

# From Rats to Robots: Engineering a Functional Biomimetic Navigation System

MICHAEL MILFORD

**Abstract:** The field of robotics is in a state of transition from traditional fixed robots located in industrial settings to mobile robots in domestic homes. Much of the new generation of mobile robots is produced using low cost hardware, making them unsuitable platforms for deploying the traditional probabilistic algorithmic approaches to mapping and navigation, which require accurate but expensive sensors such as scanning lasers. Animal navigation systems have been under investigation for many years as a potential source of inspiration for creating alternative robot navigation systems. This paper presents an overview of work pursuing the idea that animals, specifically rodents, make ideal models for creating practical robot navigation systems.

**Keywords:** Navigation, robots, rats, biomimetics, SLAM

## INTRODUCTION

Over the last decade robotics has been undergoing a transformation, with the traditional industrial robots used in manufacturing lines since the 1960s being supplanted by a new breed of mobile robots. These mobile robots are being used in increasing numbers in both domestic and industrial applications, and are now, in numerical terms, the most common form of robot (UNECE, 2005).

Perhaps one of the key reasons for this robot ‘revolution’ is the changing image of such robots; in the past regarded as novelty, luxury items, some are now regarded as practical labour-saving machines. Vacuum cleaner robots produced by companies such as *iRobot* and *Electrolux*, while not perfect, can in some households produce a net labour saving. By the end of 2004 there were an estimated 1.2 million personal domestic robots in service around the world according to the United Nations Economic Commission for Europe survey; of these one million were vacuum cleaner robots. Half of these one million were ‘installed’ during 2004, giving an indication of the rapid growth of domestic mobile robots. Lawn mowing robots are also entering into homes, although with a much slower uptake, with 46,000 units worldwide at the end of 2004 (UNECE, 2005). Predictions from the same source indicate worldwide sales of approximately seven million robots in the

period 2005 to 2008. This figure includes 4.5 million domestic robots with an estimated value of three billion US dollars, and 2.5 million entertainment and leisure robots with an estimated value of 4.4 billion US dollars. More speculative estimates predict as many as 39 million domestic robots being sold by 2010 (Horizons, 2004).

While such numerical growth is impressive, the sobering fact remains that the majority of these robots are essentially dumb machines. Most vacuum cleaner robots, for example, clean a room by following pre-programmed random paths and require, as admitted on one major manufacturer’s website, about four times as long to clean the room as a person. Most cannot navigate between rooms, requiring a human to reposition them. Many of the more capable systems require a user to pre-position beacons around the home that the robot can use to navigate. The ultimate vacuum cleaner robot, one that can ‘out of the box’ explore, learn, and then efficiently vacuum an entire house, has not yet transpired. One of the main reasons is certainly the difficulty of performing all these tasks, a problem which is described in the following section.

## THE NAVIGATION CHALLENGE

A robot that moves around its environment faces a number of significant navigation chal-

lenges. One of the most significant of these problems is known within the robotics literature as the Simultaneous Localisation And Mapping (SLAM) problem (Dissanayake, Newman, Clark, Durrant-Whyte and Csorba, 2001). When placed in an unknown environment, a robot must be able to explore its surroundings in order to learn the layout of the environment, creating a map of the environment. At the same time, however, the robot must also use that map to keep track of where it is in the environment, a process known as localisation. The two processes of mapping and localisation are interdependent – a map cannot be created without knowledge of the robot’s current location in the environment, and a robot’s location within the environment cannot be calculated without a map of the environment. The SLAM problem has been the focus of a very large amount of research in the robotics community over the past two decades.

There is a number of reasons why the SLAM problem is so difficult. A mobile robot can be thought of as a mobile sensor platform, with sensors that vary in type and capability. All of these sensors have limitations and produce readings that are noisy. The ubiquitous laser range finder, probably the most widely used sensor in robot navigation research, is generally regarded as highly accurate and reliable. This sensor measures distances to obstacles but is neither exact nor infallible. Measurements are only accurate within a margin of error. Measurements can also be grossly incorrect, such as when the laser ‘sees’ through glass wall, or when the light beam reflects off multiple surfaces before returning to the sensor. Bright sunlight in outdoor settings or even through windows in indoor settings can disrupt the sensor.

Other range sensors such as sonar (ultrasound) sensors have their own disadvantages. Sound waves from a sonar sensor project outwards in a cone-like manner from their point of origin. The ‘range’ reading for a sonar sensor is the range to the closest reflecting object within this cone. The wide beam results in significant uncertainty regarding the exact position of a detected obstacle (Moravec and Elfes, 1985). Surfaces with certain textures or surfaces that

do not transect the sonar cone at ninety degrees may not even reflect sufficient sound energy back to the sensor, meaning they are not detected. Like a laser sensor, multipath reflections can occur, producing misleading range to obstacle readings.

Vision-based sensors have their own set of limitations. Cameras are very poor at representing colours in images. A typical approach to dealing with this is to reduce the image domain to greyscale images, but this still leaves the problem of dealing with light intensity variation, which is an especially difficult problem in outdoor environments. There are techniques for dealing with illumination change but they are all partial, rather than complete, solutions to the problem (Buluswar and Draper, 2002, Tews, Robert, Roberts and Usher, 2005). Depending on the type of camera used, camera images can also be distorted, leading to further complications and inaccuracies when using them as a robotic sensor.

One of the most familiar concepts to researchers working with mobile robots is that of odometric drift. Wheeled robots are equipped with rotational encoders that measure the movement of each wheel. However, wheel slip means that the encoder’s measurement of the amount of wheel movement does not always correspond exactly to the movement of the robot along the ground. Wheel slip is unavoidable – theoretically it is impossible for the robot to even start moving without some amount of slip. Wheel slip means that even with a ‘perfectly’ calibrated system, over any significant amount of time slight measurement errors due to wheel slip will accumulate to unmanageable proportions. The problem is particularly significant in environments with varied ground surface textures, such as outdoor environments over rough or loose terrain.

Using more expensive, higher quality sensors and compensatory algorithms can reduce the impact of many of these sensing problems. A more expensive camera may have less lens distortion, and a higher quality laser will return more accurate and precise readings. Sensor fusion between multiple types of range sensors can help reduce the effects of an occasional

incorrect measurement from one particular sensor. Engineers can design the mechanical wheel drive system or tyres to minimize the amount of slippage. A Global Positioning System (GPS) can be used to remove the dependence on integration of wheel encoder values over time to update the estimate of the robot’s position.

However, many of these ‘solutions’ only work in certain environments, or add their own problems such as expense. The use of GPS is a good example; standard GPS does not work indoors and is not completely reliable in some built up outdoor environments (such as Manhattan, New York). In outdoor environments GPS can also ‘drop out’ long enough to severely test any navigating robot that is not equipped with a robust alternative navigation system. The issue of cost is also very relevant when considering low cost domestic robots, where expensive sensors or significant computational power is not possible.

What is inescapable is the fact that animals and humans manage to navigate in large and complex real world environments with a set of inexact, imperfect sensors. While engineers have developed sensors which are increasingly accurate (and often more expensive), it is obvious that in many cases nature has managed to solve the navigation problem from a completely different direction. How animals navigate so effectively poses a very interesting research question. In addition, from a commercial perspective it is desirable that robot systems be able to function with simple, inexpensive sensors.

The core SLAM problem is only one of many that must be solved by an autonomous mobile robot. When placed in a new environment, a robot must be able to effectively explore the environment, rather than staying in one section of the environment. If the robot is to perform any sort of useful navigation task, it must also be able to navigate to goal locations in the environment. Real world environments are not static, meaning that in the long term a robot must be able to adapt or at least cope with changes in the environment. Mobile robot research has been biased towards solving the SLAM problem, probably at least in part because it can be harder to investigate these

related problems if a robot cannot keep track of where it is. Consequently, much exploration and navigation research has been performed in simulation, with relatively little research on implementing these algorithms on real robots. In many cases, the focus on the SLAM problem has led to piecemeal attempts to implement goal recall and adaptation abilities, rather than consideration of the entire mapping and navigation problem from the start. While there are robots that successfully solve SLAM, or individual related problems such as exploration or goal navigation, there are very few robot systems that have been successfully demonstrated solving the complete navigation problem in real world environments.

## THE RATIONALE FOR A BIOMIMETIC APPROACH

Many animals and insects possess well documented formidable navigation abilities. By performing research into the navigation capabilities of animals, researchers are able to answer two important questions: what exactly can animals achieve using their navigation systems, and how do they do so. The answer to the first question is important because it demonstrates that many impressive feats of navigation can be achieved with a surprisingly simple set of sensors and relatively low computational resources. For instance, it is not necessary to have a high fidelity laser range finder to navigate effectively in a range of environments – most animals do not possess sensors that can tell them the range to obstacles varying in range from a few millimetres to ten or more metres, to a precision of ten millimetres (except bats). Experiments have demonstrated that rats can continue to navigate effectively even in pitch blackness, with the implication that some navigation systems can cope without a constant stream of feature rich information (Quirk, Muller and Kubie, 1990, Mizumori and Williams, 1993). The answer to the second question necessarily involves theories and models of the neural mechanisms in the animal brain used for performing navigation. It is these models that roboticists can draw upon to create their own artificial robotic navigation systems.

## Rodents

Rodents are one of the most studied animals in the field of biological mapping and navigation. Compared to other animals and even humans, their navigation capabilities are relatively well characterised. Furthermore, there has been much progress in understanding the *neural* basis of their navigation capabilities. The part of the brain thought to be responsible for much of a rodent’s navigation, the hippocampus, is one of the most studied brain regions of any mammal. Consequently, there are a number of well established models of how navigation is achieved neurally in rodents, although much is still uncertain.

One of the core concepts in navigation research is that of the cognitive map, which was introduced in early rat studies by Tolman (1948). This idea relates to observations of rats released in an environment with no reward food source. When placed back in the environment along with food sources at a later time, the rats were able to use information learned in the previously foodless environment to help them navigate to food sources, hence exhibiting a form of latent spatial learning. The cognitive map is often theorised to be a spatial reference that some animals create and use to navigate in their environments.

One of the first major neural discoveries with

respect to navigation in rodents was that of place cells in the rodent hippocampus, which fired preferentially when the animal was at a particular spatial location (O’Keefe and Dostrovsky, 1971, O’Keefe and Conway, 1978). Over the next three decades, several other types of spatial neuron were discovered, including neurons that responded to the animal’s head orientation, called head-direction cells (Ranck, 1984). Most recently, cells which fired at regular grid-like intervals in the environment were discovered and dubbed grid cells (Hafting, Fyhn, Molden, Moser and Moser, 2005, Fyhn, Molden, Witter, Moser and Moser, 2004). However, since grid cells were only discovered recently, most of the models of rodent hippocampus deployed on robots to date have focussed on head-direction and place cells.

## Bees

Because insects (and correspondingly insect brains) are generally much smaller than mammals, research on their navigation systems is biased towards observational work rather than towards recording cells in their tiny brains, which is quite difficult to do. Consequently most of the theories and models of insect navigation have been developed from careful observation of their behaviour in a wide variety of experiments.

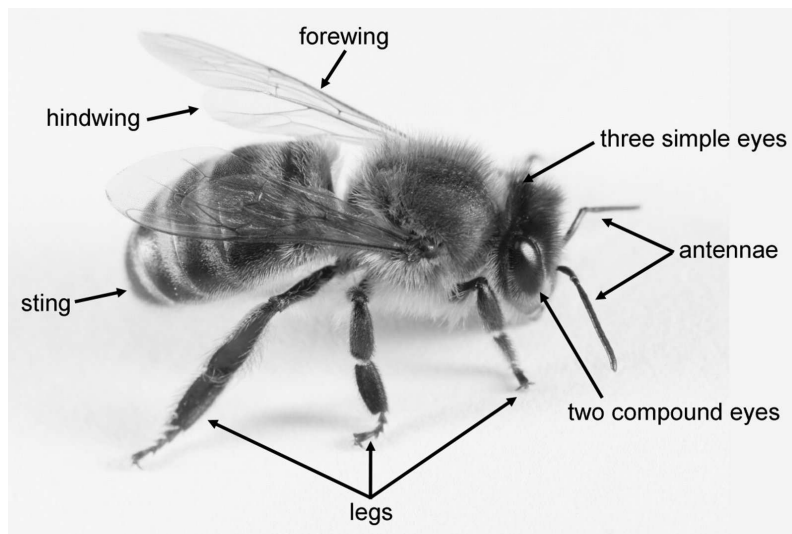


Figure 1. Anatomy of a worker bee.

Honey bee workers are equipped with a range of different sensors to help them fly many hundreds of kilometres during their lifetime of foraging for nectar and pollen. They possess two types of visual sensors, two large compound eyes and three much smaller ‘simple’ eyes arranged in the shape of a triangle on the top of the bee’s head. The compound eyes detect both light and colour and have a large field of view. The simple eyes are sensitive to light but can’t resolve images. Non-visual sensors include the antennae on the bee’s head which perform both odour detection and flight speed measurement. These features and others are shown in Figure 1.

Because honeybees start and finish at the same location (the hive) during their foraging trips, they are known as central-place foragers. Typical foraging trips range up to two to three kilometres from the hive. However, bees have been observed foraging up to 13.5 kilometres from their colony location, a huge distance when considering the tiny size of the bee (Ratnieks, 2000, Frisch, 1967). Most models of bee navigation have been developed by repeatedly subjecting bees to experiments designed to isolate one particular aspect of navigation behaviour, such as the bee judging flight distances by its apparent visual speed through the environment. Reverse engineering the bee’s navigation system in this way is not a perfect science, and seemingly identical experiments can produce very different results and interpretations. For example, one of the dominant debates in this field is whether bees possess some form of cognitive map.

### Ants

Desert ants regularly leave their nest and wander significant distances into the desert around them, following squiggly paths, only to then return via a more or less straight line directly back to their nest. In order to maintain an idea of where they are relative to the nest, the ants perform dead reckoning or path integration, updating their estimate of where they are based on their movement speed and direction. Experiments by Muller and Wehner (1988) using the species *Cataglyphis fortis* suggest that an ant keeps track of the average of all directions in which the ant moves, weighted by the distance moved in each direction. The reference vector

used for calculating the new movement angle is the previous homeward course direction. This navigation system leads to only small errors in most situations, except when an ant performs sudden backward turns, which the experiments show happens only rarely. Since the angular errors are systematic, an equally large clockwise and anti-clockwise turn results in the two generated errors cancelling out. The experiments also reveal that the ants do not usually have a direction bias during their foraging activities.

Ants also demonstrate robust coping strategies for non-ideal situations when their navigation strategy does not lead them back to their exact nest location (for instance, a gust of wind may have blown them off course). One group of researchers (Wehner, Gallizzi, Frei and Vesely, 2002) describe the navigation strategy that desert ants have adopted to deal with errors in their path integration process. When returning to the nest from a foraging location, the ants always use their *home vector* to return to the supposed location of the nest. The home vector is calculated during the outbound trip using weighted angular averaging as described previously. For long foraging trips out from the nest, there is usually some dead reckoning error, meaning the ant initially returns to a location near, but not at the nest. To find the nest, the ant starts this systematic search, initially searching a small area but gradually increasing this search area until it finds the nest. In this way, ants are able to combine two navigation strategies to produce highly reliable overall navigation behaviour in their desert environments.

### DISCUSSION

When choosing a biological navigation system upon which to base a robotic navigation system, two main issues must be considered. The first is how well the animal’s navigation capabilities are known – for some animals such as rats, bees and ants, their navigation capabilities have been widely studied and there is at least some consensus on what they can and cannot do. This in turn can provide a roboticist with an idea of what they can expect their robot to be capable of if they get the modelling correct, and what they may need to improve artificially.



The second issue regards how well the neural mechanisms underlying an animal’s navigation abilities are known. It is here that rodents stand out, as their brains, and specifically the hippocampus, have been recorded from extensively over the last thirty years. In contrast, little is known about the neural mechanisms behind navigation in insects. Because of ethical considerations, relatively little is known about navigation mechanisms in the brains of primates or humans. Consequently, the body of work described in this paper focussed on using models of rodent navigation to produce a functional robotic navigation system.

**RatSLAM**

There is a variety of theories on how rodents map and navigate in their environments,

and consequently a range of models of these processes. However there is some degree of consistency across most models regarding the major inputs, components, and functions. The model described in this section follows the general consensus that rodents use both external and internal sensory input, and that they have neural structures for representing their spatial orientation and location. The model and surrounding systems are called RatSLAM, with ‘Rat’ referring to the biological animal serving as inspiration, and ‘SLAM’ referring to the robotics term for Simultaneous Localisation And Mapping. The initial development of RatSLAM drew heavily upon the preceding rodent modelling work on robots (Arleo and Gerstner, 2000, Browning, 2001, Gaussier, Revel, Banquet and Babeau, 2002). Figure 2 shows the overall model structure.

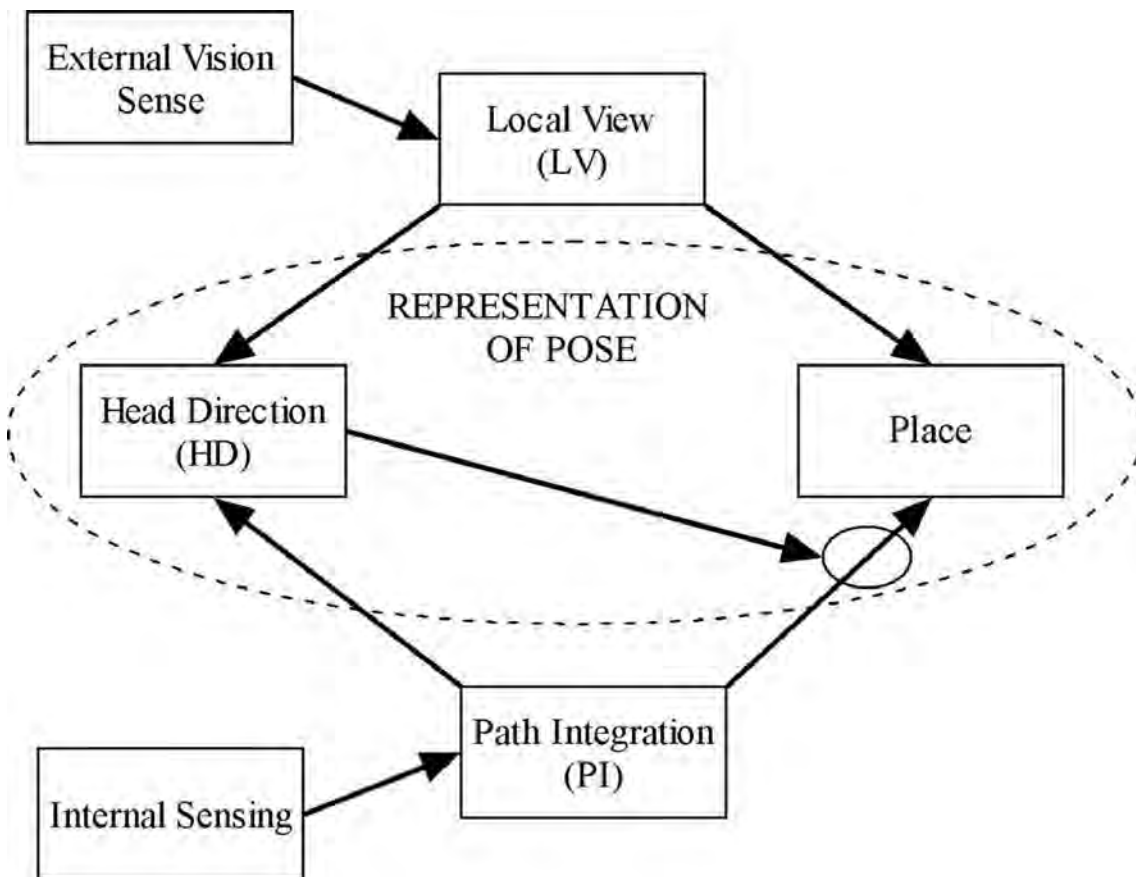


Figure 2. Overall structure of the hippocampal model.

The robot’s pose is encoded in two modules known as the head-direction (HD) and place networks. Competitive attractor networks are used to represent the cell populations, which are a type of neural network with highly interconnected excitatory and inhibitory links. The key characteristic of an attractor network is its ability to maintain stable clusters of active cells, also referred to as ‘attractors’, ‘activity packets’ or ‘bumps’. The head-direction network encodes the robot’s orientation, the place network the robot’s location. Ideothetic information in the form of wheel encoder input drives the path integration process, which updates the robot’s state information in both the head-direction and place networks. Allothetic information in the form of camera images is processed to form a representation known as the local view (LV), which calibrates the robot’s state information stored in the head-direction and place networks.

### Representing Orientation

The core component of the spatial orientation model is an array of neural units or ‘cells’ roughly corresponding to biological head-direction cells. Each cell is tuned to be maximally active when the robot’s orientation matches the cell’s preferred direction. The cell’s activity level reduces as the robot orientation rotates away from this preferred direction. The cell arrangement reflects their associated robot orientations – nearby cells encode similar robot orientations. When the ensemble activity of the head-direction cells is viewed as a bar graph, one can see a ‘packet of activity’ that resembles a Gaussian curve. The centre of this ‘activity packet’ represents the current perceived orientation of the robot.

Without recalibration the robot’s perception of its orientation will gradually become incorrect. Recalibration is a two step process; the robot must first learn associations between its orientation and the input from its external sensors, and then use that information to correct its orientation state information when it later receives that same sensory input. Once the robot has learned the associations between

sensory input and orientation, it can use them to correct its perceived orientation. Whenever a familiar visual scene is encountered, activity is injected into the head-direction cells associated with the scene. If the robot’s perceived orientation is correct, this has the effect of reinforcing its belief. However, if the current perceived orientation is incorrect, the injected activity initiates an alternative hypothesis regarding the robot’s orientation, which then competes with the current hypothesis under the competitive attractor network dynamics.

### Representing Location

The representation of the robot’s location is encoded using a network of cells roughly corresponding to biological place cells. The place cells are modelled as a two-dimensional matrix of cells, with each cell tuned to be maximally activated when the robot is at a specific location. A coarse representation is used, with the path integration system tuned so that each place cell responds within a physical area of approximately 250 mm by 250 mm.

### Representational Limitations of Place and Head-direction Cells

A series of mapping experiments were run using a Pioneer 2 robot in a small environment with artificial landmarks (Figure 3). The robot was able to use the model of place and head-direction cells to map and navigate the environment over short time periods. However, in longer experiments the network’s tracking ability proved to be unstable. Over the period of an hour the robot became lost and its perceived location moved well outside its two by two metre arena. The place cell model was able to keep the robot correctly localised for short test durations only. Because the robot was moving based on its self-estimate of position, not its actual position, a small error in pose usually compounded quite rapidly and the tests had to be terminated before the robot collided with objects outside the arena.

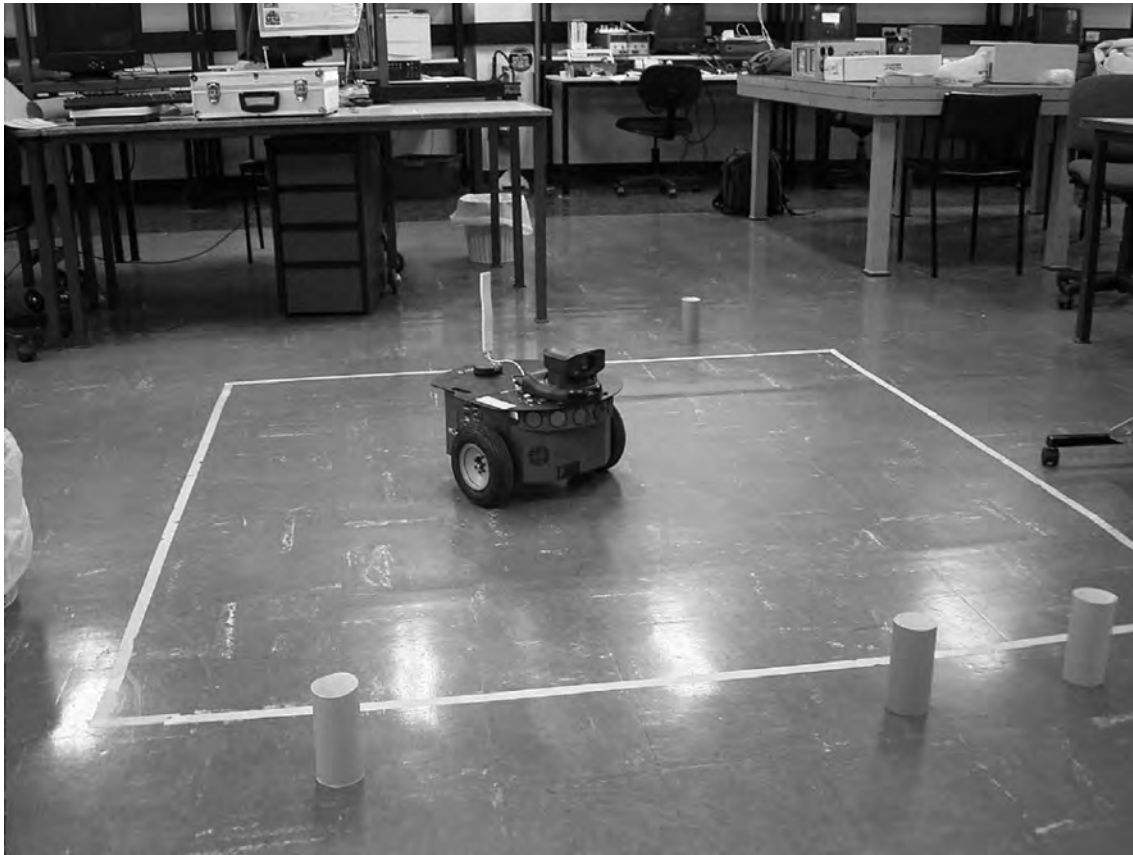


Figure 3. The first test environment was a two by two metre arena with coloured cylinders as visual cues just outside the boundary of the arena.

The reason for the system failure was the splitting of the robot’s pose (location and orientation) representation. Active place cells represented the robot’s location in the environment but not its orientation. Active head-direction cells represented the robot’s orientation but not location. This splitting of the spatial representation broke down when the robot was put in a perceptually confusing situation.

This problem is illustrated in Figure 4. It shows a schematic of the robot at two different moments in time when it has associated its current estimate of its location and orientation (as encoded by active place and head-direction cells) with seeing a certain two cylinders in the environment. In this hypothetical environment, there are actually two sets of these cylinders at different locations. Figure 4b shows the robot some time later when it encounters the same visual scene again. The familiar visual scene

activates place cells and head-direction cells representing the two locations and orientations the robot has seen the cylinders from. In a standard probabilistic robot mapping system, the robot would now have two distinct pose hypotheses, that is, two estimates of its pose. Instead, it has two estimates of its location, and two estimates of its orientation, but no binding information about which location estimate goes with which orientation estimate.

This lack of binding between location and orientation leads to problems when the robot continues moving. Ideally, as the robot then continued to move forward, the location estimate in the left of Figure 4b would move downwards and slightly to the left. However, it also has a second orientation estimate associated with it, shown by the dotted arrow. The system as it stands has no way of ‘knowing’ which orientation estimate is correct for each



location estimate. Without appropriate binding, the separate representations of place and orientation cannot sensibly maintain multiple hypotheses of pose. The neurophysiological separation of the place and head-direction cells (O’Keefe and Conway, 1978, Ranck, 1984) renders this a spatial memory form of the binding problem (Thiele and Stoner, 2003). Methods such as updating the location estimates based on the average orientation estimate (shown by the short thick solid arrow) unsurprisingly rapidly lead to the robot becoming lost. Other possible techniques such as splitting each location estimate into two, one for each orientation estimate, can be implemented using

standard probabilistic mapping algorithms, but are impossible to implement using a standard attractor network. While mechanisms such as synchronisation have been proposed as a way of binding the information of different neurons in other domains (Gray, Konig, Engel and Singer, 1989), we observed that there was a simple modification of the model that offered an elegant engineering solution to the spatial binding problem (Milford, Wyeth and Prasser, 2004). The following section describes the pose cells, developed to overcome the representational limitations of place and head-direction cells.

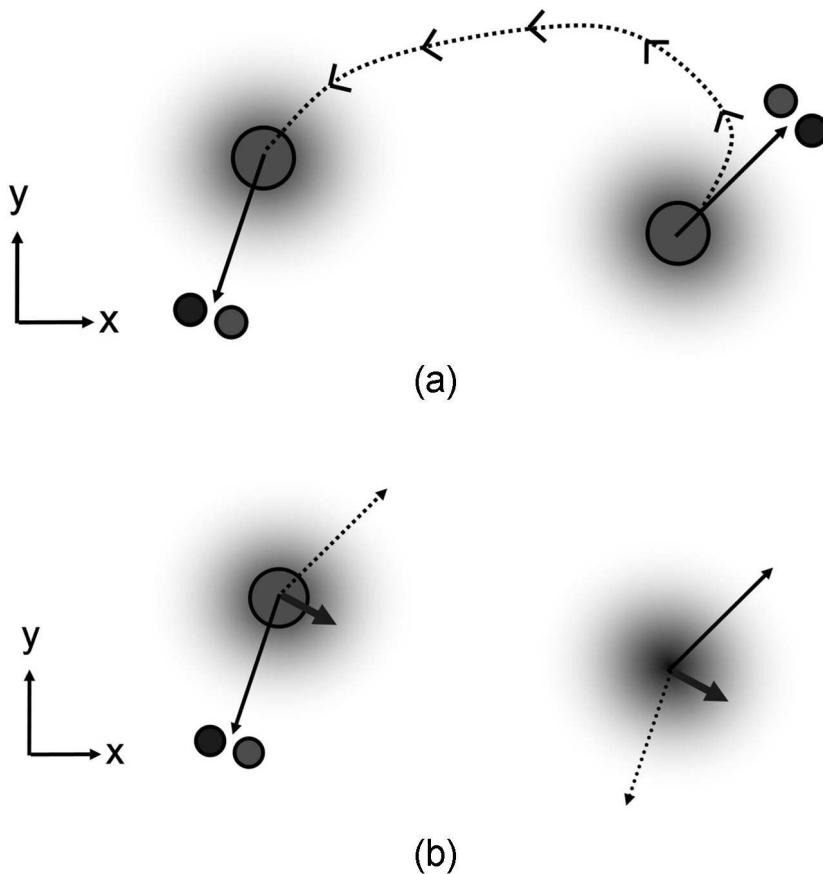


Figure 4. (a) An overhead view of the robot (large circle) at two moments in time where it has associated a cluster of active place cells (grey shading represents location encoded by cells) and head-direction cells (long solid arrow represents direction encoded by cells) of the robot with an identical view of two cylinders. (b) At a later time the robot encounters the same two cylinders as in (a) and activates place and head-direction cells supporting the two possible robot locations and two possible robot orientations. The representational problem is revealed when the robot continues to move and tries to update its location estimates. Each location estimate has two orientations associated with it, one correct one shown by the long solid arrow, and one ‘incorrect’ one shown by the dotted arrow (Milford, 2003).

### Pose Cells

The pose cells in RatSLAM are a type of cell that combines the characteristics of place and head-direction cells, in that an individual cell represents a particular robot location *and* orientation. They are formed from the same type of competitive attractor networks used to form the place and head-direction cells, but with the cells arranged in a three-dimensional structure. Each axis of the structure corresponds to a different state variable,  $x'$ ,  $y'$  and  $\theta'$  (Figure 5). For this cell structure primed co-ordinate variables were used to represent the degeneration in correspondence between co-ordinates in the cell structure and co-ordinates in physical space. For example, in indoor experiments, each pose cell initially fires maximally when the robot is in a  $0.25 \times 0.25$  m area and orientated within a 10 degree band. However as an experiment progresses, the pose volume each pose cell corresponds to can grow, shrink, warp, or even disappear under the influence of visual information. Like the previous networks, cells are highly interconnected by both excitatory and inhibitory connections, and connectivity wraps around in all three-dimensions to the opposing faces of the structure. The stable

state of the network in the absence of any input is a cluster of active cells or ‘activity packet’, representing a single estimate of the robot’s pose in the environment.

Each cell has an activation range between 0 and 1, with the activity level qualitatively encoding the probability of the robot having the specific pose (location and orientation) associated with that cell. At any one time many cells can have non-zero activations, meaning that the activation level of any individual cell is relatively meaningless. However, by viewing the activity of the network as a whole, it is possible to extract the network’s estimate of the robot’s most likely pose. Figure 6 shows a snapshot of cell activity in the pose cell matrix during an experiment. The largest and most strongly activated cluster of cells represents the robot’s most likely pose. However, other clusters of active cells represent alternative possible robot locations and orientations. These alternative possibilities are due to perceptual ambiguities in the robot’s environment. As the robot continues to move, one of these other clusters of active cells can become dominant if the robot’s perceptual input supports the location and orientation associated with those cells.

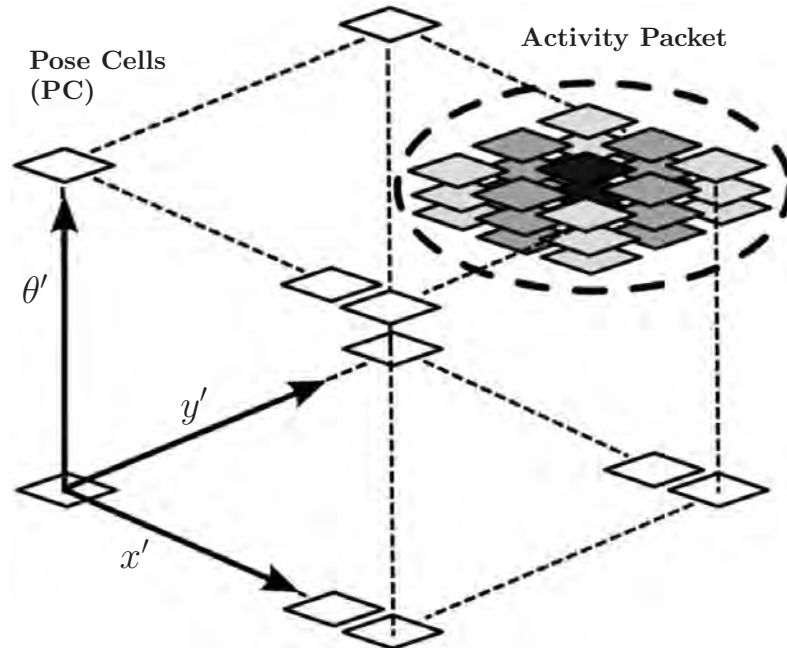


Figure 5. The three-dimensional pose cell model. Each dimension corresponds to one of the three state variables of a ground-based robot (Milford, 2003).

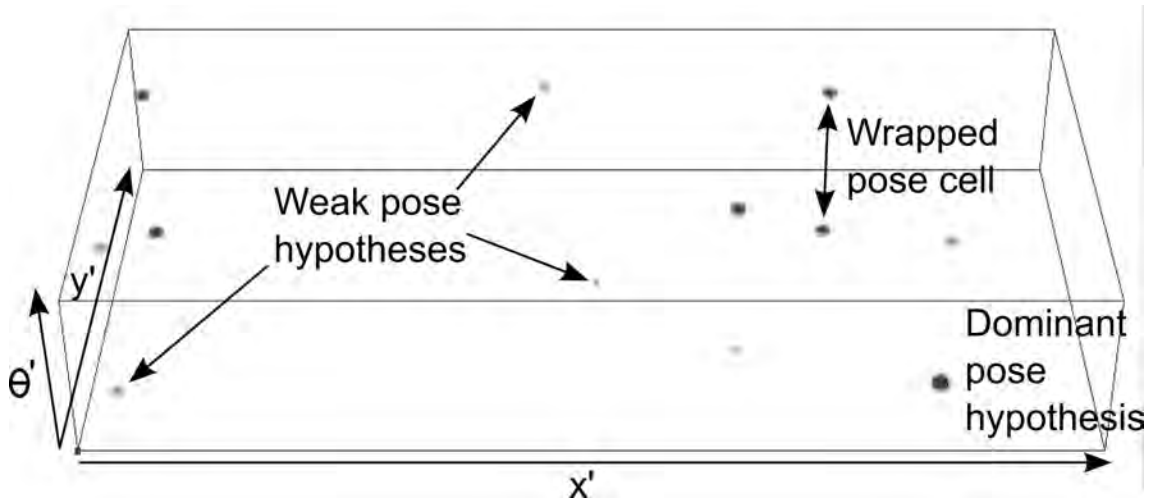


Figure 6. Snapshot of pose cell activity during an experiment. Several activity packets of varying size are evolving under the influence of the competitive attractor dynamics (Milford, 2004).

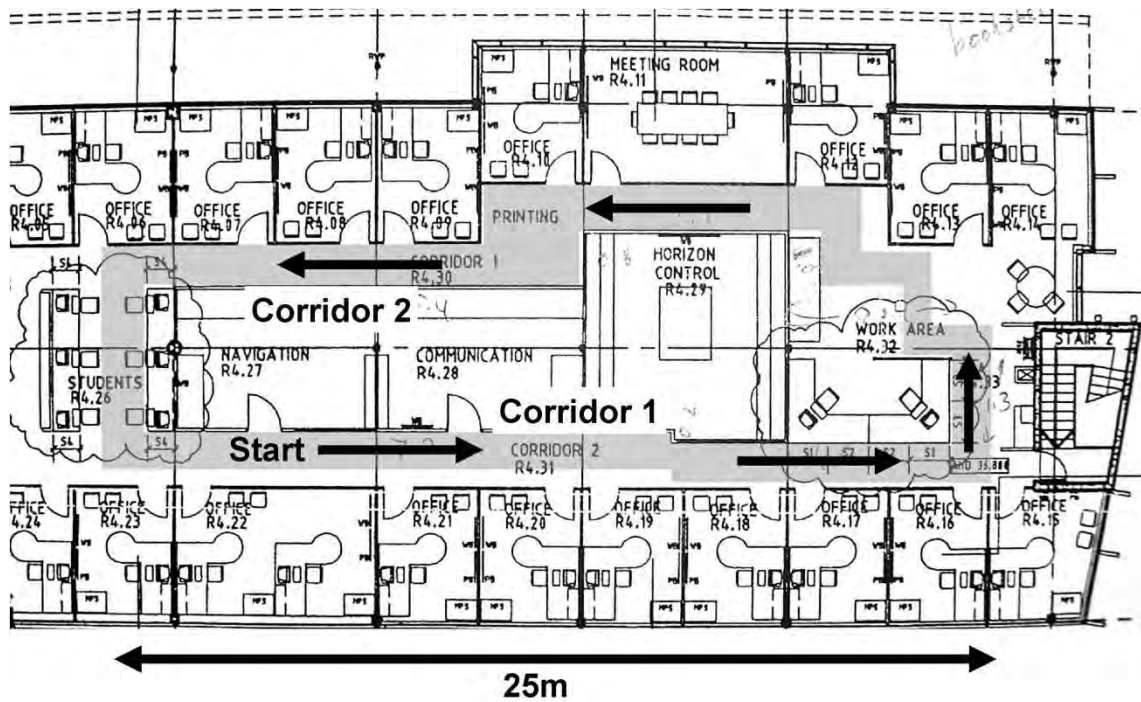


Figure 7. Floor plan of loop environment. The shaded area and arrows indicate the path of the robot. Floorplan kindly provided by the CSIRO.

### Experimental Setup

One of the test environments for the pose cells was a 70 metre long corridor loop at the Queensland Centre for Advanced Technologies (Figure 7). Loop environments are typical testing environments in robotic navigation research,

because the robot navigation system must be able to recognise that it is back where it started after completing a loop of the environment. The robot's control scheme used the sonar array to perform autonomous wall following. SLAM was performed live during the actual test.

## Closing the Loop

Figure 8a shows the trajectory of the robot calculated using only path integration. Small errors in the sensory input from the wheel encoders as well as wheel slip have accumulated over time and produced the characteristic drift in the robot’s estimate of its own position. When the RatSLAM system was used the trajectory of the robot, as represented by the path of the dominant cluster of active cells through the pose cells, was consistent over repeated laps, as shown in Figure 8b. The figure shows the path of the most highly active pose cells over time, that encode the robot’s representation of its location and orientation in the environment. At the start of each new lap the robot was able to repeatedly close the loop, with dashed lines showing the robot correcting its estimate of its location. At other stages of the experiment the system was able to recover from path integration errors.

## Navigation: Goal Memory

With the model able to create a stable representation of its environment, the next challenge was to use the map to perform navigation tasks. The task of adding a goal recall capability to the RatSLAM system was undertaken in a pragmatic way as with the development of the core mapping method. Some biological navigation concepts were considered, such as the idea of activity encoding the desired future location of the robot. However there was little attempt to model the actual firing properties of biological place cells during task behaviour, with more emphasis given to the computability of the algorithms.

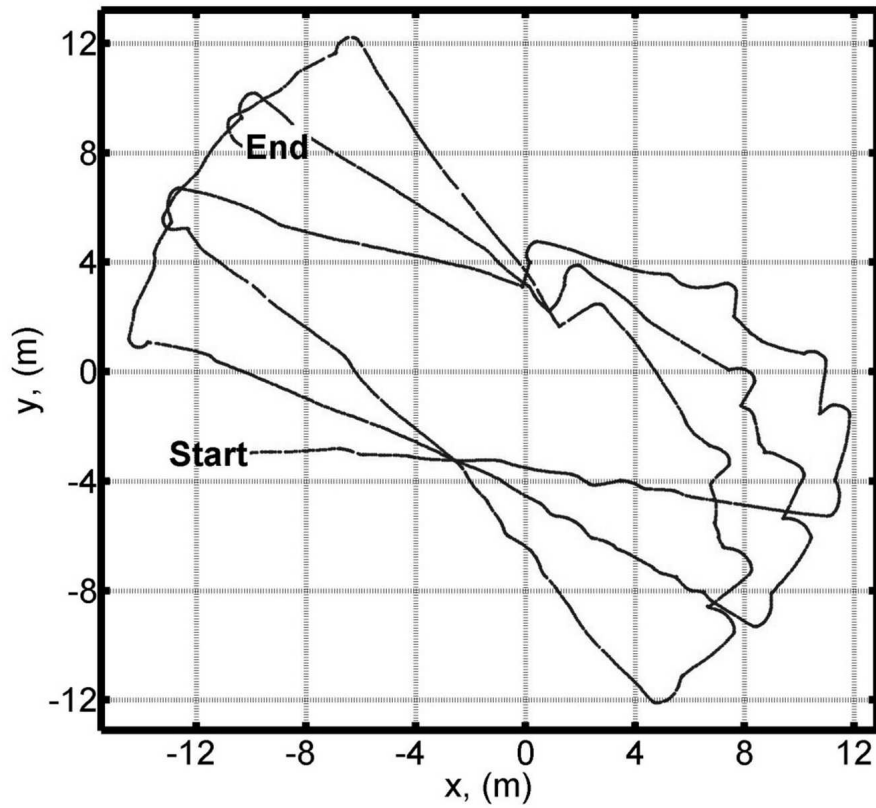
The navigation system that was developed is known as the goal memory component of RatSLAM. The goal memory system uses a copy of the pose cell structure, but adds temporal link information between cells as the robot explores the environment. In effect, the goal memory system learns the temporal gradient between places in the environment. To calculate a path from the robot’s current location to a goal location, the system integrates the temporal gradient, starting at the goal location. The integral builds quickly along long or difficult

paths, and slowly on more direct paths that the robot can navigate effectively. The result is a temporal map, essentially coding how long it takes to get from any location to the goal location. To navigate to the goal, the robot picks the direction of movement within the temporal map which moves it to a location closer in time to the goal. Because the integral operation starts at the goal location, the robot is guaranteed to only ever have one possible (and optimal) direction of movement.

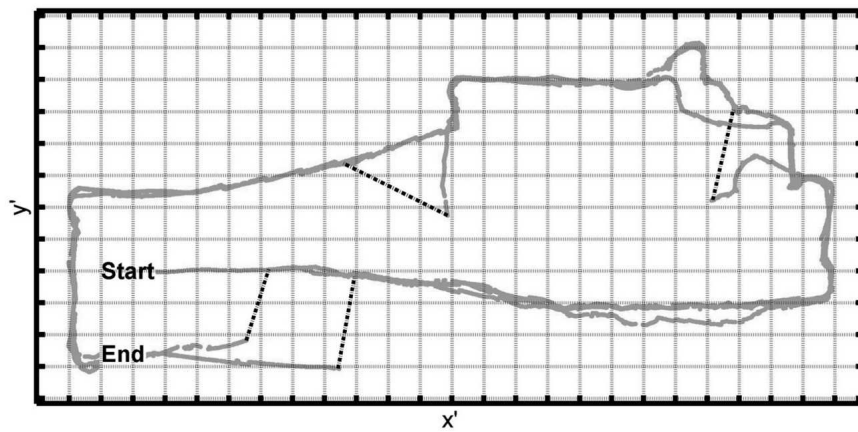
The goal memory system was tested initially in part of an indoor office environment, shown in Figure 9a. Figure 9b shows the temporal map for a typical goal navigation task from A to 1 on the map. Darker areas correspond to lower goal memory cell activity levels and hence locations close in time to the goal. The robot was able to follow the temporal gradient to reach the goal, as shown in Figure 9c.

## Increasing Environment Size: Navigation Problems

The initial goal memory experiments, while successful, were conducted in a small environment. One consequence of this was that the layout of the representation in the pose cells and goal memory cells (Figure 9b) was quite similar to the actual environment layout (Figure 9a). A second set of experiments was conducted using most of the building floor area shown in Figure 10a. This larger, more complex environment led to longer experiment durations, since the robot had a lot more ground to cover and more route possibilities. Wrapping in the pose cell structure became a consideration, since the nominal area encoded by the pose cells was smaller than the physical area of the environment. While the number of pose cells could be increased to avoid this problem, rodents are known to navigate large complex environments with a limited number of spatial cells. To test the overall system functionality with pose cell wrapping, a small  $40 \times 20 \times 36$  pose cell matrix was used with a nominal no-wrap area representation of  $10 \times 5$  m (each cell is nominally  $0.25$  m x  $0.25$  m). This is much smaller than the actual  $28 \times 13$  m environment area.



(a)



(b)

Figure 8. (a) Trajectory of the robot calculated using only path integration. (b) Robot trajectory calculated by the RatSLAM system. RatSLAM closed the loop upon starting the second and subsequent laps, and also recovered from path integration errors at other points. Dashed lines indicate the start and finish of re-localising corrections. Each grid square represents  $4 \times 4$  pose cells in the  $(x', y')$  plane of the model shown in Figure 5 (Milford, 2004).



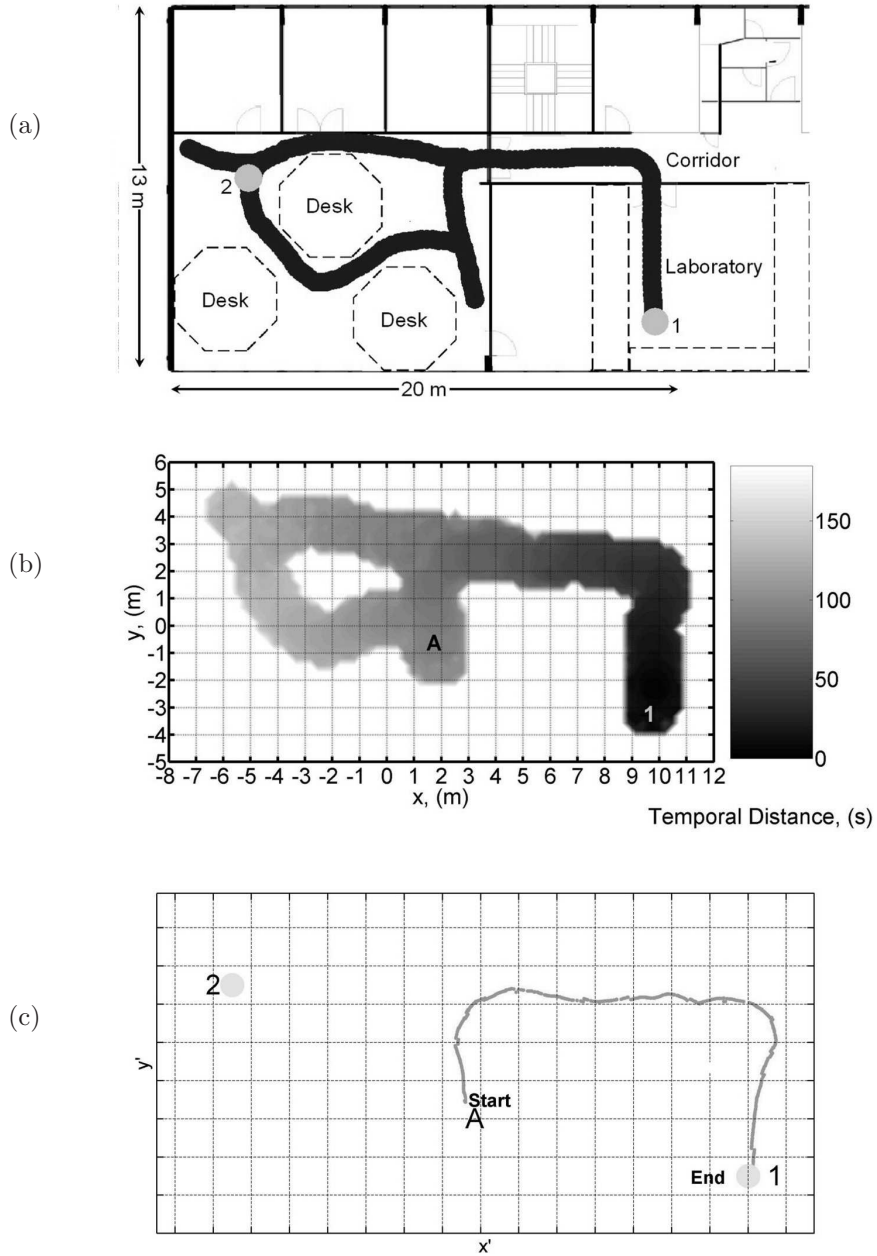
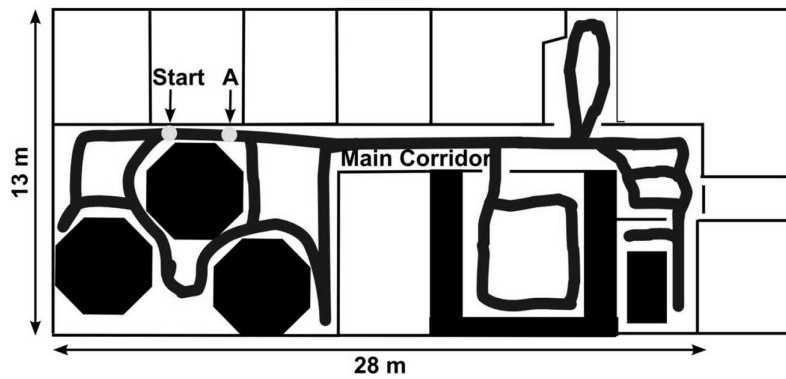
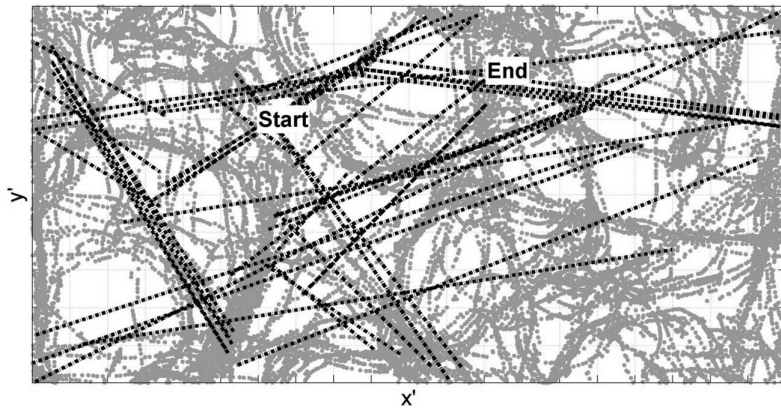


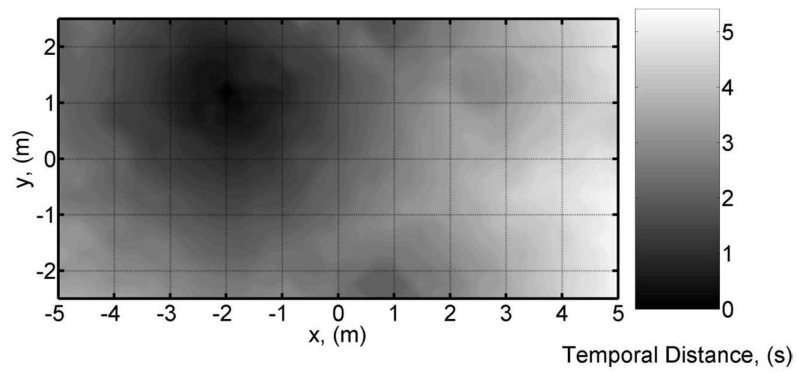
Figure 9. (a) Floor plan and robot trajectory for initial goal navigation experiments. The numerical labels indicate the two goal locations. (b) The temporal map cells after recall of the first goal. Darker areas correspond to lower cell activity levels and hence locations close in time to the goal. (c) The path the robot followed to reach the first goal. Each grid square represents  $4 \times 4$  pose cells in the  $(x', y')$  plane (Milford, 2005). ©2005 IEEE.



(a)



(b)



(c)

Figure 10. (a) Floor plan of large indoor environment. The robot’s path is shown by the thick line. (b) Dominant packet path for a  $40 \times 20 \times 36$  pose cell matrix. The path is projected onto the  $(x', y')$  plane. Each grid square represents  $4 \times 4$  pose cells in the  $(x', y')$  plane. ‘Start’ and ‘End’ mark the initial and final location of the dominant activity packet. (c) Temporal map for the large indoor environment (Milford, 2006). ©2006 IEEE.

Figure 10b shows the trajectory of the dominant activity packet through the  $(x', y')$  plane of the pose cells. As can be seen in the figure, the pose cell trajectory looks very different to the robot’s actual trajectory through the environment. This is in part due to the wrapping connectivity of the pose cell structure – many times the dominant activity packet would wrap off one ‘edge’ of the pose cell matrix and re-appear on the opposite edge. The size of the environment is also a factor – the robot travelled longer distances before returning to a familiar place, and consequently errors due to odometric drift were larger. The correction of these errors is shown by the dashed lines in Figure 10b – these represent times when the dominant activity packet in the pose cells switched to a different cluster of pose cells. These correction events were driven by the robot’s vision system supporting cells encoding a different robot location to that supported by the then dominant activity packet. With enough visual support, the activation levels of these other cells increased until they became the dominant cluster of active cells in the network.

Figure 10c shows the temporal map created for this environment by the goal memory system, for a goal located at the ‘Start’ location in Figure 10a. The temporal map is created using the goal memory cells, which themselves are a copy of the pose cells but with added temporal information. Consequently, the goal memory cells inherit the discontinuities of the pose cell representation. In addition, because the environment was larger than could be represented within ‘one pass’ of the pose cells, some pose cells represent multiple places in the environment due to the network’s wrapping connectivity. All this means that the layout of the cells within the pose cell structure can no longer be interpreted as a strict representation of the spatial layout of the physical environment. Consequently, the goal memory system, which is dependent on the layout of the goal memory cells, produces a nonsensical temporal map. Not surprisingly, robot navigation failed completely in this experiment, with the robot repeatedly trying to drive through a wall, with its local obstacle avoidance routine overriding it

whenever it got too close.

Analysis of the goal memory system revealed that it had failed because it implicitly interprets the structural arrangement of the pose (and goal memory) cells as corresponding to the physical arrangement of the places associated with them. These experiments showed that in large environments the pose cell structure loses spatial relevance, through a number of phenomena related to wrapping of the pose cells and visual re-localisation. Instead the pose cells develop a characteristic shared by biological place cells in the rodent hippocampus – the arrangement of the cells bears no correspondence to the spatial structure of the environment. This characteristic means it is not possible to directly extract spatial information from the pose cell structure for tasks such as navigating to goals. To address this limitation, the experience mapping algorithm was developed, in order to maintain the inter-cell spatial information of the pose cells while retaining the topological correctness of the pose cell maps.

### Experience Mapping

The experience mapping algorithm uses the output from the pose cells and local view cells to create an experience map. An experience map is a graph-like map containing nodes, known as experiences, and transitions or links between these experiences. Each experience represents a snapshot of the activity within the pose cells and local view cells at a certain time. In effect an experience is the robot’s final representation of a distinct place in the environment, along with information about what that place looks like and other behavioural and temporal information. When the set of existing experiences is insufficient for describing the pose and local view cells’ activity state, a new experience is created. Figure 11 shows an experience and how it is associated with certain pose and local view cells;  $x', y'$  and  $\theta'$  describe the location of the cells within the pose cell matrix associated with the experience, and  $V$  describes the local view cell associated with the experience. Each experience also has its own  $(x, y, \theta')$  state, which describes its location within the co-ordinate space of the experience map. This co-ordinate

space is completely separated from the pose and local view cell co-ordinate spaces, and through a process of map correction becomes much more representative of the spatial layout of the environment.

While experiences represent distinct places in the environment, experience transitions store information about the physical movement of the robot between one experience and another, as well as information about the robot’s behaviour during the transition and the time taken to complete the transition. The spatial information stored in the transition links is crucial to the process of map correction, which keeps the experience map layout spatially representative of the environment.

When the robot returns to a familiar part of the environment after spending time in an unexplored part of the environment, its estimate of where it is, as encoded in the pose cells, may be incorrect, due to the build up of small errors over time (odometric drift). In such a situation, familiar visual scenes will activate pose cells representing the correct location of the robot. This in turn causes the robot’s associated location within the experience map (given by the maximally active experience) to jump from the

new experience it has most recently learned to a previously learnt experience. This will cause a momentary discontinuity in the spatial layout of the experience map. The transition between the two experiences will encode only a relatively short distance of robot travel, but the actual positions of the two experiences in the  $(x,y)$  space of the experience map will be relatively far apart. The map correction process creates a spatially consistent map by shifting experiences so that the difference between the distance separating any pair of linked experiences in the experience map space, and the distance encoded by the transition link itself, is minimized.

To perform goal navigation, the experience maps are converted into temporal maps in a process similar to that used to convert the goal memory cells into temporal maps. The experience maps are also used by exploration algorithms that enable a robot to rapidly explore a novel environment, and adaptation mechanisms which enable a robot to modify its map to reflect changes in the environment – due to space restrictions, these sections of work are merely mentioned here, with further information available elsewhere.

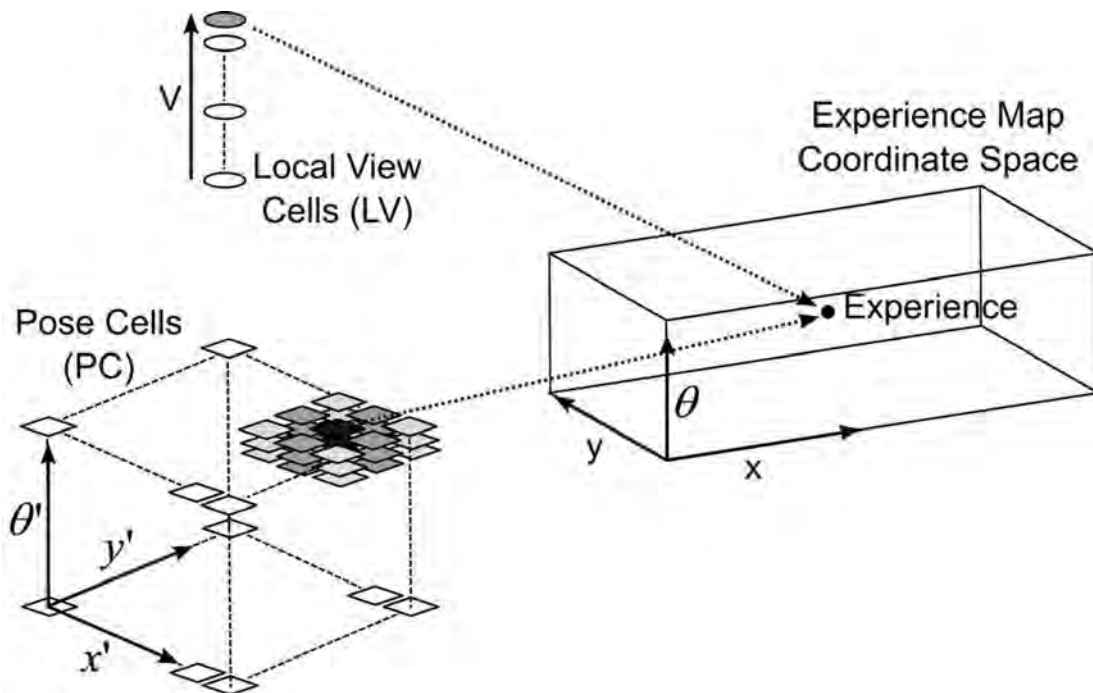


Figure 11. Experience map co-ordinate space. An experience is associated with certain pose and local view cells, but exists within the experience map’s own  $(x, y, \theta')$  co-ordinate space (Milford, 2006).

## Experience Maps and Navigation

The new navigation system was tested in the environment shown in Figure 10a. Figure 12 shows the pose cell representation of the environment, and the resultant experience map. The experience mapping algorithm is able to successfully remove the discontinuities in the pose cell map, as well as draw multiple representations of the same place into overlapping areas of the map. This is due to the ability of the experience mapping algorithm to correct its layout based on connectivity information, unlike the pose cell representations which are rigidly linked to the static pose cell matrix structure (Milford, Wyeth and Prasser, 2006).

Figure 13 shows the planned and executed routes for six typical goal navigation tasks in the environment. The temporal maps shown in Figures 13 a,c,e,g,i,k, which were now built using the experience map, clearly represent the actual layout of the physical environment. The robot followed the planned route to the goal for goals one and two. The robot followed routes that were quite similar to the planned routes for goals four and five, but with some minor variations. For the routes to goal three and six the robot deviated very significantly from the initial planned routes to the goal, but did eventually reach and stop at the goal.

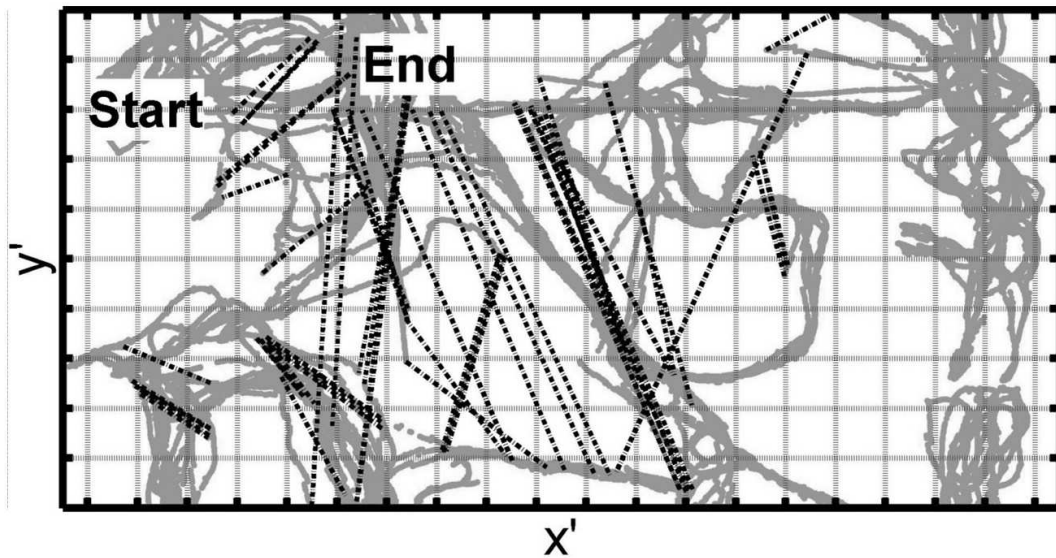
From observing these and other experiments it was clear that most of the mistakes made by the route following algorithm resulted from using the incorrect local movement behaviour at critical intersections. Once the robot took a wrong turn it was also clear that the route planning algorithm immediately started to plan new ‘shortest’ routes to the goal. After mistakenly turning into the room containing goal two while navigating to goal six, the robot attempted to exit back into the corridor, as shown by the loop at the entrance to the room. It was not able to time its switching of local movement behaviours correctly and ended up doing a complete loop of the room.

Although developed as a heuristic solution to a navigation problem, the experience map-

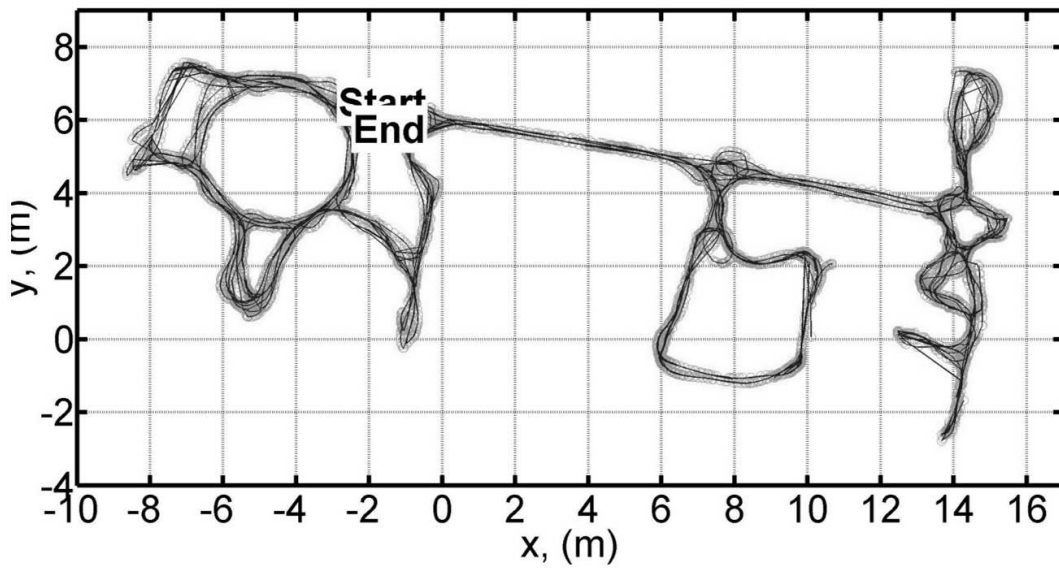
ping algorithm bears a functional resemblance to the place cells originally modelled at the start of the project. Specifically, place cells are currently thought to encode contextual spatial memories, rather than spatial memories in isolation (Hafting, Fyhn, Molden, Moser and Moser, 2005). Each experience in the experience map represents a distinct spatial location in the environment through its association with the pose cells, but also stores the visual appearance of that place through its association with the local view cells. Furthermore, transitions between experiences encode behavioural and temporal information about the robot’s movement through the environment, providing some parallels to the contextual spatial memories thought to be stored in place cells.

By the end of the project, the RatSLAM system had been developed to the stage where a robot could use it to explore an unknown environment, create a map of that environment, use that map to navigate between places, and adapt to simple changes in the environment (Milford, 2008). In the course of the work several modifications of the original models were made to overcome functional navigation limitations. One major problem was the splitting of the representation of the robot’s state into location and orientation in the place and head-direction cells, which was solved by developing *pose cells* combining both their characteristics. Another major problem was encountered when attempting to perform purposeful navigation using the spatial representations built up in the pose cells. While the cells learned stable representations, they lacked easily accessible spatial information required to plan and execute routes to goals. The experience mapping algorithm was developed to take the output from the pose cells and turn it into a spatially representative map that could be used for goal navigation and other tasks. In recent work, we have demonstrated that the mapping system can scale to mapping entire suburbs (Milford and Wyeth, 2008), and current work is investigating the scalability in time in experiments running over several days or weeks.





(a)



(b)

Figure 12. (a) Path of the most highly active pose cells through the  $100 \times 40 \times 36$  pose cell matrix. The path is projected onto the  $(x', y')$  plane. Each grid square represents  $4 \times 4$  pose cells in the  $(x', y')$  plane. (b) Resultant experience map, with all discontinuities removed and multiple representations drawn into overlapping areas of the map (Milford, 2006). ©2006 IEEE.

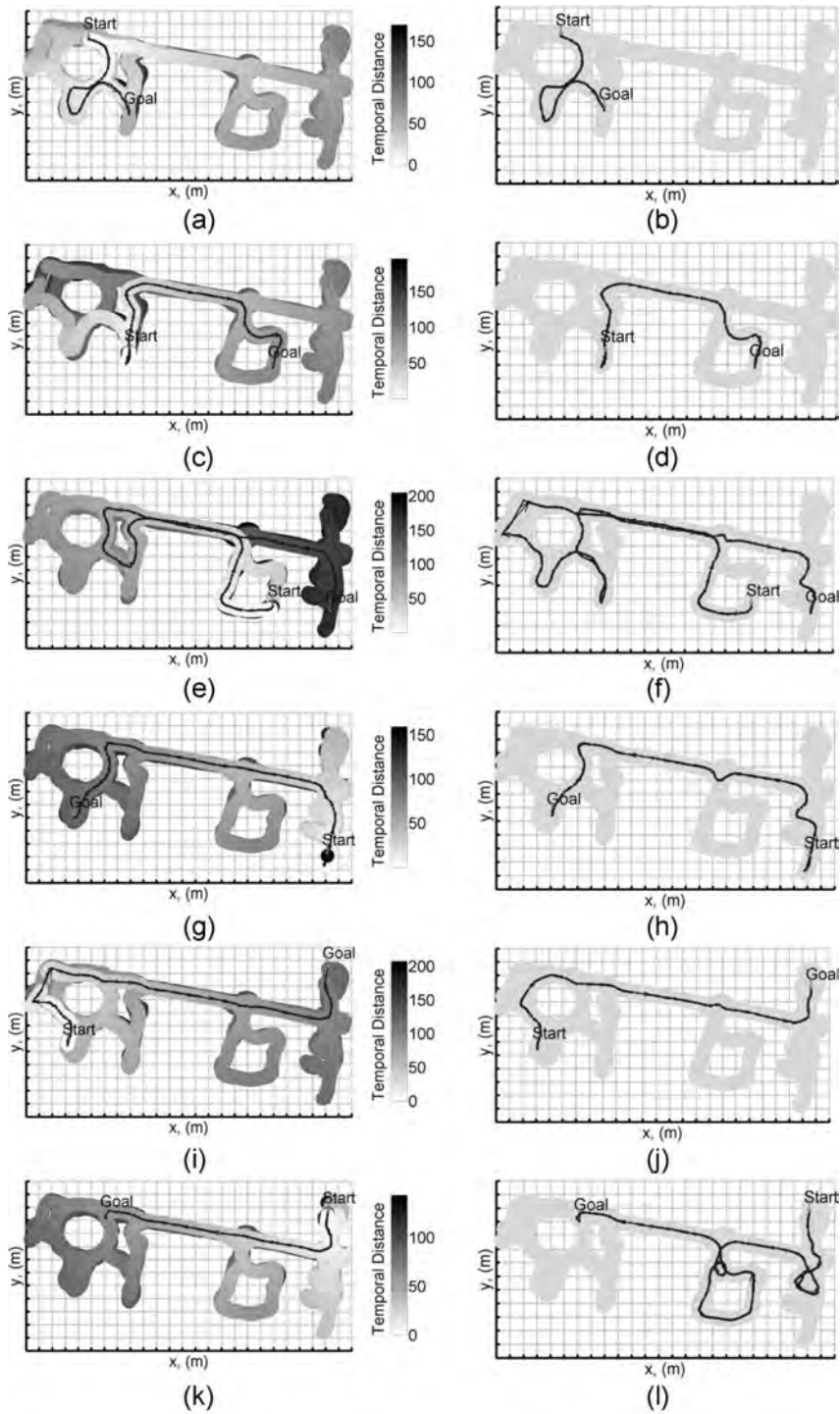


Figure 13. Temporal maps, planned and actual routes using the experience mapping algorithm for goals one to six (in order). Temporal maps and planned routes, with the temporal distance measured in seconds (a, c, e, g, i, k). Actual routes executed by the robot (b, d, f, h, j, l). Four of the executed routes were close to optimal or somewhat suboptimal (b, d, h, j). Two routes involved major movement errors (f, l). Overlapping experiences were rendered in order of their temporal value, with lower value layers rendered above higher value layers. Figure from (Milford, 2008) with permission.

The work carried out for this project raised a number of issues of interest. The RatSLAM model and experience mapping algorithm were evaluated through practical experimentation. Other computational models such as those of Arleo (2000) or Stringer et al. (2002) were evaluated in terms of their practical performance and their replication of biological observations. A set of theoretical analysis tools is required to complement the practical analysis of these systems. Being able to theoretically prove (or disprove) properties of these biologically inspired models like convergence and long-term stability, such as in the work by Cheung et al. (Cheung, Zhang, Stricker and Srinivasan, 2007), would facilitate their further development as practical robot mapping and navigation systems.

The work also touched on the issue of map usability – a map that looks good is of course useless if it cannot be used by the agent that created it, whether human, animal, or robot. Traditional robotic mapping representations such as high resolution grid maps are not necessarily optimal for integrated mapping and navigation systems. Animals navigate quite well without apparently forming any high resolution occupancy grid map. Future work on both probabilistic and biologically inspired mapping and navigation systems will benefit from a detailed study of the properties of a map that make it usable, and a start on this has been made in (Milford and Wyeth, 2007).

## CONCLUSION

In conclusion, the aim of this work was to demonstrate the potential for using biology as the inspiration for robotic mapping and navigation systems. In a relatively short period of time it was possible to develop, using the rodent brain as inspiration, a robotic control system that allowed a robot to explore, map, and navigate an unknown environment, and even adapt to very simple changes (Milford, 2008). The work on any particular stage of the project was necessarily relatively brief, but it is hoped that it will help spur further work in this area. Recent new discoveries about how

rats use their brains to map their environments, such as grid cells, are providing researchers in this area with rich new sources of inspiration (Solstad, Moser and Einevoll, 2006, Guanella and Verschure, 2006, Rolls, Stringer and Elliot, 2006, Burgess, Barry and O’Keefe, 2007). I hope and anticipate that we shall soon see robots equipped with artificial models of animal brains, adeptly navigating in our everyday lives using animal navigation systems derived from millions of years of evolution.

## ACKNOWLEDGEMENTS

The content of this paper is intended to present, in brief, an overview of some of the key themes emerging from a long-term research project into biomimetic navigation supported by grants from the Australian Research Council, and recently published as a book. I gratefully acknowledge the support and guidance of Gordon Wyeth, who accepted me to this project as a PhD student. This work is now being continued as part of the Australian Research Council and National Health and Medical Research Council Thinking Systems initiative. I also thank Jacqueline Hansson of IEEE and Thomas Ditzinger of Springer for their kind permission to reproduce certain figures in this paper.

## REFERENCES

- Arleo, A., 2000. Spatial Learning and Navigation in Neuro-mimetic Systems: Modeling the Rat Hippocampus. Verlag-dissertation, Germany.
- Arleo, A. & Gerstner, W., 2000. Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity, *Biological Cybernetics*, 83, 287–299.
- Browning, B. (2001). Using rat navigation models to learn orientation from visual input on a mobile robot. *Proceedings of Autonomous Agents*, 2001, Montreal, Canada.
- Buluswar, S.D. & Draper, B.A., 2002. Estimating apparent color in outdoor images. *Computer Vision and Image Understanding*, 85, 71–99.

- Burgess, N., Barry, C. & O’Keefe, J., 2007. An oscillatory interference model of grid cell firing. *Hippocampus*, 17, 801–812.
- Cheung, A., Zhang, S., Stricker, C. & Srinivasan, M.V., 2007. Animal navigation: the difficulty of moving in a straight line. *Biological Cybernetics*, 97, 47–61.
- Dissanayake, G., Newman, P.M., Clark, S., Durrant-Whyte, H. & Csorba, M., 2001. A solution to the simultaneous localisation and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17, 229–241.
- Frisch, K., 1967. *The Dance Language and Orientation of Bees*. Harvard University Press, Cambridge.
- Fyhn, M., Molden, S., Witter, M.P., Moser, E.I. & Moser, M.-B. (2004) Spatial representation in the entorhinal cortex. *Science*, 27, 1258–1264.
- Gaussier, P., Revel, A., Banquet, J.P. and Babeau, V., 2002. From view cells and place cells to cognitive map learning: processing stages of the hippocampal system. *Biological Cybernetics*, 86, 15–28.
- Gray, C.M., Konig, P., Engel, A.K. & Singer, W., 1989. Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature*, 338, 334–337.
- Guanella, A. & Verschure, P.F.M.J., 2006. A model of grid cells based on a path integration mechanism. *Lecture Notes in Computer Science*, 4131, 740–749.
- Hafting, T., Fyhn, M., Molden, S., Moser, M.-B. & Moser, E.I., 2005. Microstructure of a spatial map in the entorhinal cortex. *Nature*, 11, 801–806.
- Horizons, F., 2004. *Key Market Drivers Report. Future Horizons*, Kent.
- Milford, M.J., 2008. *Robot Navigation from Nature*, Springer-Verlag, Berlin-Heidelberg.
- Milford, M. & Wyeth, G., 2007. Spatial mapping and map exploitation: a bio-inspired engineering perspective. *Spatial Information Theory*, 4736, 203–221.
- Milford, M. & Wyeth, G., 2008. Single camera vision-only SLAM on a suburban road network. *International Conference on Robotics and Automation*. Pasadena, USA, IEEE.
- Milford, M.J., Wyeth, G. & Prasser, D., 2004. RatSLAM: a hippocampal model for simultaneous localization and mapping. *International Conference on Robotics and Automation*. New Orleans, USA, IEEE.
- Milford, M.J., Wyeth, G.F. & Prasser, D.P., 2006. RatSLAM on the edge: revealing a coherent representation from an overloaded rat brain. *International Conference on Robots and Intelligent Systems*. Beijing, China, IEEE.
- Milford, M.J. & Wyeth, G., 2003. *Hippocampal Models for Simultaneous Localization and Mapping on an Autonomous Robot*. Australasian Conference on Robotics and Automation. Brisbane, Australia, ARAA.
- Milford, M.J., Wyeth, G.F. & Prasser, D.P. 2004. Simultaneous Localization and Mapping from Natural Landmarks using RatSLAM. *Australasian Conference on Robotics and Automation*. Canberra, Australia, ARAA.
- Milford, M.J., Prasser, D.P. & Wyeth, G., 2006. Effect of Representation Size and Visual Ambiguity on RatSLAM System Performance. *Australasian Conference on Robotics and Automation*. Auckland, New Zealand, ARAA.
- Milford, M.J., Wyeth, G. & Prasser, D., 2005. Efficient Goal Directed Navigation using RatSLAM. *International Conference on Robotics and Automation*. Barcelona, Spain, IEEE.
- Mizumori, S.J. and Williams, J.D., 1993. Directionally selective mnemonic properties of neurons in the lateral dorsal nucleus of the thalamus of rats. *Journal of Neuroscience*, 13, 4015–4028.
- Moravec, H.P. & Elfes, A.E., 1985. High Resolution Maps from Wide Angle Sonar. *International Conference on Robotics and Automation*, St Louis, USA, IEEE.
- Muller, M. & Wehner, R., 1988. Path integration in desert ants, *Cataglyphis fortis*. *Proceedings of the National Academy of Science*, 85, 5287–5290.
- O’Keefe, J. & Conway, D.H., 1978. Hippocampal place units in the freely moving rat: Why



- they fire where they fire. *Experimental Brain Research*, 31, 573–590.
- O’Keefe, J. & Dostrovsky, J., 1971. The hippocampus as a spatial map: preliminary evidence from unit activity in the freely moving rat. *Brain Research*, 34, 171–175.
- Quirk, G.J., Muller, R.U. & Kubie, J.L. (1990). The firing of hippocampal place cells in the dark depends on the rat’s recent experience. *The Journal of Neuroscience*, 10, 2008–2017.
- Ranck, J.B., 1984. Head direction cells in the deep cell layer of dorsal presubiculum in freely moving rats. *Society Neuroscience Abstracts*, 10, 599.
- Ratnieks, F.L.W., 2000. How far do honey bees forage? *Bee Improvement*, 6, 10–11.
- Rolls, E.T., Stringer, S.M. & Elliot, T., 2006. Entorhinal cortex grid cells can map to hippocampal place cells by competitive learning. *Network: Computation in Neural Systems*, 17, 447–465.
- Solstad, T., Moser, E. & Einevoll, G., 2006. From grid cells to place cells: a mathematical model. *Hippocampus*, 16, 1026–1031.
- Stringer, S.M., Trappenberg, T.P., Rolls, E.T. & de Araujo, I.E.T., 2002. Self-organizing continuous attractor networks and path integration: two-dimensional models of place cells. *Network: Computation in Neural Systems*, 13, 217–242.
- Tews, A., Robert, J., Roberts, J. & Usher, K., 2005. Is the sun too bright in Queensland? An approach to robust outdoor colour beacon detection. Australasian Conference on Robotics and Automation. Sydney, Australia.
- Thiele, A. & Stoner, G., 2003. Neuronal synchrony does not correlate with motion coherence in cortical area MT. *Nature*, 421, 366–370.
- Tolman, E.C., 1948. Cognitive maps in rats and men. *The Psychological Review*, 55, 189–209.
- UNECE (2005) 2005 World Robotics survey. Geneva, UNECE. [http://www.unece.org/press/pr2005/05stat\\\_p03e.pdf](http://www.unece.org/press/pr2005/05stat\_p03e.pdf).
- Wehner, R., Gallizzi, K., Frei, C. & Vesely, M., 2002. Calibration processes in desert ant navigation: vector courses and systematic search. *Journal of Comparative Physiology A: Sensory, Neural, and Behavioral Physiology*, 188, 683–693.

---

Michael Milford  
 Queensland Brain Institute and School of  
 Information Technology and Electrical Engineering  
 The University of Queensland  
 St Lucia QLD 4072  
 Email: milford@itee.uq.edu.au

(Manuscript received 14.03.2008, accepted 27.06.2008)