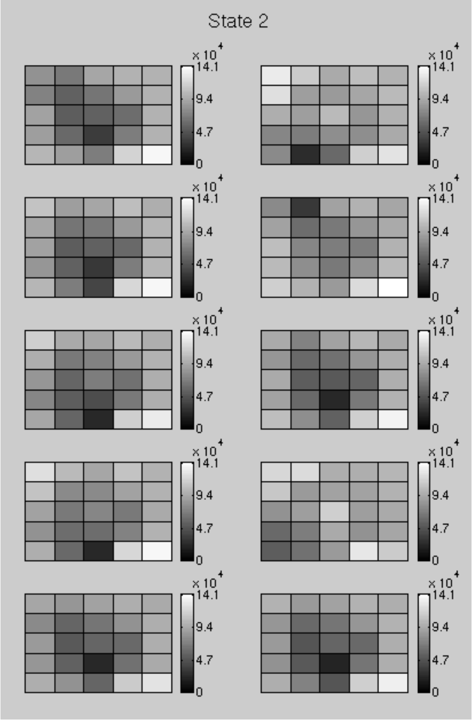


Figure 7: **SOM activation response for state 2.** The response activations of the SOM for all 10 sensory images of state 2, one of the least focused states in Figure 3. Note the difference in responses for different images. (Darker shading means a closer match to the input.)

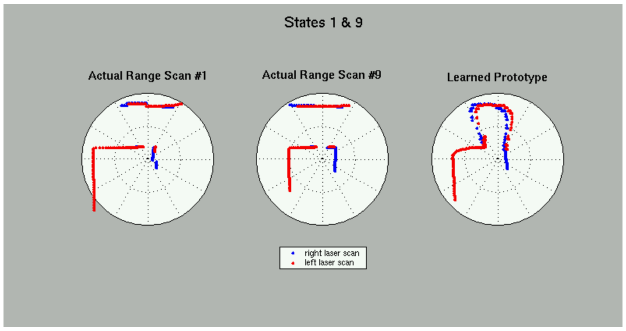


Figure 8: **Perceptual aliasing.** The the left and center images are scans from two different distinctive states in the environment (places 1 and 9). Both scans' best match is the view on the right.

Continuing and Future work

We are continuing to investigate using SOMs for view representation on several fronts:

Eliminating over-fitting

The most obvious area for continuing development is the elimination of the over-fitting problems mentioned above. We see several possible ways of doing this.

One simple means of eliminating over-fitting is to use a smaller SOM. Fewer total cells means fewer cells to allocate to any one area of the search space. The problem with this method is that it brings up the question of how to choose a good map size for an environment. Too small a map will cause unnecessary perceptual aliasing. Furthermore, we would like for our robots to be able to explore and map an environment with little or no prior knowledge of the number of distinct views that they will need.

Two methods, *active training* and *growing SOMs*, may lend themselves to this problem. In active learning, the learning rate of the SOM is dynamic, rather than on a fixed schedule. For example, if the learning rate is proportional to the distance of the training vector from the winning cell, then areas of the space that are already strongly represented by one cell will not receive a large allocation of cells from the map. Growing SOMs (Fritzke, 1996) are variants on ordinary SOMs that add cells to the network incrementally based on criteria such as the cumulative error in the winning nodes. This would allow the network to add cells only in areas of the input space where the coverage is poor. In addition, growing SOMs eliminate the need to guess in advance the size of the network needed to adequately represent the views, making growing SOMs more suitable for lifelong learning.

Quantitative evaluation

So far, our evaluation of the SOM as a view representation has been primarily qualitative. Some quantitative measure of the fitness of a particular view representation is needed to rigorously evaluate this and other representations. One possibility is to use the prediction accuracy of the $\langle V, A, V' \rangle$ as the metric. This will be possible when we have integrated the SOM view representation into an SSH-based exploration routine on a real or simulated robot.

Another possible metric is the Uncertainty Coefficient U, used by Duckett & Nehmzow (2000). It is an entropy-based metric for measuring the quality of landmark-based recognition systems, and is suitable for measuring systems learned through unsupervised methods such as SOMs or discrete clustering algorithms.

One exciting possibility, once a good metric has been discovered, is to implement on-line, adaptive learning by evaluating the quality of the view representation continually as the robot explores, and feed the results back to the active training, or growing SOM, to modify its learning as it explores.

Conclusion

Self-organizing maps show promise as a method for learning to recognize sensor views in the Spatial Semantic Hierarchy. On our test data, a SOM develops a strong, focused representation for views of several states in the environment. The remaining states' representations suffer because the SOM over-fit the data, encoding positional noise in the views. Modifying the SOM learning algorithm either through the use of active training or growing SOMs should allow learning a strong representation of the views of all states without over-fitting.

In our continuing work we hope to establish a quantitative measure of the goodness of a learned view representation, and integrate a learned causal layer into a working navigation system based on the SSH, with the ultimate goal of a system that learns all the levels of the SSH.

References

Duckett, T., and Nehmzow, U. 1998. Mobile robot self-localization and measurement of performance in middle scale environments. *Robotics and Autonomous Systems* 24(1-2).

Duckett, T., and Nehmzow, U. 2000. Performance comparison of landmark recognition systems for navigating mobile robots. In *Proc. 15th National Conf. on Artificial Intelligence (AAAI-2000)*, 826–831.

Fritzke, B. 1996. Growing self-organizing networks – why? In Verleysen, M., ed., *ESANN'96: European Symposium on Artificial Neural Networks*, 61–72. Brussels: D-Facto Publishers.

Hewett, M.; Remolina, E.; Browning, R.; Nguyen, H.; Lee, W.-Y.; and Kay, B. 1999. The flat 2-d robot simulator. http://www.cs.utexas.edu/users/qr/robotics/flat/.

Kohonen, T. 1995. *Self-Organizing Maps*. Springer.

Kuipers, B. J., and Byun, Y. T. 1988. A robust qualitative method for spatial learning in unknown environments. In *Proc. 7th National Conf. on Artificial Intelligence (AAAI-88)*, 774–779. Los Altos, CA: Morgan Kaufmann.

Kuipers, B. J., and Byun, Y.-T. 1991. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems* 8:47–63.

Kuipers, B. J. 2000. The spatial semantic hierarchy. *Artificial Intelligence* 119:191–233.

Nehmzow, U., and Smithers, T. 1991. Mapbuilding using self-organizing networks in really useful robots. In *Proc. SAB '91*.

Yamauchi, B., and Langley, P. 1997. Place recognition in dynamic environments. *Journal of Robotic Systems* 14(2).