in: *Proceedings ICANN '95 / Neuro–Nîmes '95*,
Vol. 2, pp. 381 – 386. Paris: EC2 & Cie, 1995

View–based cognitive map learning by an autonomous robot.

Hanspeter A. Mallot, Heinrich H. Bülthoff, Philipp Georg,
Bernhard Schölkopf and Ken Yasuhara

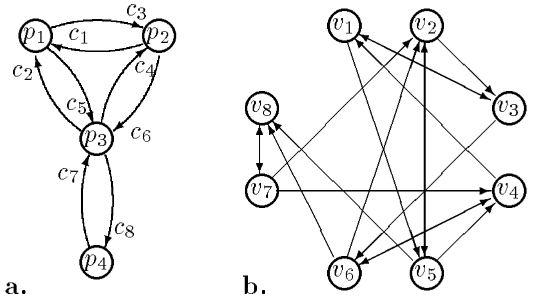
Max–Planck–Institut für biologische Kybernetik
Spemannstr. 38, 72076 Tübingen, Germany

Abstract

This paper presents a view–based approach to map learning and navigation in mazes. By means of *graph theory* we have shown that the view–graph is a sufficient representation for map behaviour such as path planning. A *neural network* for unsupervised learning of the view–graph from sequences of views is constructed. We use a modified Kohonen (1988) learning rule that transforms temporal sequence (rather than featural similarity) into connectedness. In the main part of the paper, we present a *robot implementation* of the scheme. The results show that the proposed network is able to support map behaviour in simple environments.

1 Introduction: The view–graph representation

A *cognitive map* is a neural mechanism supporting navigation and orientation tasks much as a real map of the environment. Schölkopf and Mallot (1995) presented a mechanism for the learning of a cognitive map of a maze from the sequence of local views encountered when exploring the maze. In this approach, the topological structure of a maze is represented as a graph where the *places* are the nodes and the (directional) *corridors* are the edges (see Fig. 1a). The explorative sequence of encountered views corresponds to the sequence of directed corridors passed along the way through the maze. Instead of reconstructing the place graph explicitly (e.g., Kuipers and Byun 1988), we consider an intermediate representation which is called the *view–graph*, i.e., a graph whose nodes represent the encountered views and whose (directed) links represent the temporal coherence of views (Fig. 1b). If one assumes that each corridor corresponds to exactly one view and all views are distinguishable, the view graph can be shown to contain all of the information required to reconstruct the place graph (Schölkopf and Mallot 1995). In the present paper,



c.

to view \ from view	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
v_1	–	–	g_b	g_l	–	–	–	–
v_2	–	–	–	–	g_b	g_r	g_l	–
v_3	g_b	g_r	–	–	–	–	–	–
v_4	–	–	–	–	g_l	g_b	g_r	–
v_5	g_l	g_b	–	–	–	–	–	–
v_6	–	–	g_r	g_b	–	–	–	–
v_7	–	–	–	–	–	–	–	g_b
v_8	–	–	–	–	g_r	g_l	g_b	–

Figure 1: **a.** Simple maze shown as a directed graph with places p_i and corridors c_j. **b.** Associated view–graph where each node v_i corresponds to one view, i.e. one directed connection in the place graph. Only the edges with go–labels are shown. In graph theory, b. is called the *interchange graph* of a. Simpler plots of b. are possible but not required for our argument. **c.** Adjacency matrix of the view–graph with labels indicating the egocentric movement leading from one view to another. g_l: go left, g_r: go right, g_b: go backward.

we show that the view graph can be used directly as a "cognitive map" for a mobile robot.

2 Competitive sequence learning

View graphs can be learnt from sequences of views by use of a competitive learning rule which translates temporal sequence (rather than featural similarity) into connectedness. The wiring diagram of the network is shown in Fig. 2, for details see Schölkopf and Mallot (1995). It takes two inputs, a feature vector representing the view information and a unique activity in one of a small number of movement units representing the most recent movement decision. The view vectors are mapped to view–cell activity by input weights ϱ_{nj} which specialize to one view during exploration. View–cells that become "winners" (i.e. recognize their corresponding views) in subsequent time steps are connected by an intrinsic (auto–associative) weight α_{ni}. This weight is modulated by input from movement–cell m_k, if the sequence of the two views was brought about by performing the corresponding movement.

Note that movement decisions are treated as *input* to the network. This makes it possible to use arbitrary exploration sequences such as random walks around the maze. In path–planning, possible motion decisions are presented to the network one after the other. Each decision is evaluated in terms of the time it takes for an activation of the cell representing the start view to reach the goal view cell. Path–planning is thus performed by "mental imagery" of the path to the goal before actually executing it.

In simulations with various mazes and exploration strategies, we found that the network is able to learn the view–graph from the sequence of views encountered during exploration of the maze. Convergence is rather fast (e.g. 60 learning steps for a four–place maze). It can be further improved by exploration strategies which can be derived from a theoretical analysis of view–graphs.

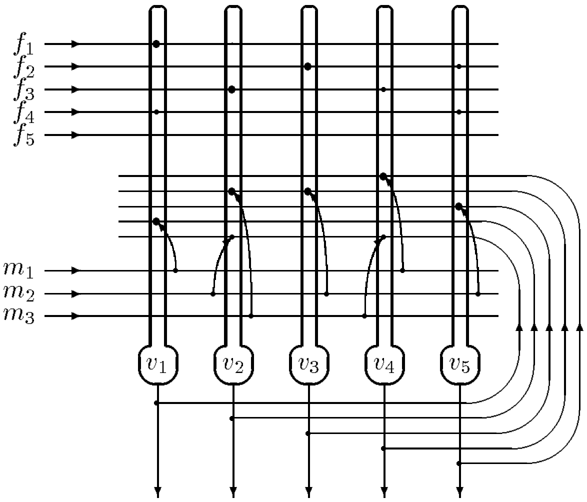


Figure 2: Wiring diagram of the neural network. $(f_1, ..., f_J)$: feature vector corresponding to current view. $m_1, ..., m_K$: motion input. $v_1, ..., v_N$: view cells. Input weights ϱ_{nj} : $f_j \to v_n$ subserve view recognition. Map layer weights α_{ni} : $v_i \to v_n$ represent connections between views. They can be modified by facilitating weights $\beta_{k,ni}$ indicating that view v_n can be reached from v_i by performing movement m_k.

3 The robot environment

3.1 The maze

In our autonomous robot experiments, we use a hexagonal maze with 12 junctions (places) and 24 directed connections between them (see Fig. 3). The views are replaced by black and white textures on the floor of the maze. These textures consist of black and white lines orthogonal to the corridor axis. The patterns which were actually used are shown in Fig. 3. Each texture is characteristic of one directed connection. Note that each corridor carries two texture patterns, one for each direction between the two places connected by the corridor.

3.2 The robot sensorium

The navigation experiments were performed with a Khepera® minature robot (Mondada et al. 1993). The robot uses eight infrared sensors, six of which are mounted evenly across the front, with the other two mounted at the rear. The two rear sensors were bent downwards in order to sense the floor texture. The robot thus uses two–pixel "image–sequences" generated by its own movement along the corridor.

When the robot enters a corridor, it uses its front sensors to center itself between the side walls. This is normally achieved by the time it reaches the midline between the two places (i.e., the narrowest point of the corridor). From then on the readings of the rear sensors are stored. The robot continues its way until it encounters an obstacle, i.e. the wall behind the place being approached. The obstacle is detected if the input from the front sensors on both sides exceeds a threshold. Thus, both obstacle avoidance and centering behavior are implemented as a "vehicle 3b" in the sense of Braitenberg (1984). The detection of the obstacle triggers the read–out of the stored sensor readings

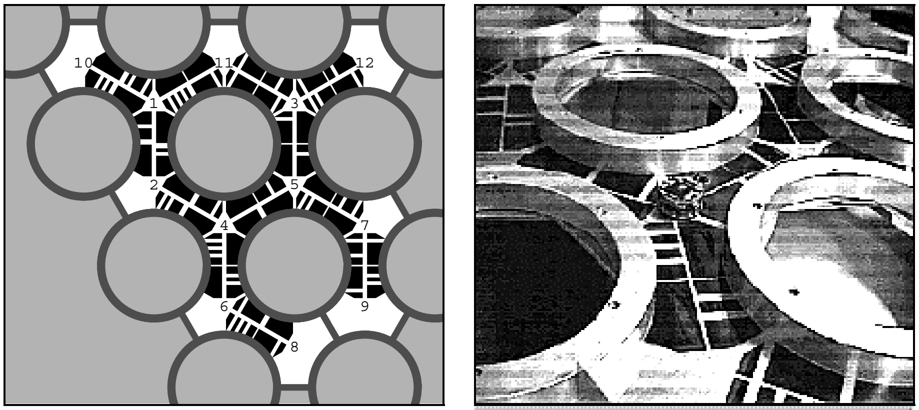


Figure 3: **Left:** Scheme of the hexagonal maze with 12 places and 24 "views". View–information is provided by the textured floor of the corridors. **Right:** Part of the real maze with place 6 in front and the robot exploring place 4.

representing the current view.

The interaction of robot and neural network is schematically summarized in Figure 4.

3.3 Preprocessing

The neural network sketched in Fig. 2 contains two parts, a hetero–associative memory for view recognition (upper weights in Fig. 2) and an auto–associative memory of the view–graph connections (middle and lower weights in Fig. 2). In a first series of experiments, we have by–passed the recognition part of the network by assuming ideal preprocessing which transforms the set of views into a set of canonical basis vectors. I.e., after this preprocessing stage, the input line f_i of the network will be 1 if view i was present and 0 otherwise. The results thus concern the learning and use of the view–graph connections. For more elaborated schemes of view recognition with associative networks, see Nelson (1991) and Crespi et al. (1993).

4 Experiments

The navigation system was tested in three behavioural modes (cf. Fig. 4):

1. In the **exploration mode**, the robot was driving through the maze using its obstacle avoidance and dead–end modules to get around. When facing an obstacle (i.e. a wall opposing a corridor), the next movement was selected at random.

   At each stop in front of an obstacle, the texture information collected prior to this stop was preprocessed as explained in Section 3.3. The resulting view vector was then presented to the network. This view information together with the movement decision was used by the neural network to learn the view–graph. In our experimental maze with 12 places and 24

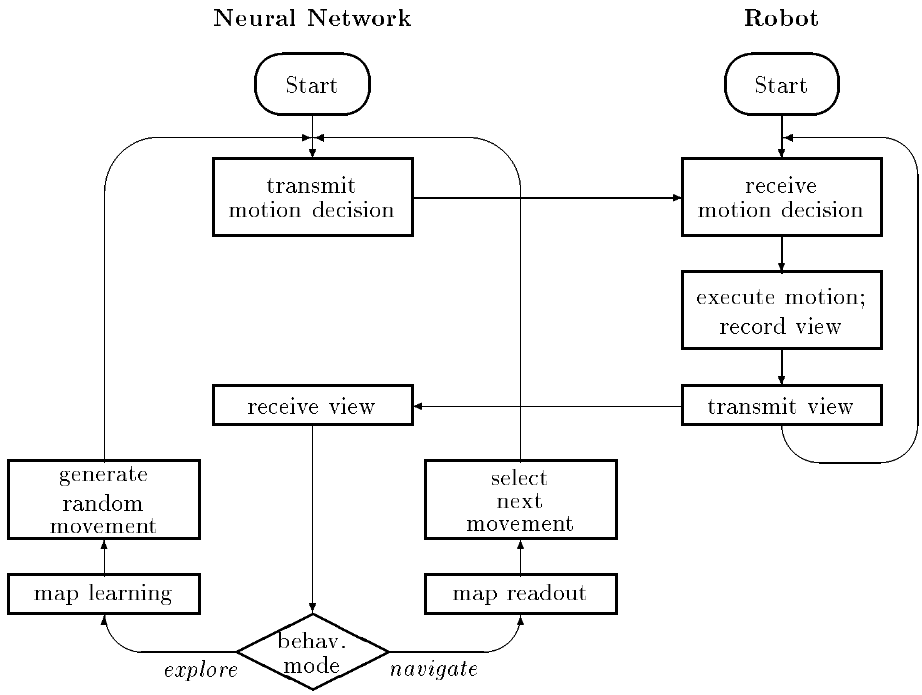


Figure 4: Control diagram for the interaction of robot and neural network.

views, learning was usually complete after about 200 learning steps. For the Khepera, this amounts to an exploration time of about an hour.

2. In the **navigation mode**, the robot used the information stored in the neural network to find a particular view (the "goal" view) from various starting positions. This goal view could be any view in the maze without additional learning required. Map readout in the neural network was performed as explained in Schölkopf and Mallot (1995). In our small experimental maze, the robot found the shortest way to the goal view in all trials.

3. In addition to the ordinary navigation task, we tested the system with a task called **navigation in the dark**. In this case, the robot had to approach teh goal view without any view information available on its way. In the neural network, the trajectory of the robot is still represented by the correct sequence of view–cell activations which are now propagated via the auto–associative weights α (see Fig. 2). In long trajecories, one would expect position errors to accumulate in this case. However, after sufficient training (200 movement decisions steps), the robot found its way in our experimental maze in more than 90 % of the trials.

5 Discussion

Our results show that the topological structure of a maze can be recovered from the sequence of views experienced when exploring the maze. The map information is suitably represented as a view–graph. Since the maze can be recovered from the view–graph using a simple algorithm, the view–graph is sufficient to support map behaviour. We conjecture that this is fact the minimal representation required to perform path planning.

Both the graph theory and the robot experiments reported here have been carried out for navigation in mazes rather than open environments. An extention to open environments can be made along the following lines. First note that we define a *view* as the input sequence encountered prior to the contact with an obstacle. Consider now an arena with a number of isolated objects and a robot moving around with the following rule: Fixate and approach an object until you (almost) touch it. Then select another visible object and proceed as before (see Sobey 1994). In this case, the arena with objects has an obvious place–graph representation, which differs from the place–graph of a maze only in that it need not be planar. The view–graph theory can be applied to this case in complete analogy to the case of maze navigation.

References

[1] V. Braitenberg. *Vehicles. Experiments in Synthetic Psychology.* The MIT Press, Cambridge, MA, 1984.

[2] B. Crespi, C. Furlanello, and L. Stringa. A memory–based approach to navigation. *Biological Cybernetics*, 69:385 – 393, 1993.

[3] T. Kohonen. *Self–Organization and Associative Memory.* Springer Verlag, Berlin, second edition, 1988.

[4] B. J. Kuipers and Y.-T. Byun. A robust, qualitative approach to a spatial learning mobile robot. In *SPIE Vol. 1003 Sensor Fusion: Spatial Reasoning and Scene Interpretation.* International Society for Optical Engineering (SPIE), 1988.

[5] F. Mondada, E. Franzi, and P. Ienne. Mobile robot miniaturisation: A tool for investigation in control algorithms. In *Proceedings of the Third International Symposium on Experimental Robotics*, Kyoto, Japan, 1993.

[6] R. C. Nelson. Visual homing using an associative memory. *Biological Cybernetics*, 65:281 – 291, 1991.

[7] B. Schölkopf and H. A. Mallot. View–based cognitive mapping and path planning. *Adaptive Behavior*, 3, in press (1995). (Available as /pub/mpi-memos/TR-007.ps.Z via anonymous ftp from ftp@mpik-tueb.mpg.de.).

[8] P. J. Sobey. Active navigation with a monocular robot. *Biological Cybernetics*, 71:433 – 440, 1994.