A Connectionist Approach to Spatial Memory and Planning*

Guido Bugmann
Centre for Neural and Adaptive Systems,
School of Computing, University of Plymouth,
Plymouth PL4 8AA, United Kingdom
http://www.tech.plym.ac.uk/soc/staff/guidbugm/bugmann.htm

18 April 1997

Contents

*To appear with a different layout as Chap. 5 in Landau L.J. and Taylor J.G. (eds) "Basic Concepts in Neural Networks: A survey", In the Series: Perspectives in Neural Networks, Springer

Abstract

This chapter describes the design and testing of a vision-based model of spatial memory (SM).

Firstly, three theories of biological SM are discussed: The Stimulus-Reaction theory (e.g. Hull (1932)), the cognitive map theory of Tolman (1948) and the Topological Network-map theory of Byrne (1979). The third one is the most promising theory of SM in humans and potentially allows for the most flexible planning at the lowest computational and memory cost in robot implementations.

Secondly, the Topological Network-map theory is translated into general principles: i) successive perceptual states are linked, ii) pairs of states are associated with the action responsible for the transition between them iii) links are exploited for building goal-directed sequences of states during planning and iv) backward planning is preferred to forward planning.

Two forms of connectionist implementation of these principles are discussed, using a sparse or a distributed representation of states. The sparse approach is the simplest and leads to a SM in the form of a view-graph (Schölkopf and Mallot, 1994). The sparse neural network model described here is characterized by: i) the use of a resistive-grid paradigm for backward planning, transforming the view-graph into a dynamic value-map; ii) the use of a separate network representing an inverse model of the robot—environment interaction; and iii) the use of state—object associations to drive planning by object-fetching tasks.

Thirdly, planning and map-learning experiments are performed with a robot. The perceptual states are defined as images from a narrow-angle on-board video camera. Objects are represented by letters drawn on cards placed in the environment. The instruction "fetch object A" is translated into "locate letter A".

These experiments reveal three problems with the implementation of the view-graph principle: i) local-minima due to interference between disjoint sequences of views built during exploration, ii) "representational dead-ends" in the form of views without a link to other views and iii) aliasing. The causes of these problems are discussed and solutions proposed. Solutions include a more complex vision system and a mechanism for internal integration of intermediate steps of plans.

1 Introduction

Spatial memory represents information on the location of objects in space in a form usable for spatial tasks. For instance, humans can be asked to indicate route (Ward et al., 1986), draw a map (Blades, 1990), estimate a distance (Sadalla and Magel, 1980, Vann Bugmann, 1995) or manipulate shapes mentally (Huttenlocher and Newcombe, 1984). Mobile robots need spatial memory and planning capabilities in several situations: Operation in environment without real- time remote control possibility (due to poor radio transmission of data or long light round trip delay in the case of planetary rovers) or no real time control possibility due to the inexperience of the user (household robotics). One of the most common use of spatial memory is the planning of routes between start and goal places. We would like to give this later ability to mobile robots to enhance their capability of executing high level commands such as "go to place X" or "bring me object Y".

Brain research is expected to provide clues on how to design an artificial spatial memory. There is a huge literature on psychological and neurophysiological aspects of spatial memory, the later notably centred on the hippocampal system (Burgess et al., 1995; Burgess and O'Keefe, 1996; Rolls, 1996) although recent imagery data indicate the involvement of wider system (Mellet et al., 1995, Maguire et al, 1997a, 1997b). So far, there is no generally accepted theory of how spatial memory works in animals. Three schools of thought have emerged over the years (see section 2): Those viewing spatial behaviour as a sequences of reactions to stimuli (S-R), those believing that spatial behaviour results from cognitive processes based on internal maps and those proposing that topological relations, not physical distances, are represented internally. In this paper we report on experiments with a model implementing topological principles which promise more flexibility. We have not attempted to incorporate anatomical details but there may be advantages in doing so, as discussed in appendix C.

An object fetching task requires vision for object recognition. We explore here the concept of also using vision for navigation. Potential long term advantages are: i) reduced number of sensors, ii) new knowledge from biology may more readily be transposed in the design, iii) users may give verbal route instructions to the robot using natural language, using visual landmarks in a way that communicates the "feeling" of taking the route (Ward et al., 1986).

Currently, there are four families of models of spatial navigation based on visual information[1], i) homing navigators, ii) grid based planners iii) view based navigator and iv) view-graph based planners.

Homing navigators use visual information to compute the direction of the goal or directly the motor command leading to the goal. Such systems are found in insects (Wehner et al., 1996) and in some models of navigation in rat (Burgess and O'Keefe, 1996) or for robots (Gaussier et al., 1996, Franz et al., 1996). These systems are essentially "attracted" by the goal and can get caught in local minima for some obstacles configurations. On the positive side, they also can flexibly accommodate new starting positions and take shortcuts.

[1] This is the most frequent biological case, and also the most interesting for autonomous robots applications. Other navigation models are based on ultrasound sensors (Recce and Harris, 1995, Yamaguchi and Langley, 1996, Thrun and Bücken, 1996) or optical range finders (Cox, 1991).

View based navigators are associating a particular action with a view. Thereby they can learn to follow a route based on views recognised along the route (McNaughton, 1989; Verschure et al, 1992, Buehlmeier et al., 1996; Röfer, 1997) or based on signs placed along the route (Joulain et al., 1996, Gaussier et al, 1996). These systems are "behaviour based" controllers (Brooks, 1986), which do not have the necessary flexibility for planning.

Grid based planners are based on the subdivision of the space into a fine grid, and vision is used for self localisation purpose. Each node in the grid is given a value, for instance the number of steps to the goal, that is then used for route planning. Various techniques can be used for computing the values (Barto et al., 1995). One of the fastest and most flexible is the resistive grid or Laplacian method (Conolly and Burns, 1993, Siemiatkowska, 1994, Bugmann et al. 1994, 1995, Althöfer, 1996, Glasius et al., 1996). With grid based planners, a viable route is always found if it exists, between any two points on the grid. In closed environments of limited size, such techniques have been used successfully by mobile robots (see e.g. Kortenkamp et al., 1993). For navigation over large distances or planning in high dimensional spaces, these techniques are not practical, using too much memory and computation time.

View-graph based planners use temporal adjacency between views[2] from given places to encode the topology of the space (Schölkopf and Mallot, 1994). By associating also the performed actions to each pair of successive views, goal- directed sequences of actions can be planned, if an uninterrupted chain of links from start-view to goal-view has been progressively assembled during past experiences. The model presented here is unique in i) its scheme for encoding actions, ii) its use of a fast generalised resistive grid planning method, iii) the possibility for planning towards objects found in views and iv) the use of narrow-angle images as views. This paper describes the first planning experiments with a view-graph in which views are images captured with a video camera on-board a real robot. These experiments reveal that view-graph-based planners can suffer from a local minima problem (section 4). Solutions are suggested in section 5.

The design investigated in this paper is a view-graph based planner evolved from an earlier grid-based planner described in (Bugmann et al., 1995). The proposed planning method is a generalisation of the resistive grid method aimed at extending the applicability of the resistive grid method into high dimensional problems, while preserving the good properties of grid based methods, namely the guaranty to find a solution, and reducing the computational load of the vision system. It should also avoid the self-localisation problem. In this generalised resistive grid method, nodes in the grid represent perceptual states, as defined by an image taken by a camera ("a view"), and do not require a Cartesian position to be known. The grid is constructed during exploration and has no predefined connectivity. Planning can be done by current spread along learned links, as in a standard resistive grid (Bugmann et al., 1995). Further, as the raw image defines a state, and implicitly the position, there is theoretically no need to extract features belonging to landmarks[3]. Obstacle detection is still needed, for avoidance purposes, but can be achieved by any low-level control mechanism, without the complexity of visual recognition and

[2] A view is, in its simplest form, a picture of the environment taken by the robot.

[3] Although this may save memory, as discussed in section 5.

localisation. The long-term storage of the positions of obstacles is not needed, as only states/views corresponding to the free space are encoded and used for planning.

In traditional grid based techniques, where nodes represent a point in the problem space, the higher the dimension of the problem, the larger the number of nodes. This is the main limitation of these methods. Even 6dimensional problems like a simple industrial robot arm require too many nodes for a precise control of the movement (Althöfer, 1996). Real life planning problems are of much higher dimension and are totally out of reach of grid based methods. However, with the approach proposed here, nodes can code for points in very highdimensional spaces but there are only as many nodes as new states discovered during the exploration process. Nevertheless, this number can become very large, as the experience of the robot increases. Some measures aimed at saving memory are discussed in section 5. These may involve off-line reorganisation of the spatial memory.

As for the organisation of the paper, in section 2 the current knowledge on biological spatial memory is summarised, with emphasis on humans. Some of the difficulties with Stimulus-Response and cognitive map models are described. The principle of a network-map (topological) model is selected for implementation.

In section 3, our implementation of a network-map model is described. First, the principles of a spatial memory in form of association between states and pairs of states and actions are formalised. In particular, recall procedures are defined for planning. Then, the potential problems associated with connectionist implementations using distributed or sparse representations of states are discussed. A connectionist implementation of a sparse representation is preferred for feasibility reasons. Finally the neural network is described in details along with the learning and planning procedures.

In section 4, the application of the model to the control of a real robot with a video camera is described. As noted by Touretzky et al, (1994) perception issues become important in real robot implementation. Accordingly, a simple visual set-up was used. Despite that, the visual image processing system, which is briefly described in the appendices A and B, is of greater complexity than the spatial memory network. The details of planning by the generalised resistive grid method are given. Finally, the learning and planning experiment are described. These reveal problems of local-minima due to interferences between disjoint sequences of views built during exploration, representational dead-ends in the form of views without a link back to other views and aliasing.

In section 5 possible solutions to the encountered problems are discussed. These include remote place recognition and internal multi-step planning and integration mechanism. The conclusion follows in section 6.

2 Biological spatial memory

How is spatial information stored in the brain, and how is it used for navigation and route planning ? Three approaches are briefly outlined below:

2.1 Stimulus-Response theory

The so called "behaviourist" school of thoughts saw spatial navigation as a sequence of response-movements caused by conjunctions of internal drives and external stimuli encountered along the path. Learning of the stimulus response pairs (S-R) is driven by reinforcements received during exploration or when reaching a goal (Spence, 1950). As the response R is produced by a conjunction of a stimulus S and a drive D, the approach should really be termed S,D-R, but the abbreviation S-R is commonly used. The S-R approach raised a number of questions. Some are due to a literal interpretation of the abbreviation "S-R": How can a same stimulus lead to different responses in case of different needs ? Other are justified. For instance: How can two stimuli with similar appearances (e.g. different intersections in a maze) be associated with different actions ? How does latent learning occur (learning of spatial relations without associating a need with each landmark) ? How can a basis of S,D-R triplets be used to respond to new problems ? Most of these questions were answered with complex theoretical constructs that were not general enough and difficult to confirm or infirm experimentally (see e.g. Hull, 1932, 1934, 1935).

O'Keefe and Nadel (1978) argue that a route based navigation system is fragile. Routes are taken to be a list of landmarks with instructions on how to go from one to the next. If an instruction is lost or if the appearance of a landmark changes, unrecoverable navigation errors can occur. O'Keefe and Nadel (1978) reminded that experienced migratory birds have a robust navigation system. Also, monkeys can head straight to a point that they have reached once using a route with numerous detours.

McNaughton (1989) and McNaughton and Nadel (1990) have suggested that the Hippocampus might be part of the route based navigation system, by associating scenes with actions. Evidences for the respective involvement of the Basal Ganglia and the hippocampus in the learning of S-R pairs are reviewed by Curran (1995).

In their present state of development, S-R concepts may account for stereotyped "habits" or "automatic" behaviours but they cannot explain the freedom of choice and creative problem solving that (sometimes) characterises human behaviour. Therefore, the S-R approach has not been accepted as a general model of biological spatial memory. However, it is still used to model insects (see e.g. Webb, 1995) and is investigated for the control of simple robots (see e.g. Brooks, 1986).

In recent S-R models for robot navigation, routes are stored in the form of sequences of S-R pairs such that the sensory input in one part of the route generates the action allowing to reach the next part of the path (Verschure et al., 1995; Buehlmeier et al., 1996; Rao and Fuentes, 1996). With a repertoire of such routes, planning can theoretically be performed (Hull, 1935). In models by Arbib and Lieblich (1977) and Schmajuk et al. (1993), drive-reduction values are associated with intermediate Stimuli in a route (as in Hull (1932)) so that routes can be selected on the basis of the drive of the moment. Such systems do not have the flexibility to execute instructions such as those mentioned in section 1, because a Stimuli would need to to have a value attached for each possible goal.

2.2 Cognitive maps

In reaction to the behaviourist thinking, Tolman (1948) introduced the concept of cognitive map where an action is the result of a "cognitive" process performed on a corpus of internal knowledge ("cognitive field"). Perception is seen as an information gathering process which modifies the cognitive field rather than driving the behaviour directly. Using this internal knowledge was seen as analogue to reading a geographic map to plan a route. Tolman described cognitive maps as "relatively narrow and strip-like or relatively broad and comprehensive":

> "In a strip-map the given position of the animal is connected by only a relatively simple and single path to the position of the goal. In a comprehensive-map a wider arc of the environment is represented, so that, if the starting position of the animal be changed, or if variations in the specific routes be introduced, this wider map will allow the animal still to behave relatively correctly and choose the appropriate route".

It has recently been noted the rat experiments cited by Tolman (1948) in support of his concept did not require the use of spatial maps. The tasks could also been realised using "cue guidance" strategies (O'Keefe, 1991) or "associative" principles (Wishaw, 1991). However, other experiments with monkey support the existence of cognitive maps. In these, locations of food caches are shown in a random order. The monkey usually retrieve the food in some economically ordered path (Menzel, 1973), which is consistent with the idea that an internal map is being used. Thomson (1980) reports that blindfolded subjects can avoid previously seen obstacles, by updating their position on an internal map of the environment. This map seems to "fade away" after 8 seconds.

Support for a geographic-map-like spatial memory can also be sought in experiments showing that subjects can draw maps "as seen by air". However, this seems easier based on a verbal description of a scene (Ferguson and Hegarty, 1994) than based on personal experience navigating in an environment (Blades, 1990).

Other finding are not accountable by a map model, such as the curious fact that the time taken to estimate the distance between two towns in different states is smaller than for two towns in the same state. This was taken as evidence that spatial memory is organised hierarchically (see a review by Tversky, 1992). Further, the number of intersections of the route linking two places (Sadalla and Magel, 1980), or the subjective "importance" of the places (Vann Bugmann, 1995), influences the estimated travel distance between these places.

On one hand these observations supports the notion that large-scale spatial memory is not represented in a strictly "geographic map-like" way[4]. On the

[4]Formally, this conclusion is not very strong, because the mechanism for retrieving the information may also be affected by these factors.

other hand, the data reported by Thomson (1980) suggest that, at least for the immediate environment and temporarily, a map preserving the exact geometry may be used.

Presson et al, (1989) distinguish two forms of cognitive maps. The first, an episodic memory, contains information in a picture-like format and with a specific orientation determined at the time of perception. The second is a stable world model in which the subject can move and observe. The orientation of the model is in register with the subjects physical orientation, but can be rotated mentally in an effortful process (Rieser, 1989).

A world model is more close to the nature of our sensation when living in the 3-D world, representing the point of view of an actor immersed in the environment (Presson et al., 1989), while a cognitive map in its literal interpretation is a picture of a map (Levine et al.,1982). Acting in the real world may recruit brain mechanisms similar to those used for mental operations in the world model (Rieser, 1989; Maguire et al., 1997b, Mellet et al., 1995, 1996).

In the hippocampus of the rat, cells are found which fire when the rat is in a given place (egocentric place representation) (O'Keefe and Speakman, 1987, Speakman, 1987). In the postsubbiculum, near to the Hippocampus, cells are found which indicate the direction of the head relative to the cues in the environment (Burgess et al., 1995). Both place cells and head direction cells give the kind of information needed to align an internal world model and visual sensory perception (see e.g. Recce and Harris, 1996).

Recently, Rolls (1996) found that cells in the Hippocampus of the monkey respond to views of specific places, providing an allocentric space representation. These cells may be used to encode the position of objects.

The map concept is attractive for theoretical reasons: A map has the property to be observer independent, can be rotated and read from various positions and can be used to solve route planning problems. In a map, relations between objects are encoded, not relations between observer and objects. O'Keefe and Nadel (1978) proposed that a cognitive map is located in the Hippocampus.

In robot applications, a strict transposition of the map concept leads to encode the positions of objects and the robot using Cartesian coordinates. For planning, the obstacle-free space must be calculated on a detailed map, this is then divided into cells, then a standard gridbased planning technique can be used (Latombe, 1991). This approach is associated with the same self-localisation problem[5] as for grid based planners (section 1). Planning over long distances, e.g. between two parts of a town, requires a very detailed map and huge computation time. Most autonomous mobile robots of today use Cartesian planners.

[5]The availability of the Global Positioning Satelite System (GPS) may solve this problem in some applications.

2.3 Topological network-maps

The network-map theory of spatial memory proposed by Byrne (1979) states that the mental representation only represents topological connectedness, but not two-dimensional distance information.

Moar and Carleton (1982) studied the acquisition of spatial knowledge as inferred from a sequence of pictures representing views along two routes with a common segment[6]. They found that after only one presentation of the two sequences, judgements of distances or directions were as accurate between two points belonging to the same route as those belonging to different routes. This was taken as evidence that the knowledge about the two routes was combined from the start in a single network of routes, in support of the network-map representation proposed by Byrne (1979).

Such a representation can be used to plan routes (although only along known segments linking places) and may be compatible with the distortions of the perceived distances noted above (although a detailed explanation remains to be produced).

However, Moar and Carleton (1982) also found that the direction judgements[7] of the subjects improved greatly with experience. This was taken as evidence that although a network-map model may account for the initial state of knowledge during the learning phase, a more vector-like representation may emerge later, in which straight-line distance and relative directions of places are represented.

Evans et al. (1981) found evidence that landmarks are used as initial anchor points and that only their relative positions[8] were stored. With increasing experience, an increasingly dense network of routes linked these landmarks[9] and provided more positional constraints, enable subjects to become more precise in estimation directions and Euclidean (straight-line) distances. Thorndyke and Hayes-Roth (1982) found a similar increase in accuracy as a consequence of repeated navigation through the environment

In contrast, the accuracy of straight-line distance estimations by taxi drivers in Paris is similar to that of office workers having lived for the same time in Paris (Peruch et al., 1989). This suggests that experience is not the factor that develops a (better) survey knowledge.

How is the length of a route encoded ? Measurements show that subjects can estimates distances relatively precisely. When asked to estimate the length of a route, they sometimes give spontaneously the travelling time first (Vann Bugmann, 1995), as if time is more readily retrieved than distance. There is a linear relation between estimated time and distance, suggesting that subjects may infer the distance from the travel time. It is not known where and how

[6]The routes formed a H and their common segment formed the horizontal bar of the H.

[7]"Assume that you are in the same position as the photographer who took this picture. Point in the direction of landmark x"

[8]Relative position refers to "A is closer than B" or "A is to the East of B", etc

[9]It commonly assumed that there is something distinctive about certain elements of a landscape which makes them good candidates to become landmarks. This aspect of the theory is rather vague. Formally there is a difference between a place and a landmark, a place being best defined by the configuration of objects or landmarks seen from that place. One can stay at a place but see a landmark. Thus route are formally links between places rather than landmarks.

time is stored in the brain[10] nor how it is retrieved by subjects to produce their verbal responses.

Muller et al. (1996) have proposed that the Hippocampus might implement a "cognitive graph" in which the links between places-cells corresponding to neighbouring positions are strengthened.

In robotics, the use of topological network-maps has only recently started to be explored (Schölkopf and Mallot, 1994; Bachelder and Waxmann, 1995). It may be noted that, although the links between landmarks in network-maps may be implemented in the form of S-R-S chains, they represent task-independent spatial relations. This is to be contrasted with S-R systems in which some "goal-specific utility" is encoded in the links or nodes[11]. Thus S-R systems can only be used for reaching predefined, built-in goals.

2.4 Planning

How do humans plan their movements? Human route planning is a multi-criteria optimisation process, involving considerations such as safety, scenery, speed, cost, etc. It is not just a problem of linking source position to goal position. Planning social behaviour is even more complex, due to uncertainty regarding current situation and also the outcome of actions.

In the early days of AI, methods were designed to mimic the way humans solve problems like mathematical puzzles, demonstrate equations or play chess (Newell and Simon, 1972). In all these activities, the main problem was to discover a set of operations and intermediate states allowing to link an initial state to a goal state. In the spatial domain, this is equivalent to exploration of an environment with unknown connectedness. In contrast, planning consists of finding a path in a situation where links and intermediate states are already established. There are no indications on how this process is performed by humans. For instance is forward or backward planning used ? (see section 3). Gray (1995) argues that planning is a subconscious process where only the result is available to consciousness.

Not much is known on the neural mechanisms of planning (Altman, 1995). Jeannerod (1994) proposes that planning is the assembling of sequences of movement schema stored in posterior parietal cortex. Shallice (1988) gives evidence that planning is crucially dependent on the prefrontal cortex.

The notion of planning by spread of activation has been mentioned several times in relation to network-map models (Byrne, 1979; McNamara et al., 1989) or in S-R models (Mataric, 1990; Maes, 1989).

2.5 Summary

In summary, there are more questions than answers regarding biological spatial memory. Topological maps seem to stand the best chances to form the basis of a plausible model biological spatial memory. However, they need to be

[10] Animals can learn time intervals and conditioning experiments have been performed with durations of over 15 minutes (Lejeune and Wearden, 1991). A model of a neural mechanism for timing is found in (Bugmann, 1997)

[11] With resistive grid planning method described in section 3.5, the value of a Stimulus is assigned dynamically, in a task dependent way.

made consistent with the availability of map-like (vector-like) knowledge to experienced subjects and the memory experiments cited by Thomson (1980).

One approach is to assume that the relatively simple routes used in the experiments of Moar and Carleton (1982) may have enabled subjects to integrate the paths mentally and infer the bird's eye view of the local topography. Thus its is conceivable that a picture-like map may be constructed using mental inference, similarly to a map built from verbal descriptions (Ferguson and Hegarty, 1994), possibly on a temporary basis (Presson et al, 1989). Another possibility, which is touched on in the discussion (section 5), is that the inference of straight-line distances may lead to the incorporation of straight-line links into a topological spatial memory which thereby may gain vector-like characteristics.

Speculatively, if the network of links then becomes dense enough, it may be possible that a topological map becomes functionally similar to a world model. This is in the sense that, to whatever object the minds-eye devotes attention, the map will produce an associated view and indication of direction. Conversely for attention devoted to a direction. It is unclear if the memory experiments cited by Thomson (1980) can be explained along similar lines.

Topological network-maps could perhaps also be made consistent with the existence of habits. These are mainly revealed when something in the environment has been changed so that an usual route becomes inadequate. Taking the usual route in such a situation may be due to planning being done with obsolete information or, in other cases, with information provided by the memory rather than by visual perception. Thus the expression of habits may be due to planning with a "lazy" information management system rather than the use of a separate S-R system.

Regarding robot implementation, cognitive maps and S-R models have a number of known limitations. Thus we will explore further the concept of topological network-map in the remaining of the paper.

3 Connectionist implementation

3.1 Principles of a Network-Map based Artificial Spatial memory

The following general principles are proposed to describe a network-map model:

i) The sensory experience of a robot are called states (or "scenes" or "view"[12] when that experience is predominantly visual): S_{t1}, S_{t2}, ... The indexes t1, t2 indicate different instants, separated by any time interval.

ii) The states are represented in a linked way, such that if S_{t2} is the next state experienced after S_{t1}, the fact that S_{t2} can be reached from St1 is encoded and exploitable for planning.

iii) The action $a(S_{t1}, S_{t2})$ performed to reach S_{t2} from S_{t1} is also memorised, and encoded in such a way that it can be recalled and re-executed if S_{t2} has to be reached again when the system is in state S_{t1}. An action can be a "ongoing task" rather than an action with a predefined duration. For instance, the statement of a goal S_{t2} and the instruction "move forward", can be taken to mean "move until S_{t2} is encountered".

iv) Backward planning, to reach a state S_{tn} is performed by recalling all states S_{tm} which have been immediate precursors to S_{tn}, then by recalling all states precursors to the states S_{tm}. The process is repeated iteratively until one the recalled states corresponds to the current state. The first action to be done in the plan is the one that is memorised as having caused the transition from the current state to the one following it in the recalled sequence.

It should be noted that backward planning differs from forward planning which operates by recalling all know successors to the current state, then the successors of these successors, and so on until the target state is found (Schölkopf and Mallot, 1994). The advantage of backward planning over forward planning is that, at the end of the search, the action to be done is immediately known because it is inferred from the last updated states. Forward planning only tells that a path (sequence of actions) exists.

An "ongoing task" could be implemented by a "Behaviour" in the sense of Brooks (1986)[13]. An example is to "walk on the pavement in this direction until...". Using "ongoing tasks" allows to save memory, as illustrated by the problem of planning a route that may include a train trip. In that case, there is no point to create a memory of all places along the route or to use this information for planning. By using an ongoing task like "staying in the train", a single link suffice between the starting station and the arrival station. Indeed, a control system is needed to determine when to encode or not to encode a scene. Alternatively, sequences of stored action / perception could be "folded" at a later time to reorganise the memory and clear useless sub- sequences. Therefore, the principle v) is proposed.

v) The content of the spatial memory can be examined after a series of experiences in such a way that a sequence of states $\{S_{ti}, S_{tj}, ..., S_{tk}, S_{tl}\}$ can be "folded" and a direct link between S_{ti} and S_{tl} can be created. Concurrently, a "procedure" p(S_{ti}, S_{tl}) is created which replaces the sequence of actions

[12] The terms "view", "state" and "scene" will be used interchangeably.

[13] In the sense of Brooks (1986), a "Behaviour" is a low level S-R program dealing with tasks like "keeping balance", "avoiding obstacles", "follow a wall", etc. The combination of several "Behaviours" is supposed to result in the observable behaviour in the usual sense.

$\{a(S_{ti}, S_{tj}), ...,a(S_{tk}, S_{tl})\}$. The states $S_{tj},..,S_{tk}$ and the associated actions can then be removed from memory.

In our model, only the principle i) to iv) will be implemented. The need for principle v) and constraints on its implementation are discussed further in section 5.

3.2 Sparse versus distributed representation of views

A view is a set a of measurements made on the environment which are characteristic for a given position and orientation of the robot. The view can be a set of marks on the floor (Mallot et al., 1995), a 360 degree image provided by a conic mirror (Franz et al., 1996) or a rotating photo sensor (Röfer, 1997), a 360 degree panoramic view log-transformed for maximum detail in front (Bachelder and Waxman, 1995) a set of objects associated with the viewing angle (Gaussier and Zrehen, 1995), a recognition grid based on ultrasound reflectometry (Yamauchi and Langley, 1996) or a set of ultra- sound distance readings covering 360 degree (Recce and Harris, 1996). In the system used here, the view is a 30 degree narrow angle image produced by a video camera.

The spatial memory system to be designed is a memory of linked states (views), associated with action information. One may see it as a memory of sequences of $state_i \rightarrow action_{ij} \rightarrow state_j$ but there is more than sequence storage in this problem. The states and actions must be represented in such a way that the system allows planning and extraction of action information. Two approaches to that, distinguished by a distributed or a sparse representation of states, are now discussed briefly.

Distributed representation:

In a distributed representation, a state is represented by the activity of a large number of nodes. The most distributed representation is the raw image, as all pixels contribute to the definition of the scene.

Figure 1a illustrates a distributed approach. A scenes is encoded by the activity of a set of nodes N_1, representing features extracted from an image or being a copy of the image itself. A neural network "Linking" learns to reproduce the past scene S_{t1} from the current scene S_{t2} and projects it to N_2. Another neural network "Transition" learns that action $a(S_{t1}, S_{t2})$ has caused the transition between the two scenes. Replaying a sequence of scenes proceeds by setting the activity in N_1 to S_{t2}, and choosing an action a in the Transition network. The combination of S_{t2} and the action a allows then the "Linking" network to reconstruct in layer N_2 the scene S_{t1} having preceded S_{t2}. Although N_1 and N_2 are represented here as two separate layers, they could also be merged into a recurrent architecture of the type seen in (Schmajuk et al., 1993) and (Bachelder and Waxman, 1996), although these authors do not use distributed representations.

Backward planning is done by setting the activity in N_1 equal to that of the goal scene. Then all possible predecessors are replayed until the current state is found. As the actions distinguish the predecessors, the linking must be action-dependent. For forward planning, the system should be modified to reproduce S_{t2} from S_{t1}. Planning is done by replaying all scenes that are possible successors of the current one until the goal scene is found. Planning by sequence replay is a sequential process taking a time proportional to the depth and the number of branches of the tree to explore. There is also a need for some form of memory of scenes already replayed. So far, there is no working example of a planner based on a distributed representation.

Sparse representation:

In a sparse representation, the activities of a small number of nodes represent the scene. The most sparse representation is the one where a single node represents a scene.

Figure 1b illustrates a sparse approach. A scenes is represented by one node at the output of a "decoder" network. Directional lateral connections encode the experienced transitions between scenes. A "transition" network encodes the action that was done to cause the transition.

For backward planning, these links allow to activate S_{t1} from S_{t2}. Planning is done by activating the goal state and letting the activation propagate until the current state is activated. For forward planning, the links must activate S_{t2} from S_{t1} and planing is one by activation propagation from the current state to all the states until the goal state is activated. If the view graph is implemented as a resistive grid, current gradients or potential gradients can be used to determine the neighbour nearest to the goal. In that case, backward and forward planning are parallel search processes exploring all branches at the same time and taking a time proportional to the depth (length of sequence to find). The use of spiking neurons for planning is discussed in appendix C.

A sparse approach was used in (Schölkopf and Mallot, 1994) for encoding the links. However, backward planning was used. For that purpose, the action nodes were connected to the relevant links, in order to modulate their strength. During planning, each action node was activated in turn to determine which neighbour of the current state node the action allowed to reach. Activity was then spread forward from this node to all connected nodes until the goal was reached. The neighbour that was nearest to the goal, in terms of spreading time, was then chosen as the next to be reached. This search process had to repeated for each intermediate state and each possible action. With backward planning using a resistive grid method, a single current spreading process can suffice to set-up a potential landscape that is usable in each state and does not need to be recomputed until the goal is reached (Bugmann et al., 1995).

In (Bachelder and Waxman, 1996), conjunctions between the current state and the current action are used as input to a network that is trained to predict the next state. It is a representation of spatial knowledge in the form of S,R-S triplets. This form of spatial memory could support forward planning with the some of the complications described above for distributed representations.

A variant of sparse representations are topological maps[14] (see for instance Gaussier and Zrehen, 1994, Buehlmeier et al., 1996; Zimmer, 1996). These maps area usually two or three-dimensional grids of nodes. States with similar sensory signatures are assigned to nodes with positions close to each other in terms of physical distance in the grid. This property may be used to generalise the properties discovered in one place to another place (Gaussier and Zrehen, 1994). For planning however, this generalisation may cause difficulties: Similar sensory patterns do not necessarily correspond to similar positions in the world (Bachelder and Waxman, 1995; Rao and Fuentes, 1996). This is

[14] These "topological maps" inspired by Kohonen (1988) are not to be confused with topological network-maps or viewgraphs.

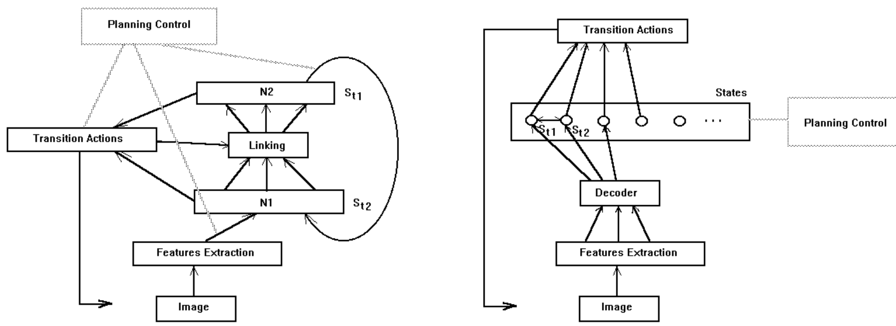

Figure 1: **a) Left:** Distributed spatial memory. **b) Right:** Sparse spatial memory. The operation principles of these models are described in the text.
i) In both cases the raw image is first pre processed to highlight relevant features, e.g. regions of high contrast. In the distributed case, these features are copied into the layer N_1 and represent, as a whole, a state. In the sparse case, the features are classified and the state is represented by one active node at the output of the decoder which activates its corresponding node in the "State" layer.
ii) During exploration, in the distributed case, the "Linking" module learns to reproduce the past state from the current state and from the action just done. In the sparse case, lateral connections are built in the state layer to encode transitions and the action having caused the transition is encoded in the "Transition Actions" layer.
iii) During planning, in the distributed case, the planning control module must manage the sequential replay of predecessors of a state S_{tn} (goal state). In the sparse case, the controller manages the spread of activation from the goal state nodes in the grid to all other connected nodes in the grid. In both cases, planning ends when the current state is reached (this is an example of backward planning).
iv) The state which was activated just before the current state become the next subgoal of the plan. The pair of states (current state; state which was activated just before) defines the action to be executed.

partly avoided in (Zimmer, 1996) where the estimated position is part of the "sensory" pattern[15]. Overall, the approach may deserve more investigations. One reason is that it makes sense to reduce the high dimensional feature space into the low dimensional problem space (position and orientation).

We may note that all existing models use a form of sparse representation. For practical implementation and testing with a robot, there are a number of reasons to prefer a sparse representation over a distributed one:

i) Planning time is shorter in the sparse model than in the distributed case because only single nodes need to be activated, taking at most on time step per pair of scenes. In the distributed case, the reproduction of scenes involves at least one extra-layer of nodes in the network "linking". Thus a pair of scenes takes at least 3 time steps, including the time to copy N_2 into N_1. In simulation of the distributed case on a serial computer, extra time is used to loop through all the neurons representing a scene.

ii) Planning in the distributed case requires a more complex control circuitry that in the sparse case, to manage the sequential search process. In the sparse case, a simple resistive grid algorithm can perform the search (Bugmann et al., 1995).

iii) The distributed case requires a powerful "Linking" module that can generate one image from another. The only architecture that can learn fast without catastrophic interferences is a layer of RBF nodes where one node is recruited per association to learn. This uses potentially more nodes than the sparse representation where there is one node per state (image) and each association is represented by a link[16].

iv) Every aspect of a sparse implementation is realisable with existing neural network tools and there seems to be no functional advantages to justify the design of the more complex distributed system. Details of the implementation of a sparse model are given in the next section.

3.3 Implementation of a sparse model.

Figure 2 summarises the model investigated here. The view nodes represent a low resolution version of the image of the environment seen by a video camera (see Appendix B). Each state node is trained to respond to only one view, as described below. The focus nodes represents the part of the image in which an object is found. In our tests, objects are characters. These are surrounded by a frame, which simplifies the focusing process, as described in Appendix A. Each "character" node learns to responds to a character when it is first met during exploration (see appendix A). State nodes and Characters nodes use Gaussian Radial Basis Functions (RBF) (see equation 1) which can be trained from a single presentation.

The identities of the characters found in the scene are not explicitly used to define a state. Characters are already part of the visual features defining the scene and there is no need for additional high-level "semantic" concepts, as represented by the outputs of "character", to be used to define states. However, the high-level representation of letters needs to be associated with the states

[15]This in turn requires to estimate ones position, one of the problems that were at the origin of the view-graph based approach...

[16]The sparse model ends up requiring one node per transtion too, in the action-encoding module (see section 3.4).

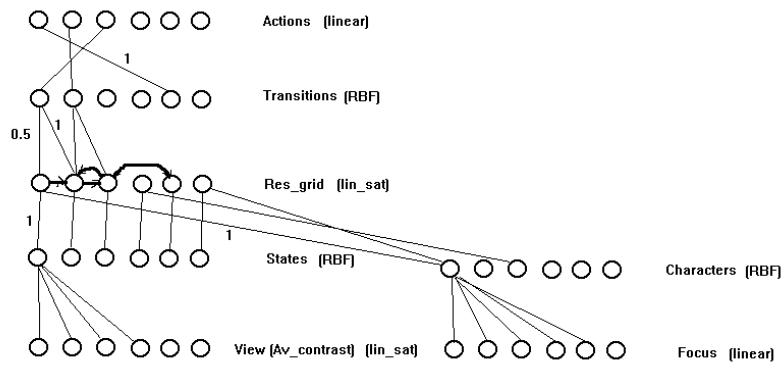

Figure 2: Diagram of the spatial memory model. The type of neurons used in each layer are indicated and the weights of the connections. This network is built during exploration and increases in size as the spatial knowledge increases. See details in the text.

in which they are observed so that they can activate the corresponding state nodes as part of the planning process described further below (section 3.5).

Due to a peculiarity of RBF nodes, the association between characters and states needs to be done via a second layer of node: The output y_i of a RBF node (1) is significantly activated only when all its inputs x_j have an activity close enough to the learned activations w_{ij}, typically much closer than the half width σ (see Appendix B).

$$y_i = \exp(-\frac{\sum_j (x_j - w_{ij})^2}{2\sigma_i^2}) \tag{1}$$

Therefore an active "character" node would not alone be able to activate a state node and a view would not activate a state node if the associated character was not activated beforehand. For this technical reason, a second layer of nodes is used which is a copy of the "states" layer (see figure 2). This layer is the "resistive_grid". It has nodes with linear-saturating transfer functions:

$$\begin{aligned} y_i = & \quad \textstyle\sum_j x_j w_{ij} & if\ 0 < \sum_j x_j w_{ij} < 1 \\ y_i = & \quad 0 & if\ \sum_j x_j w_{ij} < 0 \\ y_i = & \quad 1 & if\ \sum_j x_j w_{ij} > 1 \end{aligned} \tag{2}$$

The control of this system in figure 2 is not implemented using connectionist techniques. A relatively complex procedure written in a Basic-type language determines when given layers of the neural network are updated. Decisions like switching to an exploratory behaviour are based on tests done on the activity of nodes at given times. More details on the robot-user interaction scenario built into the procedure are given in section 4.2.

A purely connectionist implementation of the procedure would represent an extremely complex neural network. The case of the planning procedure is discussed in the appendix C where it is suggested that the use of spiking neurons may simplify greatly the control circuit. In contrast, the present connectionist implementation involves neurons with graded responses or binary gates.

3.4 A constructive learning procedure for states and actions

How are views and actions selected fro building a view graph ? This is the problem of exploration and novelty detection which has not yet been solved in a conceptually satisfactory way. In the model of Yamauchi and Langley (1996), new state nodes are created when the robot has moved a given distance from the last known position. In (Bachelder and Waxman, 1995), and (Recce and Harris, 1996), the space is divided into regular squares and it is attempted to create only one place node per square. In (Franz et al., 1996) novelty is determined by the response level of nodes. None of these methods is optimal. It has been suggested that predictive mechanisms, possibly based on egomotion, could be used to detect new (defined as "unpredicted") situations and drive the learning of new states (Denham and McCabe, 1995, 1996). We have experimented with an approach based on the principle of maximum information gain (see a discussion in Franz et al., 1996), in which the preferred next action was one that had never been executed in the current state. This was too time-consuming to be practical (section 4.2). Consequently, we simply used repeated displacements of small amplitude. New views also appeared during execution of a task (section 4.2)

A number of the connections in the network in figure 2 are constructed during exploration or during the task, to encode new knowledge regarding {state - letters} associations and {state pairs - action} associations. Thereby, the size of the neural network increases with experience. For most of the existing connections, the weights are set during the task. Only the connection weights between the states and the resistive grid are preset to 1.

When the robot meets its first scene, the first node in "state" is trained to recognise that scene (with the RBF nodes used here, this corresponds simply to set the input weights equal to the input activity pattern). If a letter is also found in the scene, a connections is created between the corresponding character node and the node in the resistive grid corresponding to the current state, with weight 1. When an action results in a new scene to be found, the next state node is recruited and trained for this new scene. A scene is considered to be new if no state node responds with an amplitude above a certain threshold (see Appendix B).

A connection from the first node to the second node in the resistive grid is made, with weight 0.1. These two nodes are also connected to a transition node with weights 0.5 from past state-node and 1 from current-statenode (transition nodes are of RBF type with $\sigma = 0.3$). Finally, the transition node is connected to the node coding for the action just performed, with weight 1.

The connections from resistive grid to action nodes are designed in such a way that, after the successor of the current state is identified by the planning process (see below), setting the activations of the res_grid node corresponding to the current state to 0.5 and that of its successor to 1 leads to the activation of the action known to have caused the transition between the two states. The connections in the resistive grid may be bidirectional, if there are corresponding actions. With lateral weights of 0.1, repetitive updating of the grid does usually not lead to a saturation of the activities.

At early stages of the design of the system, a N_s x N_s matrix was used to encode the actions linking any two pairs of the N_s possible states[17] . While experimenting with the robot, it turned out that several action could cause the same transition. For instance, a "forward" command could result in a leftward rotation if one of the wheels was blocked by some small obstacle on the table. This would erase the previous information stored in the matrix, because the "left" motion previously encoded as linking the current view to the view to its left would be replaced by the "forward" motion. It was therefore necessary to enable the encoding of several possible actions as linking two states. The constructive encoding procedure proposed here enables several transition nodes to be assigned to the same pair of states.

A future refinement of the encoding should reflect the probability that an action causes a given transition between two states, requiring to develop a neural version of Q- learning (Watkins and Dayan, 1992) for planning.

3.5 Planning and search procedure

In the task assumed here, letters are objects that the robot may be asked to retrieve. As part of the planning procedure, the node in "character" corresponding to the target letter is set to an activity of 1. This activates the nodes in the resistive grid corresponding to the states where the letter has been observed. In addition, the node in the resistive grid corresponding to the current state is inhibited, with an activity set to zero.

For planning, the resistive grid is then updated several times, until a stable distribution of activations is obtained (Bugmann et al., 1995). If the current-state node has a neighbour with non-zero activity, this indicates that there is a path leading to one of the states where the target letter has been observed. Otherwise, the program assumes that no path is known and enters into an exploratory mode which, at present, consists in asking the user for a rotation direction.

If during this exploration, a state is encountered that is part of a path linking to the goal, exploration is abandoned and plan-driven motion is resumed.

Any new state encountered is encoded in a new state node and new links are created to transition nodes. Any new letter encountered is learned and a link is formed between the corresponding character nodes and the current state node in the resistive grid.

[17] Despite the constructive principle, the size of the array had to be declared in advance. We used $N_s = 100$.

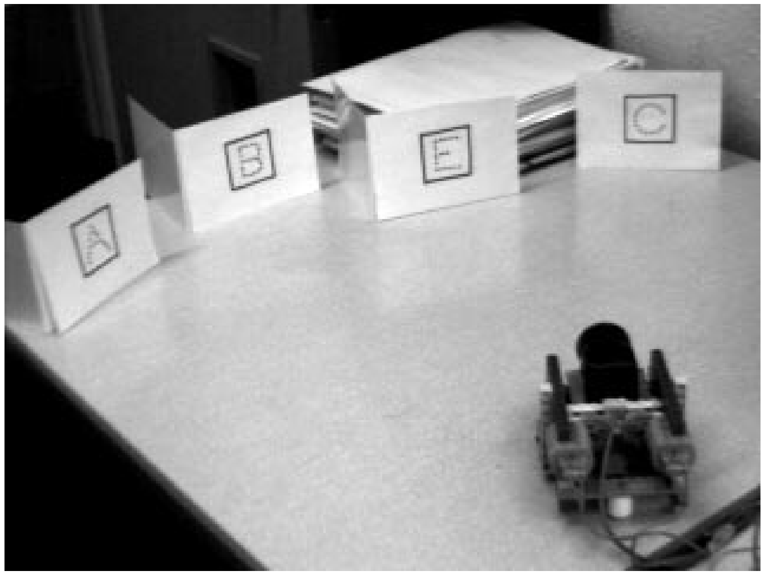

Figure 3: Example of the experimental set-up. A robot made of LEGO components carries a VVL video camera with a viewing angle of 30 degrees and faces a number of "objects" represented by letters surrounded by a frame. These are drawn on cards of size A6.

4 An experiment with a robot

4.1 Task and experimental set-up

The simple task designed for this robot is analogous to the task in which a household robot has to fetch a named object. As our robot has only a video camera and no gripper, it is assumed that "looking at" the object, i.e. localising it in the image, is equivalent to grip it. To simplify the vision aspect of the problem, "objects" are letters drawn cards of size A6 and placed on a table (figure 3).

The initial tests reported here use only rotation movements. Spatial memory is tested by asking the robot to "look at" a given letter by rotating towards it from any initial orientation. A maze-like set-up is planned for future experiments. Therefore, the artificial spatial memory system is designed to be compatible with translation movements.

The Basic-type program that controls the operation of the network in figure 2 was written to realise following scenario:

A robot is asked to retrieve a given object (character).

If the character is unknown, ask the operator to place a sample of that character in front of the camera, and then learn the character.

If the current view is unknown, create a new view-node and learn the view. If there is no path to the character, ask the operator for a direction in which to start searching.

If the character is known and there is a path, perform planning, execute the first step of the plan, recognise the new view and perform planning again, until the goal character is in sight.

Such a procedure results in interesting interactive demonstrations (see below) where the robot starts without knowledge of either the shape of characters or the appearance of the experimental set-up, and ends being (hopefully) able to execute the command of looking at a character from any starting position. As shown below, most experiments actually failed at the final planning stage, but that helped us to better understand view-based spatial memory models and propose improvements (section 5)

4.2 Learning and planning test

Exploration:

The experimental set-up was similar to the one in figure 3, except that letters were distributed over approximately 120 degrees. Exploration was done by successive steps of leftwards or rightwards rotation. Initially a "information maximising" procedure was designed, by which the next chosen action was the one for which no outcome was known (no corresponding link). However, this led to time consuming random-looking moves. Therefore, for these initial tests, simple rightwards and leftwards sweeps were used. Each time an exploration procedure was required, the robot asked the operator for a direction of rotation.

To build the spatial memory, the robot was first pointed towards the most rightwards letter, a "C" and was asked, by calling the relevant subroutine in the neural network simulation environment CORTEX-PRO, to "look at" "A". The robot first noted that "A" was not known and asked for an example to be shown. It then noted that there was no path connecting "A" to any of the nodes in the resistive grid and asked for a direction in which to rotate to "explore" the environment. It then created a view node coding for the current scene and found a letter in the scene that it did not know. It asked the name of this letter which it then linked to the first state node. It then rotated leftwards by approximately 15 degrees, found that the view was unknown and recruited the second state node to code for it. A link between the two corresponding nodes in the resistive grid was created and a transition neuron recruited to encode the "left" action linking the states. The letter "C" was also visible in that scene and was linked to the state 2. Figure 4 illustrates the history of this exploration. A chain of states 1 to 10 was generated until the goal letter "A" was observed.

The robot was then asked to "look at" C. He knew how "C" looked like, and found that it was linked to the resistive grid. It started the planning routine but could not activate any neighbouring node of state 10 (remember that the connections are in the direction opposite from the one of the arrows drawn in the figure 4). Consequently the robot asked again for an exploration direction,

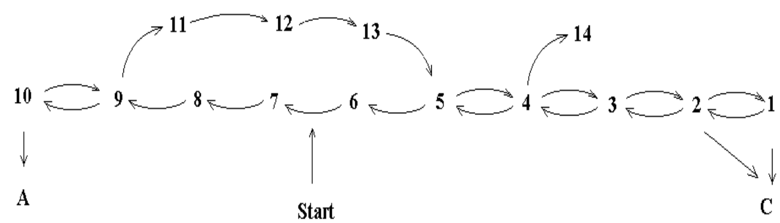

Figure 4: Sequence of states constructed during leftward exploration (1-10), rightward exploration (11-13) and during planning (14). Each state is represented by a node in "states" in figure 2. The arrows indicate the direction of rotation and the opposite direction of the links built into the network (for backward planning, the activity must spread from the goal to its predecessors).

which was given as "left". The first view encountered corresponded to view 9 and a "left" link was created between node 10 and 9. At the next step, for some reason, the node 8 was not activated strongly enough and the robot recruited a new node for a new view 11. A succession of 3 new state nodes were created in that manner, until the state node 5 recognised its view. The robot recognised then all expected views and created "right" links between views 5, 4, 3, 2 and 1. Figure 5 shows the computer screen at the end of an exploration process. The characters "C" is in sight and the number of connections reflects the acquired spatial knowledge.

Planning:

The robot was then rotated by hand to face a direction indicated by "Start" in the figure 4 and was "told" to "look at" "A". The recognised view was view 12. The planning procedure indicated the state 13 as the next one to be reached and a rightwards movement was executed. The next view was recognised as view 6. The planning procedure indicated state 7 as the next state and caused a leftwards movement. Then view 12 was recognised and a rightwards movement occurred. This repeated itself several times: The robot was stuck in a local minima! Such a problem has not been mentioned before.

Another problem that was found is the creation of state 14. When the start position faced view 5 and the goal was the letter "C", there was no possibility for a local minima in the structure of the spatial memory. However, after two rightward steps, the state 3 was not recognised and a new state 14 was created. As there was no known path from this state to the states 1 or 2, where "C" is found (or to any other states), the robot decided to initiate an exploration procedure.

Finally, the aliasing problem occurred frequently. It occurred that different views appeared sufficiently similar to be assigned the same node during exploration, which resulted in a view-graph with erroneous links. During planning also, a view could be recognised by the wrong node in figure 2 which led to an inadequate action plan.

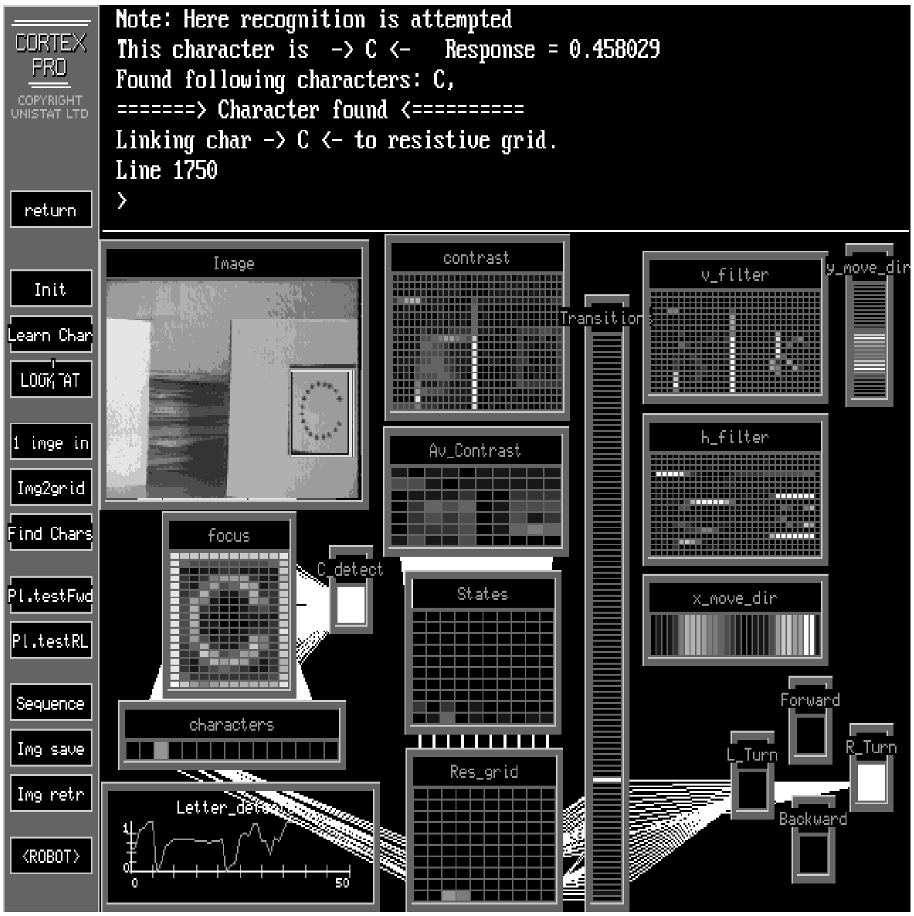

Figure 5: Screen showing the vision and spatial memory system (A pile of papers is behind the "C"). The grids have following X x Y sizes: Av_Contrast: 10 x 7, Contrast: 30 x 18; States and Res_grid: 10 x 10, Transitions 1 x 100, Characters: 15 x 1, Focus: 12 x 18; v_filter and h_filter: 30 x 18; x_move_dir: 30 x 1; y_move_dir: 1 x 30. The simulations are done using the package CORTEX-PRO. The operations of the systems vision and state recognition systems is described in details in the appendices A and B.

5 Discussion

In the experiment described above, a form of local minima occurs, resulting in repeated left and right rotations. A second problem is the occurrence of a representational dead- end when a new state is created. A third problem is aliasing. The experiments used only rotations "on the spot" possibly similar to eye saccades, but the observed problems are general and can also occur during translations.

Local minima, disjoint paths and internal path integration:

The causes of the local minima problem may lie in the existence of disjoint paths in figure 5 and/or in the planning process.

Disjoint paths can arise when the state space is not densely covered during exploration. In our case, as exploration is not done with very small rotation steps, "holes" in the graph do occur. These allow a second disjoint path to be built during a later exploration sweep. Disjoint paths are probably a characteristic property of view-based graphs, and can be avoided only when a very comprehensive exploration process is practically possible (and affordable in terms of memory costs).

Considering planning, if a planned path is followed step by step, interferences with other paths built during exploration may occur. That is what causes the local minima problem. It also leads to a peculiar behaviour of reproducing the exploratory movements to reach a goal. Normally, humans or monkeys tend to use direct trajectories.

This behaviour and the local minima problem can be avoided if the successive steps of the plan are integrated internally before being executed. In the case shown in figure 4, a single rotation to the target would result, whether state 6 or state 12 is recognised at the start. Exploration would then not need to be done in regular sweeps or in small steps. Thus neighbourhood in the real world would not necessarily need to correspond to neighbourhood in the view graph, although this might be corrected during off-line reorganisation (see below).

If a plan contains combinations of translations and rotations, as for mazes, successions of rotations "on the spot" and successions of collinear translations need to be integrated and executed separately. If these integrated movements were also stored in the view graph, the planning process would progressively contain less internal steps (or "detours"), at least for frequently used routes.

The different practical constraints existing on rotations and translations may affect the final organisation of a view-graph. One view can be linked to many other views by simple rotations on the spot, while a translation can link a view to only one other destination view. As a result of the integration process, view graphs may thus become enriched in links encoding rotations from given points but have a more limited number of links encoding translations between views attached to these points (see also Rieser, 1989).

Some thought should also be given to the practical meaning of "translation". Robots can not keep moving in straight line for a long time. It may be more practical for them to head towards some landmark, which would then provide visual feedback on the quality of the trajectory. Thus "translations" would become "heading procedures" of a certain complexity, very different in nature of simple ballistic rotations on the spot. Thus there might be a good case for using two different action-encoding schemes.

Representational dead-ends and visual recognition of distant places:

When the new state 14 is created, the encoded link and action only carries information on how to reach state 14. There is no information on how to reach the rest of the graph from state 14.

It is tempting to solve this problem by using bi- directional links. However, not all actions are reversible. There are one-way roads and exit doors.

Another approach is to use vision in a more natural way, for recognising distant places and inspect possible routes to these places. This may involve the extraction of the 2-D layout of the space ahead (see e.g. Onillon et al, 1995), then a "virtual displacement" on this "local map" to a previously visited place and the establishment of new links between the current view and a view seen from a position reached by this virtual displacement. Thus a temporary map of limited size may be needed.

A less geometric approach would require the visual system to perform size-invariant view recognition. This may enable the robot to recognise a distant view, and encode a link possibly associated with a heading-based translation procedure (as introduced above), using a feature of the distance view as a goal. In any case, more work on the visual component of the model is needed.

Aliasing problem and priming

Another problem found during the experiments was the aliasing problem (Whitehead and Ballard, 1991). Typically, in maze situations, identical local configurations occur in different positions. Thus a purely view based system does not provide unequivocally the position and orientation information needed for route planning. This is the classical reconstructibility problem in control theory. The solution usually proposed is to use the history of the measurements to define the current state (see e.g. Rao and Fuentes, 1996). Using this approach in our experiments, a view would be recognised as an image <u>and</u> a sequence of previously recognised views. However, if a surrounding is explored with random visual saccades, it is unlikely that any useful sequence may be stored.

Alternatively, if the robot has a purposeful behaviour, then any view is the goal of the previous movement, and it could be primed as such (see also Denham and McCabe, 1995, 1996). Even random motion may predict views if the connectivity in the view graph is sufficiently dense. In practice, priming would be done via back-projections from the action-encoding part of the network to the state layer in figure 2. Further work along these lines is needed.

Memory requirements, landmarks, and off-line graph reorganisation:

The upper part of the network in figure 2 is a partial inverse model of the effect of actions on views, constructed progressively during experience. As one node represents each transition, the inverse model has a potential for using a large memory space.

A related potential problem with our approach is the increasing number of state-nodes when the experience of the robot increases. For instance, assuming that a robot sees a new unknown scene every second, it will build 2.6 million view-nodes in a month (To these must be added the corresponding transition nodes). If the views are encoded at a low level, by the intensity of each pixel of a video image, a month-long experience would represent 600 GBytes of memory[18]. Most robots would probably not experience such a fast changing and unpredictable environment. Nevertheless, some care is needed to maintain the memory size within manageable limits.

One way to a reduce the memory load is to reduce the number of encoded perceptual states and keep only those that are really useful. One approach might involve attentional mechanisms or novelty detection (Denham and McCabe, 1995, 1996). Alternatively, visually salient features or landmarks (Thrun, 1996) could be used as focus points around which views are memorised, similarly to (Gaussier et al., 1997).

In addition, the problem of deciding if a landmark is "salient" or "memorable enough" may be solved automatically during an off-line "cleanup" process. For instance, if a simple ongoing action or procedures can be discovered to link distant views, then intermediate views may become redundant, and eliminated according to principle v) in section 3.1. Eventually, the only surviving nodes may be those shared by several route and those where low-level procedures fail to produce the adequate action.

Other models:

The model presented here is functionally similar to that used in mobile robot experiments by Mallot et al., (1995). However, the experiments reported here were done under different conditions, which revealed problems that Mallot et al, (1995) could not have observed. For instance, Mallot et al., 1995 did not use vision to recognise views, but bar-code signs on the floor which were read during translation towards a junction. The maze was hexagonal so that there were only 3 possible actions: "rotate left 60 degree then advance until junction", "rotate right 60 degree then advance until junction" or "reverse until junction". During translation, the robot was following the corridor linking two junctions. Neither aliasing, nor disjoint paths, nor representational dead ends could occur.

In Franz et al., (1996), the coverage of the space was dense enough to enable the robot to "sense" neighbouring views and test, by moving towards the position of the neighbouring view, if a link could be added to the memory. Aliasing was the only possible problem.

A model of biological spatial memory ?

How does the model compare to biological spatial memory ? The model conforms to the idea that spatial memory may take the form of a network of routes. During exploration, views and the action linking them are encoded, which defines a route in the usual sense. If views are common between two route, a network of route forms automatically, as observed by Moar and Carleton (1982).

A feature that needs to be added is the encoding of travel distance or time. In conjunction with the proposed internal integration, the model may then be used to estimate directions and distances. If all actions of a plan were integrated using a form of internal dead reckoning, this would result in an estimation of the straight line distance to the goal and its direction. This information could also be added to the view graph, but the associated action should then be a straight-line flight ! Mataric (1990) proposed a similar method for finding shortcuts. Thus, the evolution of the capabilities of the model with increasing

[18] Assuming an image with 320x240 pixels, the color of each pixel being coded in 24 bits.

experience may parallel a similar evolution in human subjects, from sequence specific knowledge to vector-like knowledge.

Finally, it is noted that:

- Due to the structure of the model, comprising views and displacements, it may be possible to add nodes and links to this spatial memory by using verbal route instructions. This would be facilitated by the introduction of "heading" translation procedures and size independent landmark / feature recognition.
- One problem that needs to be considered is planning with constraints. To include constraints on the type of places that are part of a route, it may be beneficial to use a biologically plausible implementation (Appendix C). However, to include constraints on the type of action (e.g. vehicle type, or to avoid the "flight-links" mentioned above), some restructuring of the model may be needed, possibly using feedback for priming view nodes.
- Real world experiments submit the model to constraints such as noisy sensory information, unreliable actuators and timing constraints. Such constraints affect the design of a model. They have for instance led to a modification of the action encoding scheme described in section 3.4.

6 Conclusion

The original motivation of the presented model was its potential for avoiding Cartesian maps, avoiding the dimensionality curse and avoiding complex visual information processing.

However, robot experiments revealed a number of problems that were neither readily predictable from the principles formulated in section 3.1, nor from the connectionist structure described section 3.3, nor from the procedures in sections 3.4 and 3.5, nor from the literature in the field. Consequently, a number of modifications to the model are proposed.

To eliminate representational dead-ends, it is proposed to use vision for remote recognition of places and assessment of the route to that place, so that links can be established between the current view and a view from the remote place. Technically, this may require to replace the concept of "photographic" view by that of group of visual features linked by geometrical relations. Concurrently, a procedure for "heading towards a visual feature" needs to be added to the repertoire of translation actions.

To avoid local minima during planned movements, it is proposed to use internal integration to plan rotation movements. Such a mechanism may also enable off-line restructuring of the view-graph so that an "inactive" robot may progressively refine the content of its spatial memory.

To reduce the aliasing problem, it is proposed to investigate mechanisms for priming view nodes by actions, in a predictive way.

The proposed solutions still avoid Cartesian maps and the dimensionality curse, but call for an increase in the complexity of the the visual system, the planning system and the action representation.

Acknowledgements:

This work has greatly benefited from discussions with Phillipe Gaussier. I gratefully acknowledge comments by Mike Denham, Philippe Gaussier and Loukas Michalis on earlier versions of the manuscript. I am grateful to Davi Vann Bugmann who provided pointers to psychological work on spatial memory. Earlier discussions with John G. Taylor, Chris Hindle and the Neural and Adaptive Systems Group have allowed to set the scene for this investigation. Many thanks to Alan Simpson for his support.

References

[1] Althoefer K. (1996) "Neuro-Fuzzy Path Planning for Robotic Manipulators", PhD Thesis, Dept. of Electronic and Electrical Engineering, King's College London, Strand, London WC2R 2LS, UK

[2] Altman J. (1995) "Deciding what to do next", Trends in Neuroscience, 18, 117-118.

[3] Arbib M.A. and Lieblich I. (1977) "Motivational Learning of Spatial Behaviour", in Metzler J., (ed) "Systems Neuroscience", Academic Press, New York, pp. 221-239. (a more accessible reference may be Lieblich and Arbib, 1982)

[4] Bachelder I.A. and Waxman A.M. (1995) "A View-Based Neurocomputational System for Relational Map-Making and Navigation in Visual Environments", Robotics and Autonomous Systems, 16, pp. 267-289.

[5] Barto A.G., Bradtke S.J. and Singh S.P. (1995) "Learning to Act using Real-Time Dynamic Programming", Artificial Intelligence, 72, 81-138.

[6] Blades M. (1990) "The Reliability of Data Collected from Sketch Maps", J. of Environmental Psychology, 10, pp. 327339.

[7] Brooks R.A. (1986) "A Robust Layered Control System for a Mobile Robot", IEEE Journal of Robotics an Automation, RA2, pp. 14-23.

[8] Buehlmeier A., Duerer H., Monnerjahn J., Noelte M. and Nehmzow U. (1996) "Learning by Tuition, Experiments with the Manchester "FortyTwo"", University of Manchester, Computer Science Technical Report Series, UMCS-96-1-2, Can be downloaded from: ftp://ftp.cs.man.ac.uk/pub/TR/UMCS-96-1- 2.ps.Z

[9] Bugmann G., Taylor J.G. and Denham M.J. (1994) "Sensory and Memory Based Path Planning in the Egocentric Reference Frame of an Autonomous Mobile Robot" Research Report NRG-94-01, School of Computing, University of Plymouth, Plymouth PL4 8AA, UK. http://www.tech.plym.ac.uk/soc/research/neural/neur20/pub licat/papers/nrg9401.zip

[10] Bugmann G., Taylor J.G. and Denham M.J. (1995) "Route finding using Neural Nets", in Taylor J.G (ed) "Neural Networks", Alfred Waller Ltd, Henley-on-Thames, pp. 217-230.

[11] Bugmann G. (1997) "Biologically Plausible Neurocomputation", Biosystems, 40, 11-19.

[12] Burgess N. Recce M. and O'Keefe J. (1995) "Hippocampus: Spatial Models", in Arbib M.A. (ed) "The handbook of Brain Theory and Neural Networks", Bradford Books/MIT Press, pp. 468-472

[13] Burgess N. and O'Keefe J. (1996) "Neural Computation Underlying the Firing of Place Cells and their Role in Navigation", Hippocampus, 6, pp. 749-762.

[14] Byrne R. (1979) "Memory for urban geography", Quarterly Journal of Experimental Psychology, 31, 147-154.

[15] Connoly C.I. and Burns J.B. (1993) "A Model of the Functioning of the Striatum", Biological Cybernetics, 68, pp. 535-544.

[16] Cox I.J. (1991) "Blanche - An experiment in Guidance and Navigation of an Autonomous Robot Vehicule", IEEE Transactions on Robotics and Automation, 7, pp 193-204.

[17] Curran T. (1995) "On the neural Mechanisms of Sequence Learning", Psyche, Vol. 2 (on-line journal: http://psyche.cs.monash.edu.au/volume2-1/psyche-95-2-12 sequence-1-curran.html)

[18] Denham M.J. and McCabe S.L. (1995) "Robot control using temporal sequence learning", Proc World Congress on Neural Networks (WCNN'95), Washington DC, USA, July 1995, INNS Press/Lawrence Erlbaum Associates, vol 2, pp 346-349.

[19] Denham M.J. and McCabe S.L. (1996) "Biological basis for a neural model of learning and recall of goal-directed sensory-motor behaviours", Proc World Congress on Neural Networks (WCNN'96), San Diego, USA, September 1996, INNS Press/Lawrence Erlbaum Associates, pp. 1283-1286.

[20] Evans G.W., Marrero D.G. and Butler P.A. (1981) "Environmental Learning and Cognitive Mapping", Environment and Behaviour, 13, pp. 83-104.

[21] Ferguson E.L. and Hegarty M. (1994) "Properties of Cognitive Maps Constructed from Texts", Memory and Cognition, 22, pp. 455-473.

[22] Franz M.O., Schölkopf B., Georg P., Mallot H.A., and Bülthoff H.H. (1996) "Learning View Graphs for Robot Navigation", Technical Report no 33, Max-Planck Institute für Biologische Kybernetik, 72076 Tübingen, Germany. Can be downloaded from the website: http://www.mpik tueb.mpg.de/projects/techr/list3.html

[23] Gaussier P. and Zrehen S. (1995) "PerAc: A neural Architecture to Control Artificial Animals", Robotics and Autonomous Systems, 16, pp. 291-320.

[24] Gaussier P. and Zrehen S. (1994) "A topological Neural Map for On-Line Learning: Emergence of Obstacle Avoidance in a Mobile Robot", Proc SAB'94 (Brighton, UK), pp. 282-290.

[25] Gaussier P., Joulain C., Revel A. and Banquet J.-P. (1996) "Are Shaping Techniques the Correct Answer for the Control of Visually Guided Robots ?", Proc. Control'96 (Exeter, UK), IEE Publ. London, pp. 1248-1252.

[26] Gaussier P., Joulain C., S. Zrehen, Banquet J.-P. and Revel A. (1997) "Visual Navigation in an Open Environment without Map", Submitted to IROS'97, Grenoble.

[27] Glasius R., Komoda A. and Gielen S. (1996) "A biologically Inspired Neural Net for Trajectory Formation and Obstacle Avoidance", Biological Cybernetics, 84, pp. 511-520.

[28] Gray J.A. (1995) "The contents of consciousness: A neurophysiological Conjecture", Behavioural and Brain Science, 18, pp. 659-676

[29] Hull C.L. (1932) "The goal Gradient Hypothesis and Maze Learning", Psychological Review, 39, pp. 25-43.

[30] Hull C.L. (1934) "The Concept of Habit-Family Hierarchy and Maze Learning", Psychological Review, 41, pp. 33-52.

[31] Hull C.L. (1935) "The Mechanism of the Assembly of Behavior Segments in Novel Combinations suitable for Problem Solution", Psychological Review, 42, pp. 219-245.

[32] Huttenlocher J. and Newcombe N. (1984) "The Child's Representation of Information about Location", in Sophian C. (ed) "Origin of Cognitive Skills", Lawrence Erlbaum Associates, NJ, pp. 81-111.

[33] Jeannerod M. (1994) "The Representing Brain: Neural Correlates of Motor Intention and Imagery", BBS, 17, pp. 187- 202.

[34] Joulain C., Gaussier P. and Revel A. (1996) "Apprentissage de Catgories Sensory Motrices par un Robot Autonome" Research Report ENSEA/ETIS, Paris. Can be downloaded from: http://bunny.ensea.fr/Pages_Perso/Arnaud_Revel/articles/nsi2.ps

[35] Kohonen T. (1988) "Self-Organisation and Associative Memory", Springer Verlag, Berlin

[36] Kortenkamp D., Huber M., Cohen C., Raschke U., Bidlack C., Bates Congdon C., Koss F. and Weymouth T. (1993) "Integrated Mobile-Robot Design (Winning the AAAI'92 Robot Competition)", IEEE Expert, 8, pp. 61-73.

[37] Latombe J.C. (1991) "Robot motion Planning", Kluwer Academic Publ.

[38] Lejeune H. and Wearden J.H. (1991) "The Comparative Psychology of Fixed-Interval Responding: Some Quantitative Analyses", Learning and Motivation, 22, pp. 84-111.

[39] Levine M., Jankovic I.N., Palij M. (1982) "Principle of Spatial Problem Solving", Journal of Experimental Psychology: General, 111, pp. 157-175.

[40] Lieblich I. and Arbib M.A. (1982) "Multiple Representations of Space", The Behavioral and Brain Sciences, 5, 627-659.

[41] Maes P. (189) " How to do the right Thing", Connection Science, 1, pp. 291-323.

[42] Maguire E.A., Frackowiak, R.S.J. and Frith C.D. (1997a) "Learning to find your way - a role for the human hippocampal region", submitted.

[43] Maguire E.A., Burgess N., Donnett J.G., O'Keefe J. and Frith C.D. (1997b) "Knowing where things are: Parahippocampal involvement in encoding object locations in virtual large-scale space", Submitted.

[44] Mallot H.A., Bülthoff H.H., Georg P., Schölkopf B. and Yasuhara K. (1995) "View-Based Cognitive Map Learning by an Autonomous Robot", Proc. ICANN'95 (Paris), Vol. 2, pp. 381- 386. A number of papers related to this work can be retrieved at the following website: http://www.mpik tueb.mpg.de/projects/navigation/navigation.html

[45] Mataric M.J. (1990) "Navigating with a Rat Brain", Proc. of Simulation of Adaptive Behavior: From Animals to Animats (SAB'90), MIT Press Bradford Books, pp. 169-175.

[46] Mellet E., Tzourio N., Denis M., and Mazoyer B. (1995) "A positron Emission Tomography Study of Visual and Mental Spatial Exploration", J. of Cognitive Neuroscience, 7, pp. 433-445.

[47] Mellet E., Tzourio N., Crivello F., Joliot M., Denis M., and Mazoyer B. (1996) "Functional Imagery of Spatial Mental Images Generated from verbal Instructions", J. of Neuroscience, 16, pp. 6504-6512.

[48] Menzel E.W. (1973) "Chimpanzee Spatial Memory Organisation", Science, 182, pp. 943-945.

[49] Moar I. and Carleton L.R. (1982) "Memory for routes", Quarterly Journal of Experimental Psychology, 35, pp. 381-394.

[50] Muller R.U., Stead M. and Pach J. (1996) "The Hippocampus as a Cognitive Graph", J. Gen. Physiol., 107, pp. 663-694.

[51] McNamara T.P., Hardy J.K. and Hirtle S.C. (1989) "Subjective Hierarchies in Spatial Memory", J. of Experimental Psychology: Learning, Memory and Cognition, 15, pp. 211-227.

[52] McNaughton (1989) "Neural Mechanisms for Spatial Computation and Information Storage", in: Nadel L., Cooper L.A., Culicover P. and Harnish R.M. (eds) "Neural Connections, Mental Computation", MIT Press, Cambridge MA, USA, pp. 285- 350.

[53] McNaughton B.L. and Nadel L. (1990) "Hebb-Marr Networks and the Neurobiological Representation of Action in Space", In Gluck .M.A. and Rumelhart D.E. (eds) "Neuroscience and Connectionist Theory", Lawrence Erlbaum Associates, Hillsdale NJ, USA, pp. 1-63.

[54] Newell A. and Simon H.A. (1972) "Human problem Solving", Prentice Hall, Inc, Englewood Cliffs, NJ

[55] O'Keefe J. and Nadel L. (1978) "The Hippocampus as a Cognitive Map", Clarendon Press, Oxford.

[56] O'Keefe J. and Speakman A. (1987) "Single Unit Activity in the Rat Hippocampus during a Spatial Memory Task", Experimental Brain Research, 68, pp. 1-27.

[57] O'Keefe J, (1991) "An Allocentric Spatial Model for the Hippocampal Cognitive Map", Hippocampus, 1, pp. 230-235.

[58] Onillon V., Bugmann G., Simpson A. and Nurse P. (1995) "Artificial vision for micromouse" Research Report NRG-95-05, School of Computing, University of Plymouth, Plymouth PL4 8AA, UK, can be downloaded from http://www.tech.plym.ac.uk/soc/research/neural/

[59] Peruch P., Giraudo M-D. and Gäling T. (1989) "Distance Cognition by Taxi Drivers and the General Public", J. of Environmental Psychology, 9, pp. 233-239.

[60] Presson C., DeLange N. and Hazelrigg M. (1989) "Orientation Specificity in Spatial Memory: What Makes a Path Different From a Map of the Path ?", Journal of Experimental Psychology: Learning, Memory, and Cognition., 15, pp. 887- 897.

[61] Rao R.P.N. and Fuentes O. (1996) "Learning Navigational Behavior using a Predictive Sparse Distributed Memory", To appear in "Proc. from Animals to Animats'96: The Fourth International Conference on Simulation of Adaptive Behavior", MIT Press

[62] Recce M. and Harris K.D. (1996) "Memory for places; A navigation model in support of Marr's Theory of hippocampal function", Hippocampus, 6, pp. 735-748.

[63] Rieser J.J. (1989) "Access to Knowledge of Spatial Structure at Novel Point of Observation", Journal of Experimental Psychology: Learning, Memory, and Cognition, 15, pp. 1157- 1165.

[64] Röfer T. (1997) "Controlling a wheelchair with image based homing", in N. Sharkey and U. Nehmzow (eds.) "Spatial Reasoning in Mobile Robots and Animals", University of Manchester Technical Report UMCS-97-4-1

[65] Rolls E.T. (1996) "The representation of space in the primate hippocampus, and its role in memory", in Ishikawa K., McGaugh J.L. and Sakata H. (eds) "Brain Processes and Memory, Elsevier: Amsterdam, pp. 203-227.

[66] Sadalla E.K. and Magel S.G. (1980) "The perception of travelled distance", Environment and Behaviour, 12, pp. 65- 79.

[67] Schmajuk N.A., Thieme A.D. and Blair H.T. (1993) "Maps, Routes and the Hippocampus: A neural Network Approach", Hippocampus, 3, pp. 387-400.

[68] Siemiatkowska B. (1994) "Cellular Neural Networks for Mobile Robot Navigation", Proc. 3rd Int. Conf. on Cellular Neural Networks and their Applications (CNNA94), Rome, Italy, pp. 285-290.

[69] Shallice T. (1988) "From Neuropsychology to mental Structure", Cambridge University Press, Cambridge, UK. (see section 14.6 on planning and frontal lobe lesions).

[70] Schölkopf B. and Mallot H.A. (1994) "View-Based Cognitive Mapping and Path Planning", Technical Report no 7, MaxPlanck Institute für Biologische Kybernetik, 72076 Tübingen, Germany. Can be downloaded from: http://www.mpik-tueb.mpg.de/projects/techr/list1.html Also availbable as: Schölkopf, B. and H.A. Mallot (1995) "View-based cognitive mapping and path planning.", Adaptive Behavior 3, pp. 311-348.

[71] Speakman A. (1987) "Place cells in the brain: Evidence for a cognitive map", Sci. prog., Oxf., 71, pp. 511-530.

[72] Spence K.W. (1950) "Cognitive Versus Stimulus-Response Theories of Learning", Psychological Review, 57, pp. 159-172.

[73] Tesauro G.J. (1995) "Temporal difference learning and TD Gammon", Communication of the ACM, 38, pp. 58-68.

[74] Thomson J.A. (1980) "How do we use visual information to control locomotion", Trends in Neuroscience, October 1980, pp. 247-250.

[75] Thorndyke P.W. and Hayes-Roth B. (1982) "Differences in Spatial Knowledge Acquired from Maps and Navigation", Cognitive Psychology, 14, pp. 560-589.

[76] Thrun S. (1996) "A Bayesian Approach to Landmark Discovery and Active Perception in Mobile Robot Navigation", Technical Report CMU-CS-96-122, Carnegie Mellon University, Pittsburgh, PA.

[77] Thrun S. and Bücken A. (1996) "Learning Maps for Indoor Mobile Robot Navigation", Technical Report CMU-CS-96-121, Carnegie Mellon University, Pittsburgh, PA.

[78] Tolman E.C. (1948) "Cognitive maps in Rats and Men", Psychological Review, 55, pp. 189-208.

[79] Touretzky D.S., Wan H.S. and Redish A.D. (1994) "Neural Representation of Space in Rats and Robots", in Zurada J.M., Marks R.J. and Robinson C.J. (eds) "Computational Intelligence: Imitating Life", IEEE, pp. 57-68.

[80] Tversky B. (1992) "Distortions in cognitive maps", Geoforum, 23, pp. 131-138.

[81] Vann Bugmann D. (1995) "Human Spatial Memory: A Study of Cognitive Distance", Research report, Department of Psychology, University of Plymouth, UK.

[82] Verschure P.F.M.J., Kröse B.J.A. and Pfeifer R. (1992) "Distributed Adaptive Control: The Self-organization of Structured Behaviour", Robotics and Autonomous Systems, 9, pp. 181-196.

[83] Verschure P.F.M.J., Wray J. Sporns O., Tononi G. and Edelman G.M. (1995) "Multilevel Analysis of Classical Conditioning in a Behaving Real World Artefact", Robotics and Autonomous Systems, 16, pp. 247-265.

[84] Ward A.K., Newcombe N. and Overton W.F. (1986) "Turn Left at The Church, or Three Miles North: A Study of Direction Giving and Sex Differences", Environment and Behaviour, 18, pp. 192-213.

[85] Watkins C.J.C.H. and Dayan P. "Q-Learning", Machine Learning, 8, pp. 279-292.

[86] Webb B. (1995) "Using Robots to Model Animals: a Cricket Test", Robotics and Autonomous Systems, 16, pp. 117-134.

[87] Wehner R., Michel B. and Antonsen P. (1996) "Visual Navigation in Insects: Coupling of Egocentric and Geocentric Information", Journal of Experimental Biology, 199, pp. 129- 140.

[88] Whishaw I.Q. (1991) "Latent Learning in a Swimming Pool Place Task by Rats: Evidence of the Use of Associative and Not Cognitive Mapping Processes", The Quarterly J. of Experimental Psychology, 43B, pp. 83-103.

[89] Whitehead S.D. and Ballard D.H. (1991) "Learning to Perceive and Act by Trial and Error", Machine Learning, 7, pp. 45-83.

[90] Yamauchi B. and Langley P. (1997) "Place Learning in Dynamic Real-World Environments", Intl. Workshop on Learning in Autonomous Robots (RobotLearn'96), May 19-20, Keywest, Florida, pp. 123-129. (http://www.cs.buffalo.edu/~hexmoor/robolearn96/workshop.html). Possibly more accessible: Yamauchi B. and Langley P. (1997) "Place recognition in dynamic environments" J. of Robotics System, 14, pp. 107-120.

[91] Zimmer U.R (1996) "Robust World-Modelling and Navigation in a Real World", Neurocomputing, 13, pp. 247-260.

A Vision system for letter detection and recognition

This part of the vision system is described briefly. Its function is to precisely localise letters in the image to enable their recognition with a simple neural network (a single layer perceptron). The letters displayed in the environment are surrounded by a frame of 5 x 5 cm drawn in a full line. The letters are drawn with a dotted line. The local contrast of the image is calculated and copied into a grid of size 30x18 named "contrast" (figure 5). This coarse image is filtered with vertical and horizontal centre-ON local operators and copied to two grids "v_filter" and "h_filter". Average intensities are computed along respectively horizontal or vertical axis and copied into the layers "y_move_dir" and "x_move_dir".

Using the size of the regions of high activity in these layers, the position of the frame surrounding the letter is approximately determined. Two frames are created successively for each image. If none of them contains a letter, it is assumed that the image contains no letter. After a first frame has been found, the corresponding region in the grid "contrast" is reduced in intensity. This allows other regions to attract the "attention" of the localisation process for the second frame.

The content of a frame is copied to the grid "focus". As the initial video frame does not fit exactly the frame surrounding the letter, a routine shifts the video frame slightly until the frame in the image fall exactly on the most external nodes of the grid "focus". Therefore the letter to be recognised has always the same position in the grid, independently of the distance and angle of view. However, a positional jitter of the size of one pixel in the grid focus is still possible. To reduce the effect of fluctuation of intensity and of the jitter, the image is blurred artificially. It is then recognised by RBF nodes (see Appendix B) in the layer "characters". These nodes are only connected to the central region of "focus", where the relevant information lies. A two-node wide belt of nodes not used for recognition. This belt is connected to a node "C_detect" which responds maximally when a frame has been placed correctly. Its activity is plotted in the graph "letter_detector" in figure 5.

Recognition is accepted if a node in the layer "character" wins with an activity more than twice the activity of its nearest competitor, and with an absolute activity higher than 0.18. These settings are found by trial and error.

The number of letters that the systems knows is predefined, and each node in "Characters" is assigned to a letter. However, the appearance of a letter is not pre-encoded. Thus a detected character may be unknown to the system. This situation is characterized by a high activity of the node "C_detect" and an insufficient response in the layer "Characters". This triggers a learning procedure in which the program asks the user to identify the letter, then the weights of the corresponding node in "characters" are set so that the current visual pattern elicits a maximal response.

B Vision system for state recognition

The vision system is shown in figure 5. The part assigned to letter detection and recognition is described in the appendix A. This appendix focuses on the recognition of states based on the current view. To reduce the number of connections to the state nodes, the activity of the "contrast" grid is copied into a coarser grid named "Av_Contrast". Each node of that grid represents the average activity over a receptive field of relative area 0.12 x 0.12 in the grid "contrast" (of size 1 x 1). Thus blurring occurs in the transformation, in addition to resolution reduction. Blurring makes the recognition process more tolerant to translation but also reduces the discrimination between views. An more robust approach would be to make the recognition of a state explicitly shift invariant, e.g. by use of shift maps (Bugmann et al., 1994), which would allow a more selective view recognition method. State neurons are of RBF type, with a width of $\sigma = 0.75$, and a response y_i given by (3):

$$y_i = \exp(-\frac{\sum_j (x_j - w_{ij})^2}{2\sigma_i^2}) \tag{3}$$

where x_j is the activity of a node in the layer "Av_Contrast". A state is assumed to be recognised when the activity of one of the RBF-node in the grid "states" exceeds 0.25, for a maximum of 1. If a view is not recognised, a new node is recruited and trained to recognise the new state. This is done by copying the outputs x_j of the nodes in "Av_Contrast" into the "weights" w_{ij}. Note that in RBF nodes, weights indicate the centre of the receptive field and do not have their usual multiplicative role.

C Notes on a biologically plausible implementation

i) Parts of the system are not implemented using connectionist techniques. For instance, after the resistive grid has been updated, the next state is selected by scanning through the nodes connected to the current-state node, not all the nodes. Its activity is then set to 1 and that of the current state to 0.5. Using a purely connectionist design, a quite complicated circuit needs to be designed to realise the same function. An added complication comes from the fact that current-state nodes are changing all the time. Thus, although the basic principles of this model are simple, a fully connectionist implementation involves a complex control circuitry. On the other hand an implementation with networks of spiking neurons (Bugmann, 1997) may again be simpler, as discussed below.

ii) The incomplete coverage of the state space by state nodes and the use of on-going actions calls for some caution in the planning process using current spread: In a graph where places are at unequal distance, a resistive grid with equal resistances between all nodes will only minimise the number of intermediate nodes, not indicate the shortest path. A solution to that is to set the resistances in proportion to the distances between places. In this case, the path of least resistance will be indicated by the link carrying the highest input current (Althöfer, 1996).

If Leaky Integrate-and-Firer (LIF) Neurons were used in place of the artificial neurons (3), they would fire earlier with a large input current than with a small one. The activation of the goal node would lead to a spread of spiking activity through the net. Using only the wavefront, the first node that activates the current-state node would be the one that has to be reached by the next action. Using transition nodes with sequence detection capability (Bugmann, 1997) would automatically lead to the correct action to be executed (or integrated). Thus, with biological neurons, the circuit in figure 2 would be almost all that is needed. The only notable addition is probably a feedback inhibition scheme to control the persistence of activity in the grid and preserve only the wave front.

iii) An additional potential advantage of an implementation with spiking neurons is the capability for multicriteria planning. For instance, "finding a route between A and B which maximises the number of views of the sea" is a task that can be achieved by increasing the readiness of firing of all view-nodes associated with a view of the sea. This would increase the speed of the wave front along routes including these nodes. With a connectionist implementation of the resistive grid as used here, it is a very delicate matter to modify the potential distribution so that certain nodes are more likely to be part of the path.