Simultaneous Localisation and Mapping from Natural Landmarks using RatSLAM

Michael Milford, Gordon Wyeth, David Prasser
School of Information Technology and Electrical Engineering
The University of Queensland, St. Lucia, QLD 4072
{milford, wyeth, prasserd} @itee.uq.edu.au

Abstract

This paper describes the current state of RatSLAM, a Simultaneous Localisation and Mapping (SLAM) system based on models of the rodent hippocampus. RatSLAM uses a competitive attractor network to fuse visual and odometry information. Energy packets in the network represent pose hypotheses, which are updated by odometry and can be enhanced or inhibited by visual input. This paper shows the effectiveness of the system in real robot tests in unmodified indoor environments using a learning vision system. Results are shown for two test environments; a large corridor loop and the complete floor of an office building.

1 Introduction

Simultaneous Localisation and Mapping (SLAM) is one of the most challenging problems that must be solved before we have truly autonomous robots. Typical approaches involve the use of grid representations [Thrun, 1998], landmark representations [Nieto et al., 2003] or topological representations [Endo and Arkin, 2003]. Regardless of the representation used, the systems that have produced the most impressive practical results to date have been probabilistic to some degree.

There have also been a large number of biologically inspired navigation and mapping approaches based on animal systems such as the rodent hippocampus. Although these systems have been valuable in exploring various neuroscience theories of mapping and navigation in the brain, they have not been able to produce comparable practical results. RatSLAM[1] has been developed from models of the hippocampal complex in rodents, but with a strong focus on producing competitive real world SLAM results. Although there is still much debate over some of the details of these models, there are areas of key agreement.

It is generally agreed that rodents have *place fields*, patterns of neural activity that correspond to locations in space [Arleo, 2000; Gaussier et al., 2001]. Early work with the RatSLAM model revealed the inability of place field models to represent and propagate multiple hypotheses of pose [Milford and Wyeth, 2003]. The system was developed to incorporate *pose* cells to overcome this limitation. Subsequent investigation of the literature revealed results from biological experimentation supporting the pose cell concept [Knierim, 2002; Redish, 1999].

RatSLAM has already been demonstrated performing SLAM in an indoor environment with artificial landmarks [Milford et al., 2004]. Further work has resulted in a new natural vision module [Prasser et al., 2004] which has been integrated into the system. This and significant improvements in already existing components have allowed the testing of RatSLAM in a range of large, complex indoor environments.

This paper proceeds with the following structure. Section 2 describes the state of the art in robotic SLAM. Section 3 details the overall RatSLAM architecture and defines what type of SLAM method it is by comparing it with current techniques. The actual implementation of each system module is described in Section 4. Section 5 covers the experimental setup and the test environments. Results of SLAM tests in these environments are shown and discussed in Section 6, before the paper concludes in Section 7.

2 Background

There have been a large number of approaches to robotic mapping over the last few decades. Traditionally these approaches could be classified into *metric* or *topological* mapping methods. Metric mapping approaches attempt to represent the geometric properties of an environment. One of the seminal metric mapping techniques was the *Occupancy Grid Algorithm* by Elfes [Elfes, 1987].

Topological approaches represent important or interesting places in an environment and store information on how these places are connected. In addition topological methods often store information on actions required to navigate between these places or *nodes* as they are often called. The metric/topological divide however is an arbitrary one, and has progressively become less useful as hybrid approaches are developed that incorporate aspects of both conceptual categories. For example, recent work by Endo [Endo and Arkin, 2003] has involved the formation of a cognitive map incorporating both spatial and topological information. RatSLAM is a hybrid approach with both metric and topological characteristics.

[1] This research is sponsored in part by an Australian Research Council grant.

In the last decade robotic mapping has become known as CML (Concurrent Mapping and Localisation) or SLAM. Major advances in computing power and the introduction of new statistical methods for solving the SLAM problem [Smith et al., 1990] have allowed the development of a large range of probabilistic techniques using Kalman filters [Dissanayake et al., 2001] or the Expectation Maximisation (EM) algorithm by Dempster [Dempster et al., 1977]. The best practical robotic mapping results to date have mostly come from some variation on or combination of Occupancy Grid, Kalman filter or Expectation Maximisation methods.

Kalman filter approaches describe the location of features or landmarks in the environment, which themselves are obtained through processing of sensory input. A typical feature might be vertical lines in an image obtained from a camera. Uncertainties in landmark (feature) locations and robot pose are represented by Gaussian distributions. The Gaussian model comprises the full state vector *x*, given by:

$$x_t = (s_{x,t}, s_{y,t}, s_{\theta,t}, m_{1,x,t}, m_{1,y,t}, ... m_{K,x,t}, m_{K,y,t})^T \quad (1)$$

where the robot's co-ordinates are represented by s_x, s_y, s_θ, the co-ordinates of the k^{th} feature are represented by $m_{k,x}$, $m_{k,y}$, $m_{k,\theta}$, and K is the total number of landmarks [Thrun, 2002].

The original Kalman filter approach relies on two assumptions. Firstly, the function that describes the future motion and perceptual states must be a linear one with superimposed Gaussian noise. This is addressed by using Taylor series expansions to linearize the robot motion and sensor models. Secondly, the uncertainty in robot pose and landmark position must be of Gaussian form. This second assumption means that basic Kalman filter techniques struggle when there are multiple indistinguishable landmarks in the environment [Thrun, 2002]. Recent work on *multi hypothesis Kalman filters* has combined Gaussian representations with particle filters in order to maintain a small number of the most likely modes [Montemerlo et al., 2002]. In this way a limited amount of landmark ambiguity can be accommodated, but without the un-computable problem of having to maintain the full posterior under unknown correspondences.

Expectation Maximisation algorithms are much more suited to solving the correspondence problem in robotic mapping than Kalman filters, and can still function in highly ambiguous environments. EM methods involve iterations of a two step process. The *expectation step* calculates the possible robot paths through the environment given a specific map and the sensor readings to date. The *maximisation step* calculates the most likely map given all the possible robot paths. Under certain conditions the maps progressively become more accurate each iteration. The maximisation function is:

$$m^{[i+1]} = \underset{m}{\operatorname{argmax}} E_{S^t}\left[\log p\left(d^t, s^t \mid m\right) \mid m^{[i]}, d^t\right] \quad (2)$$

where s^t is the robot's pose at time t, d^t is the sensor information at time t and $m^{[i]}$ is the map produced by the i^{th} EM iteration [Thrun, 2002].

Currently no EM method is able to deal with a continuous map. Instead a high resolution grid map is used, and the maximisation problem is solved for each grid cell independently. Grid cells typically have a continuous rather than discrete range of values. This helps to decrease the chances of the iteration process converging to local maxima, which is one of the major weaknesses of EM approaches. This weakness is caused because they hill-climb towards a best map. Extensions of the EM algorithm such as the *Lu/Milios* algorithm guess correspondences and can allow the system to recover from small pose estimate errors.

The relatively high computational requirements of EM methods restrict them to being an off robot and often offline technique on current robotic platforms. Despite these limitations EM techniques have managed to produce some impressive real world results especially in large scale loop environments, where they have demonstrated an ability to 'close the loop' even in ambiguous environments such as in Figure 1.

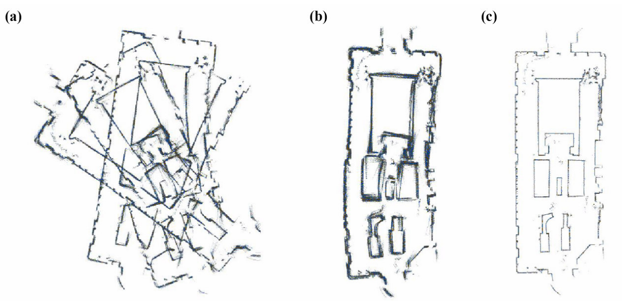

Figure 1 – (a) Unprocessed occupancy grid map. (b) Occupancy grid map after EM processing. (c) Occupancy grid map after application of Lu/Milios algorithm to pre-aligned data [Thrun, 2002].

3 RatSLAM Architecture

This section starts by describing the overall structure of the RatSLAM system and providing an overview of each major component. It then compares and contrasts RatSLAM with existing SLAM techniques, and discusses exactly what type of SLAM technique it is.

Figure 2 shows the structure of the RatSLAM system. The robot's pose is represented by activity in a competitive attractor neural network called the *pose cells.* Wheel encoder information is used to perform path integration by appropriately shifting the current pose cell activity. Vision information is converted into a *Local View* (LV) representation that is associated with the currently active pose cells. If familiar, the current visual scene also causes activity to be injected into the particular pose cells linked to the current scene in the *Pose-View* map.

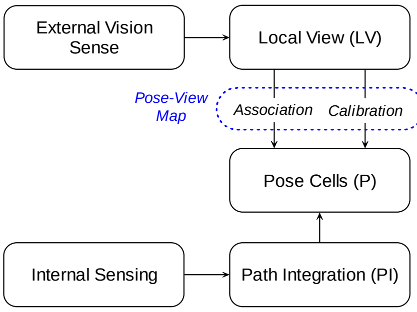

Figure 2 – Overview of RatSLAM system

3.1 Pose Cells

The pose cells are implemented as a competitive attractor network, a type of neural network that converges to a stable pattern of activation across its inputs. The network units can be arranged in many configurations, but generally each unit will excite units close to itself and inhibit those further away, which leads to a clump of activity known as an *activity packet* eventually dominating. Activity injected into the network near this winning packet will tend to move that packet towards it. Activity injected far away from it will create another packet that competes with the original. If enough activity is injected the new packet can 'win' and the old packet disappear. The pose cell structure is an extension on models of place cells in the rodent hippocampus [Arleo, 2000].

By using a competitive attractor network the robot is able to simultaneously represent multiple position estimates in x, y and θ. We arrange the pose cells in an (x, y, θ) arrangement for ease of visualisation although there is no biological justification for this sort of ordered arrangement.

3.2 Path Integration

The path integration process updates pose based on robot wheel encoder velocities. The process is not meant to be strictly Cartesian, although the implementation described here is roughly Cartesian to aid in visualisation and weight assignment. For indoor experiments each pose cell represents approximately 0.25m x 0.25 m in area and approximately 10 degrees in bearing.

The original RatSLAM path integration system was quite inferior to the on-board robot path integration (separate to our system). It was based on a biologically plausible model of path integration in the rodent hippocampus. Extensive experimental testing was carried out using this path integration system. However, in order to maximize the chances of success for the next stages of the project, which involve goal memory and mapping of very large areas, it was decided to adopt a more practical path integration model that could provide better performance. The new path integration model, which is described further in section 4.3, achieves quite respectable path integration performance and can be relied on to be reasonably accurate over short time periods.

3.3 Local View

The Local View module is a collection of neural units that represent what the robot sees in a form usable by the pose cells. In the original system the Local View cells formed a three-dimensional matrix; which cells were activated depended on the colour, bearing and distance to coloured cylinders in the environment [Prasser and Wyeth, 2003]. In the new natural vision implementation the Local View cells form a one-dimensional structure (Figure 3). This representation layer is several layers removed from the raw camera data. The Hebbian learning that RatSLAM uses cannot usefully associate raw camera data with pose cells. The data must be processed to reduce the dimensionality of the camera image whilst preserving distinctive information.

Our approach to vision processing has two layers: a biologically motivated feature extraction of the image followed by a primitive scene recognition stage. The output of the system is a group of neural units – the Local View structure - each of which responds to a different visual scene. The details of the vision system architecture are described in [Prasser, et al., 2004].

3.4 Defining RatSLAM

RatSLAM is a hybrid method – it has characteristics of topological, grid based and landmark/feature SLAM techniques. Like Kalman filter approaches, RatSLAM uses visual features or landmarks, but does not require that all landmarks in the environment be uniquely identifiable. Instead the system uses ambiguous visual input to support any number of robot pose hypotheses, which can converge to the correct hypothesis without the robot ever seeing a unique landmark.

Unlike EM methods, RatSLAM is an online algorithm that performs SLAM in real-time during actual experiments on a continuously moving robot. Path integration errors are corrected by topological 'snaps' in the map rather than iterative correction of a global metric map. Even if the system makes an incorrect localisation snap, with further visual input it can recover and still produce a topologically usable map. Indeed, as the testing environment becomes more complex, the topological characteristics of the RatSLAM maps become more dominant, but whilst still retaining some local spatial properties.

Furthermore RatSLAM has characteristics of grid-based approaches, but without the strict adherence to a metric map. Each pose cell starts off corresponding to an approximate physical area and robot orientation, but can be stretched, compressed or twisted during mapping to represent an arbitrary physical pose space. Topological jumps in the pose cell structure can even remove cells that no longer represent any physical space in the environment.

RatSLAM combines beneficial characteristics of the three major SLAM methodologies whilst avoiding many of their associated weaknesses. It is a truly hybrid approach, but has been developed independently of the methodologies it shares properties with, rather than being created from them. As such RatSLAM occupies a new, unique niche in the field of Robotic SLAM.

4 Implementation

The RatSLAM system is centred around iterations of the pose cell competitive attractor network. Each iteration consists of the following steps:

1. Visual association – Pose-View map update
2. Visual calibration
3. Path integration
4. Competitive attractor network dynamics

The following sections detail the dynamics involved in each step.

4.1 Visual Association

The visual association process associates scenes with robot poses to create a Pose-View map. RatSLAM uses this map to keep the robot localised - vision based localisation is needed to correct for drift in the path integration system. In larger loop style experimental arenas visual information allows the robot to 'close the loop'. When the experimenter kidnaps the robot, visual information allows the robot to re-localise to the correct position.

Figure 3 shows a link between an active Local View cell and an active pose cell. Connection strengths between the Local View and pose cells are strengthened using Hebbian learning, by:

$$\beta^{t+1}_{(i)(lmn)} = \beta^{t}_{(i)(lmn)} + \eta\left(P_{lmn}V_i - \delta\right). \tag{3}$$

where β is the strength of the connection between the Local View and pose cells, η is the learning rate, P is the activation level of the pose cell and V is the activation level of the Local View cell. δ is a temporal decay constant that is currently set to zero (no map decay over time).

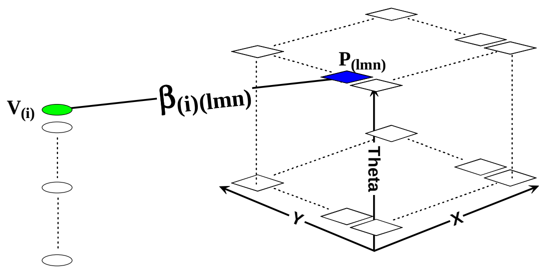

Figure 3 - Illustration of the Local View network and pose cell network. Activated units in the Local View become associated with activated units in the pose cells through strengthening of connections between the two networks.

4.2 Visual Calibration

When the robot encounters a familiar visual scene, energy is injected from activated Local View cells into the pose cells through non-zero weighted connections in the Pose-View map:

$$P^{t+1}_{lmn} = P^{t}_{lmn} + \min\left(\beta_{(i)(lmn)}V_i,\ (\beta V)_{\max}\right) \tag{4}$$

where $(\beta V)_{max}$ is the saturation level of a Pose-View link.

4.3 Path Integration

Early versions of the RatSLAM path integration module were based on models of path integration in the rodent hippocampus [Arleo et al., 2001; Stringer et al., 2002]. These models were found to be adequate for the original separated *place cell – head direction* representations, but had degraded performance when used with a single pose cell structure. The models were also only valid over a limited velocity range and required a constant, high frequency update regime. The current RatSLAM path integration method shifts activity rather than projecting a copy of the current activity forwards in time. This makes its performance independent of variable sensory update rates and robot velocity and in practice produces more accurate robot paths, as well as virtually removing the need for parameter tuning.

Equation 5 shows how activity is shifted and re-injected back into the pose cell structure. The amount of activity injected is based on the product of the activity of the sending pose cell unit, P, and a residue component, α. The residue component is spread over a 2×2×2 cube to account the quantisation effects of the grid representation. It is based on the fractional components of the offsets, δx_f, δy_f and $\delta\theta_f$ which themselves are based on the translational velocity v, rotational velocity ω and preferred cell orientation θ.

$$P^{t+1}_{lmn} = \sum_{x=\delta x_o}^{\delta x_o+1} \sum_{y=\delta y_o}^{\delta y_o+1} \sum_{z=\delta\theta_o}^{\delta\theta_o+1} \alpha_{xyz} P^{t}_{(l+x)(m+y)(n+z)} \tag{5}$$

where δx_0, δy_0, $\delta\theta_0$ are the rounded down integer offsets in the x, y and θ directions. The integer and fractional offsets are given by Equations 6 and 7 respectively:

$$\begin{bmatrix} \delta x_o \\ \delta y_o \\ \delta\theta_o \end{bmatrix} = \begin{bmatrix} \lfloor k_x v \cos\theta \rfloor \\ \lfloor k_y v \sin\theta \rfloor \\ \lfloor k_\theta \omega \rfloor \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} \delta x_f \\ \delta y_f \\ \delta\theta_f \end{bmatrix} = \begin{bmatrix} k_x v \cos\theta - \delta x_o \\ k_y v \sin\theta - \delta y_o \\ k_\theta \omega - \delta\theta \end{bmatrix} \tag{7}$$

The residue matrix α is calculated using Equations 8 and 9:

$$\alpha_{ijk} = g\left(\delta x_f, i - \delta x_o\right) g\left(\delta y_f, j - \delta y_o\right) g\left(\delta\theta_f, k - \delta\theta_o\right) \tag{8}$$

$$g(a,b) = \begin{cases} 1-a, & b=0 \\ a, & b=1 \end{cases} \tag{9}$$

4.4 Competitive Attractor Network

Competitive attractor dynamics ensure that the total activity in the pose cells remains constant. This is consistent with the interpretation of activity in the pose cells as a probability distribution of pose. Activity packets located near each other move towards and reinforce each other, coalescing similar pose representations. Separated activity packets representing multiple hypotheses of pose compete with each other. Figure 4 shows a snapshot of typical activity in the pose cell structure, with multiple activity packets. Global inhibition means that without visual or path integration input the activity will eventually stabilize to one packet, and hence one pose hypothesis. There are three stages to the internal network dynamics:

1. Excitation
2. Global inhibition of all cells, and
3. Normalisation of pose cell activity

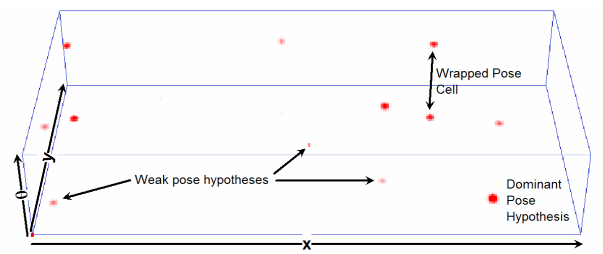

Figure 4 – Snapshot of activity/energy in the pose cells. The three-dimensional pose cell matrix wraps in all three axis directions. An example of a packet wrapping in the θ direction is shown in the diagram.

Excitation

The excitatory weight matrix, ε, is created from a three dimensional discrete Gaussian distribution. This matrix is used to project activity from each cell P to surrounding cells:

$$\Delta P_{ijk} = \sum_{a=0}^{N_X} \sum_{b=0}^{N_Y} \sum_{c=0}^{N_\theta} \varepsilon_{(a-i)(b-j)(c-k)} P_{abc} \qquad (10)$$

where N_X, N_Y, N_θ are the dimensions of the pose cell matrix.

Global Inhibition

Each network iteration every pose cell undergoes a fixed level of inhibition independent of its current activity level, with cell activity limited to non-negative values:

$$P_{ijk}^{t+1} = \max\left(P_{ijk}^{t} - \varphi,\ 0\right) \qquad (11)$$

where φ is the inhibition constant, which influences packet size and longevity. The fixed inhibition level has been found to give more desirable network dynamics then an activity dependent level.

Normalisation

Normalisation is the last network process to occur each iteration and maintains the total activation level at unity.

$$P_{ijk}^{t+1} = \frac{P_{ijk}^{t}}{\sum_{x=0}^{N_X} \sum_{y=0}^{N_Y} \sum_{z=0}^{N_\theta} P_{xyz}^{t}} \qquad (12)$$

5 Experimental Setup

RatSLAM is a flexible SLAM system and can in theory be employed on most platforms that have vision, range and path integration suitable sensors. Most of the testing to date has been carried out on a Pioneer 2DXE mobile robot equipped with a 50 degree field of view forward facing camera, eight sonar sensors and wheel encoders. RatSLAM is currently being tested on an outdoor autonomous tractor, however this paper will focus on its performance using the Pioneer robot platform.

The visual complex cell processing is performed on the robot's onboard 400 MHz Athlon K6 processor and the results are wirelessly transmitted to a 1.1 GHz Pentium III laptop where the rest of the RatSLAM system resides. The iteration speed averages 10 Hz but can vary throughout an experiment without adverse effects.

Extensive testing has been carried out in several different environments of which two are presented here. Environment A was a corridor loop of length 70 metres at the Queensland Centre for Advanced Technologies (Figure 5, Figure 6). The robot’s control scheme used the sonar array to perform autonomous wall following. SLAM was performed live during the actual test.

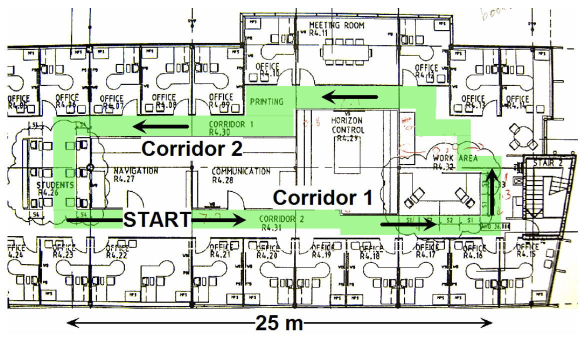

Figure 5 - Floor plan of loop environment (environment A). The shaded area and arrows indicate the path of the robot.

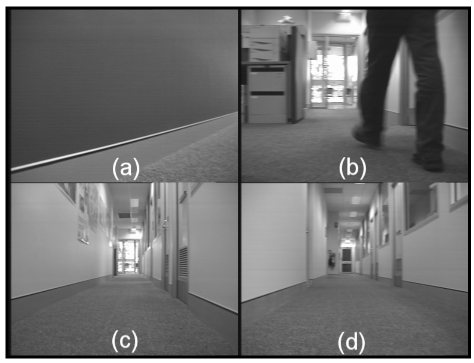

Figure 6 – Robot camera captures from loop environment. (a) Dark uniform wall (b) Bright window and person (c) Corridor 1 (d) Corridor 2.

Environment B was the complete floor of a building at our university campus measuring about 45 metres by 13 metres (Figure 7, Figure 8). The floor layout of this building is complex and there are multiple possible pathways between most points. To simulate movement of the robot under an exploration / goal memory system currently in development, the robot was tele-operated. At intersections the robot was driven down a 'random' path. Excursions were made into offices, meeting rooms and laboratories as well as open plan spaces. Routes were traversed both forwards and backwards.

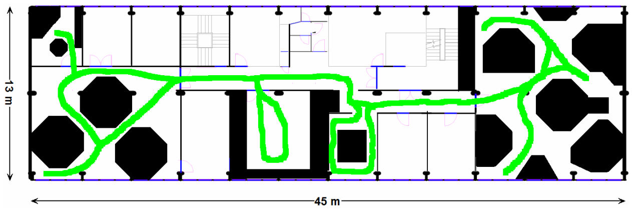

Figure 7 – Complete floor environment (environment B). Note that only large obstacles are shown, there were a host of small irregular objects such as people, chairs and boxes present. The routes traversed by the robot are shown in the diagram.

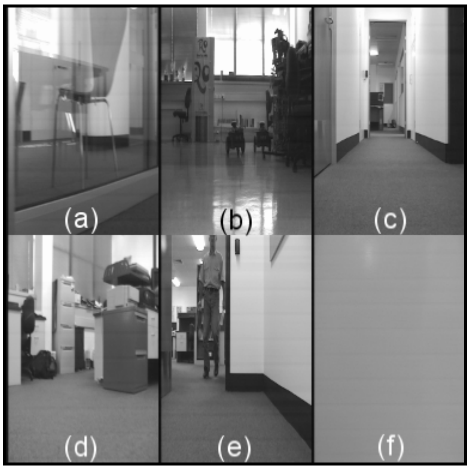

Figure 8 - Robot camera captures from Environment B. (a) Reflective glass (b) Laboratory (c) Corridor (d) Open Plan Space (e) Person and doorway (f) Featureless wall.

Results presented are for tests performed during work hours. Both environments were dynamic – during tests people were moving about, doors were being opened or closed and lighting conditions changed as can be seen in Figure 6(b) and Figure 8(e). The visual complexity of the environment was increased further by lighting and surface conditions such as floor and glass reflections shown in Figure 8(a) and Figure 8(b). Separate corridor sections such as in Figure 6(c) and Figure 6(d) appeared similar to the robot's camera.

6 Results

This section shows and discusses the mapping results for the two environments. Results for environment A were generated in a completely autonomous and online manner – the robot's movement was autonomous and SLAM was performed during the actual test. Results for environment B were generated with the robot being tele-operated. A summary of the experimental conditions is shown in Table 1:

	Environment A	**Environment B**
Environment Description	Corridor Loop	Corridors Open plan space Offices
SLAM	Real-time	Real-time
Robot Control	Autonomous wall following	Driven
Translational Velocity	0 – 0.5 m/s Average = 0.36 m/s	0 – 0.5 m/s Average = 0.36 m/s
Rotational Velocity	0 – 40 degrees / second	0 – 90 degrees / second
Total Distance	~215 m	~400 m
Total Time	600 seconds	1100 seconds

Table 1 – Experiment summaries. Environment B was very cluttered and hence, even with a human in control, the robot could not be driven any faster than when the wall following behaviour was used.

6.1 Performance Indicators

The representations that RatSLAM creates of its environment are not strictly Cartesian although local areas within the representations may have strong Cartesian properties. Consequently conventional performance metrics analysing the closeness of a geometric map with the physical reality are not suitable. Instead we analyse the performance of the system by using indicators that illustrate system consistency over time.

Until recently two test indicators have been used – consistency in measured trajectory and consistency in the mapped locations of artificial landmarks. Uncorrected robot odometry drifts in an unbounded manner; the RatSLAM system produces trajectories that are bounded and consistent over time. The recent addition of an occupancy grid generator to the RatSLAM system has provided a third performance indicator, which is discussed in the following section.

Occupancy Grid Map

We use a simplified version of Moravec and Elfes' 1985 model of a wide angle sonar beam and a Bayesian update scheme to create occupancy grid maps [Elfes, 1987; Moravec and Elfes, 1985]. RatSLAM produces occupancy grid maps that become stable over time. On a local scale the occupancy maps spatially match parts of the environment, but are discontinuous on a global scale. As environments become larger and more complex the maps become increasingly discontinuous. This is to be expected; larger environments will involve longer periods of exploration without familiar visual input and hence larger re-localisation snaps when re-entering familiar territory. Consistency and stability over time in the occupancy grid maps indicate that RatSLAM has built a stable representation of the world.

It should be noted that the occupancy maps shown here *are not* the Pose-View maps that RatSLAM uses to represent the environment. They are shown because they are more visually intuitive than a representation of the Pose-View map. Also note that for all the results presented here, a square in the occupancy grid roughly represents a spatial area of 0.25 m by 0.25 m.

6.2 Path Integration Only Performance

In both environments tests were run using only the RatSLAM path integration module, with no visual association or calibration. Small errors in the reported velocities obtained from the wheel encoders accumulated over time, shown clearly in the final trajectory and occupancy grid maps (Figure 9, Figure 10).

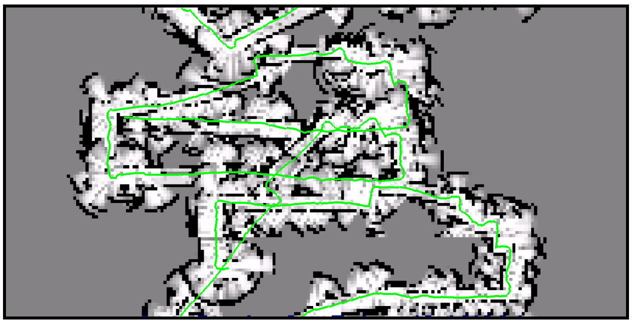

Figure 9 – The robot rapidly becomes lost when relying only on path integration in a loop environment (Environment A).

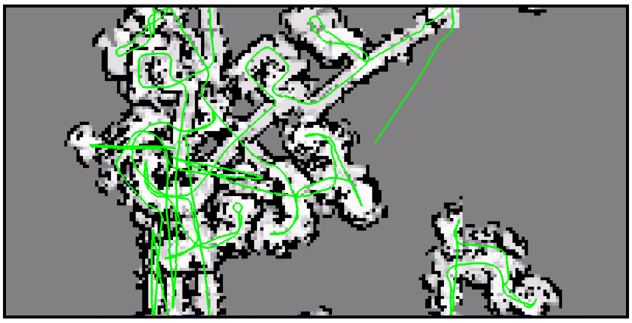

Figure 10 – There is no observable repeatability in the trajectory map for Environment B.

6.3 Loop SLAM

Figure 11 shows the trajectory and occupancy grid map for the loop environment when the complete RatSLAM system is used. The trajectory is consistent over repeated laps, and at the start of each new lap the robot is able to repeatedly close the loop. Also, at other stages of the lap the system is able to recover from path integration errors.

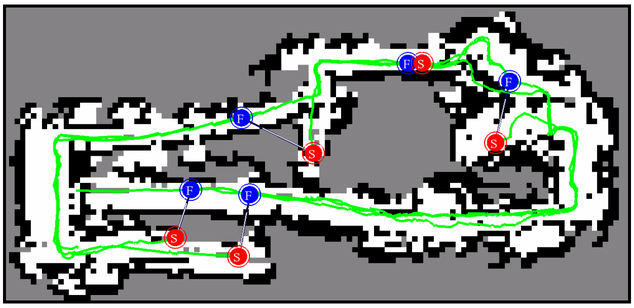

Figure 11 – RatSLAM enables the robot to close the loop upon starting the 2nd lap and so on, and also recover from path integration errors at other points. Red circles with an 'S' and blue circles with a 'F' indicate the start and finish of re-localising snaps.

The robot did not close the loop at exactly the same point because of the jerkiness of the wall following behaviour. The extremely poor sonar performance in these indoor environments meant that the robot adopted a slight zigzagging path travelling down a corridor. With only an effective field of view of 50 degrees, the robot experienced very different visual sequences on different laps. Part of the aim of current work on an exploration scheme is to create robot movement that is more suitable for such a narrow field of view camera.

Kidnap Recovery

In this series of experiments we let the robot complete one or two laps of the testing arena, before kidnapping the robot, without any communication to the robot that it was being kidnapped. The initial kidnaps consisted of simply mid-experiment picking the robot up and rotating it 90 degrees before putting it down again. The trajectory plots were not affected by this type of kidnap – the RatSLAM system is capable of re-localising the robot almost instantaneously in response to this type of kidnap.

We increased the severity of the problem by kidnapping the robot in both position and orientation, once again without 'informing' the robot that it had been kidnapped. After these global kidnaps the RatSLAM system was still able to re-localise, after sufficient visual information was obtained. Figure 12 shows how the robot was physically kidnapped from point A a few metres backwards to point B and also rotated. Immediately multiple pose hypotheses were introduced into the system, and after travelling for a while the robot was eventually able to re-localise.

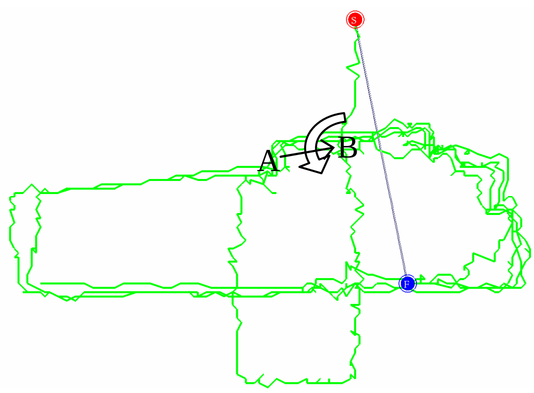

Figure 12 – The robot was kidnapped in both position and orientation from point A to point B. After travelling 'lost' for a while, it was able to eventually re-localise to the correct position.

As with closing the loop, there is currently no backwards in time correction of the map after re-localising snaps. This does not prevent RatSLAM from creating maps that are *topologically* (although not spatially) usable even when subjected to kidnapping. The author knows of no real-time SLAM methods that have produced *spatially* consistent maps in experiments where the robot was kidnapped. We propose that RatSLAM's topological / locally spatial maps are sufficient for the robot to perform efficient goal orientated navigation. Current work is investigating this hypothesis, which is discussed further in Section 7.

6.4 Complex Environment SLAM

Figure 13 and Figure 14 show the perceived RatSLAM robot trajectories and occupancy grid maps for Environment B using two different learning rates.

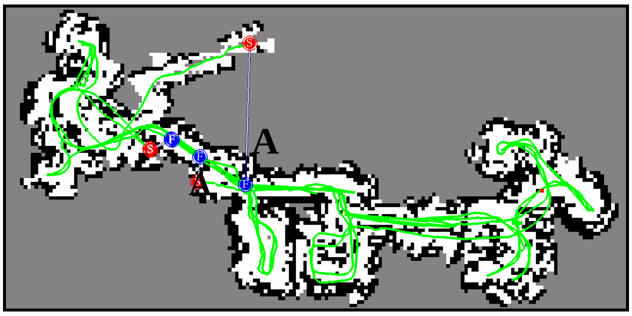

Figure 13 – Trajectory and occupancy map of the complete floor for $\eta = 0.20$. Note the re-localising snaps. Also note the bend in the corridor at A, which doesn't exist in the actual environment.

RatSLAM stays well localised for most of the experiment shown in Figure 13 except for one significant deviation finally corrected by a re-localisation snap. The two other, much smaller re-localisation snaps represent recovery from momentary path integration errors.

Figure 14 - Trajectory and occupancy map of the complete floor for $\eta = 0.25$. Note the consistency of the robot trajectory.

With a more suitable learning rate one obtains the map shown in Figure 14. The RatSLAM system is very well localised at all times except during one short excursion into a meeting room, but it re-localises immediately upon exiting the room. Although this is not our purpose, the occupancy grid map would be sufficient for performing goal orientated navigation. This is a good result especially considering the very poor performance of the sonar sensors in this environment. We are currently investigating the use of these occupancy grid maps to supplement the exploration scheme and also to instantly correct momentary path integration errors.

6.5 Localisation Accuracy

Absolute measures of localisation accuracy are hard to obtain for two reasons – the lack of a ground truth measurement device such as indoor GPS, and the non-Cartesian nature of the RatSLAM representations. However, one can make some quantitative localisation observations with respect to consistency of the representation. By observing the correspondence between points on the trajectory and robot poses in the real world, one can compare the localisation accuracy in terms of consistency on subsequent passes through an area.

In the experiment shown in Figure 14 the robot's most likely pose is consistent to within two pose cells (approximately 0.5 metres) with previous passes through the environment for the entirety of the experiment except for one 33 second period. During this period the error increased gradually to about 3 metres in position and 25 degrees in orientation. The error was then corrected by a re-localising snap.

6.6 Discussion

RatSLAM has been extensively tested for over a year, in at least six distinctly different environments and on two very different robotic platforms. During this time it has become apparent that the SLAM performance is primarily dependent on having an appropriate learning rate for the Pose-View map. A low learning rate allows the system to go on learning and associating ambiguous visual input indefinitely, but increases the length of visual input required for it to re-localise. Conversely, a high learning rate gives the system rapid re-localisation ability, but limits how long the system can go on learning ambiguous input. This is a robotics specific version of Grossberg's *stability-plasticity* dilemma [Hertz et al., 1991].

All SLAM systems face this dilemma – the trade off between retaining plasticity and continuing to learn indefinitely, and retaining stability and recognising enough visual scenes to maintain localisation. Future work will address developing an intelligent learning rate control module for the RatSLAM system. This module will dynamically vary the learning rate based on the strength of visual input into the pose cells and the confidence in the robot's localisation during an experiment.

7 Conclusion

RatSLAM has developed to the stage where it can autonomously wander around a complex environment and perform SLAM online. The system is able to recover from major path integration errors and even deliberate kidnap attempts. RatSLAM's robust mapping performance in dynamic, unmodified real world environments is achieved through the probabilistic nature of the system rather than any specific and hence brittle techniques.

Over the last year, a large amount of experimental data has been gathered using the RatSLAM system. There has also been much feedback from the general research community. This and comparison to the original biological models that inspired RatSLAM has allowed the identification of key areas for future work.

The current autonomous control scheme of the robot is based on low level reactive behaviours. Development of an efficient exploration scheme and a goal memory system will embody the robot with more advanced navigation abilities. Rather than passively mapping the environment, RatSLAM will be able to intelligently and actively carry out exploration. Goal memory will allow the ultimate test of the RatSLAM system: can it use its environmental representations to perform useful tasks?

Acknowledgements

Sincere thanks go to the staff of the CSIRO Mining Automation group for allowing us to map their office space in our experiments.

References

[S. Thrun, 1998] S. Thrun, "Learning Maps for Indoor Mobile Robot Navigation," *Artificial Intelligence*, vol. 99, pp. 21-71,

[J. Nieto, et al., 2003] J. Nieto, et al., "Real Time Data Association for FastSLAM," *International Conference on Robotics & Automation*, 2003, Taipei, Taiwan.

[Y. Endo and R. C. Arkin, 2003] Y. Endo and R. C. Arkin, "Anticipatory Robot Navigation by Simultaneously Localizing and Building a Cognitive Map," *Intelligent Robots and Systems*, 2003, Las Vegas.

[A. Arleo, 2000] A. Arleo, "Spatial Learning and Navigation in Neuro-Mimetic Systems: Modeling the Rat Hippocampus," PhD, University of Milan, Milan, 2000.

[P. Gaussier, et al., 2001] P. Gaussier, et al., "From view cells and place cells to cognitive map learning: processing stages of the hippocampal system," *Biological Cybernetics*,

[M. J. Milford and G. Wyeth, 2003] M. J. Milford and G. Wyeth, "Hippocampal Models for Simultaneous Localisation and Mapping on an Autonomous Robot," *Australian Conference on Robotics and Automation*, 2003, Brisbane, Australia.

[J. Knierim, 2002] J. Knierim, "Dynamic Interactions between Local Surface Cues, Distal Landmarks, and Intrinsic Circuitry in Hippocampal Place Cells," *Journal of Neuroscience*, vol. 22, pp. 6254-6264, July 15, 2002.

[D. Redish, 1999] D. Redish, 1999 *Beyond the Cognitive Map*. Massachusetts: Massachusetts Institute of Technology.

[M. J. Milford, et al., 2004] M. J. Milford, et al., "RatSLAM: A Hippocampal Model for Simultaneous Localization and Mapping," *International Conference on Robotics and Automation*, 2004, New Orleans.

[D. Prasser, et al., 2004] D. Prasser, et al., "Biologically Inspired Visual Landmark Processing for Simultaneous Localization and Mapping," *International Conference on Intelligent Robots and Systems*, in press, 2004, Sendai, Japan.

[A. Elfes, 1987] A. Elfes, "Sonar-based Real-world Mapping and Navigation," *IEEE Journal of Robotics and Automation*, vol. 3, pp. 249-265,

[R. Smith, et al., 1990] R. Smith, et al., "Estimating Uncertain Spatial Relationships in Robotics," *Autonomous Robot Vehicles*, pp. 167-193, 1990.

[G. Dissanayake, et al., 2001] G. Dissanayake, et al., "A solution to the simultaneous localisation and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 229-241,

[A. P. Dempster, et al., 1977] A. P. Dempster, et al., "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, pp. 1-38,

[S. Thrun, 2002] S. Thrun, 2002 "Robotic Mapping: A Survey," Carnegie Mellon University, Pittsburgh, Faculty Report.

[M. Montemerlo, et al., 2002] M. Montemerlo, et al., "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem," *AAAI National Conference on Artificial Intelligence*, 2002, Edmonton, Canada.

[D. Prasser and G. Wyeth, 2003] D. Prasser and G. Wyeth, "Probabilistic Visual Recognition of Artificial Landmarks for Simultaneous Localization and Mapping," *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, 2003, Taipei.

[A. Arleo, et al., 2001] A. Arleo, et al., "Place Cells and Spatial Navigation based on Vision, Path Integration, and Reinforcement Learning," *Advances in Neural Information Processing Systems*, vol. 13, 2001.

[S. M. Stringer, et al., 2002] S. M. Stringer, et al., "Self-organizing continous attractor networks and path integration: two-dimensional models of place cells," *Computational Neural Systems*, vol. 13, pp. 429-446, 2002.

[H. P. Moravec and A. E. Elfes, 1985] H. P. Moravec and A. E. Elfes, "High Resolution Maps from Wide Angle Sonar," *IEEE International Conference on Robotics and Automation*, 1985.

[J. Hertz, et al., 1991] J. Hertz, et al., 1991 *Introduction to the Theory of Neural Computation*, vol. 1: Addison-Wesley Publishing Company.