Experience Mapping: Producing Spatially Continuous Environment Representations using RatSLAM

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

Abstract

RatSLAM is a system for vision-based Simultaneous Localisation and Mapping (SLAM) inspired by models of the rodent hippocampus. The system can produce stable representations of large complex environments during robot experiments in both indoor and outdoor environments. These representations are both topological and metric in nature, and can involve multiple representations of the same place as well as discontinuities. In this paper we describe a new technique known as *experience mapping* that can be used on-line with the RatSLAM system to produce world representations known as *experience maps*. These maps group together multiple place representations and are spatially continuous. A number of experiments have been conducted in simulation and a real world office environment. These experiments demonstrate the high degree to which experience maps are representative of the spatial arrangement of the environment.

1 Introduction

In previous work we have shown a system called RatSLAM performing on-line Simultaneous Localisation and Mapping (SLAM) in a range of unmodified indoor [Milford *et al.*, 2004] and outdoor [Prasser *et al.*, 2005] environments using only vision and wheel encoder information. More recently we have also demonstrated a new module known as *goal memory* that can use the distributed world representations RatSLAM learns to perform efficient goal orientated navigation. Results were presented showing the system performing vision based SLAM on-line and immediately using its learnt representations to navigate directly to designated goal locations [Milford *et al.*, 2005].[1]

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

In this paper we describe a new technique known as experience mapping that builds on top of the RatSLAM system to create world representations that are spatially *continuous* — local areas of the map accurately portray the Cartesian properties of the mapped area in the environment. This technique has several fundamental advantages over the original RatSLAM representations. As environments become larger and more complex, the experience map remains continuous, whilst the RatSLAM representation that underlies it becomes increasingly topological and discontinuous. Because it is made up of experiences that incorporate both pose and visual information, the likelihood of hash collisions is drastically reduced. Lastly, the experience map gathers multiple representations of the same places into overlapping areas in the map.

Section 2 discusses the current SLAM methodologies and highlights the weaknesses in the field. In Section 3 we describe the RatSLAM structure and dynamics and detail some of the limitations that were highlighted by attempts to move to longer experiments in larger, more complex environments. Section 4 describes the implementation of the experience mapping technique. Section 5 describes the setup of the simulation experiments and presents the results. In Section 6 the simulation results are validated by results from a real world experiment. Discussion of the results and implications follows in Section 7 before the paper is concluded in Section 8.

2 Background

Simultaneous Localisation and Mapping is one of the most challenging problems in mobile robotics. Over the last 10 years the exponential increase in computational power available to roboticists has allowed theoretical solutions to the problem to finally be implemented in large scale real world experiments. Impressive results have been generated using a number of techniques, such as Kalman Filters and Expectation Maximisation approaches. However, even with the computer capabilities of today, compromises are invariably made in one or

Figure 1: High resolution occupancy map of the University of Freiburg Campus [Grisetti *et al.*, 2005].

more aspects of these methods. For instance, in order to map a large outdoor environment, some methods resort to computationally intensive off-line procedures which occur after the actual experiment. Alternative techniques incorporate assumptions about the environment that greatly simplify the mathematics and reduce the computational processing required to solve the SLAM problem.

Furthermore a large fraction of the most impressive results in the field have been generated from techniques requiring high fidelity sensors, most commonly laser range scanners. From a mathematical perspective these sensors are ideal because they provide plenty of high precision data. But when one looks at animals and humans, it is quite clear that there are, "other ways to do it." The use of laser based techniques also has a significant consequence — the data processing requirements for the high resolution maps generated (Figure 1) is significant and often forces the computation off-line. Landmark based techniques avoid some of this computational problem but require either environmental assumptions about the types of landmarks or a very accurate and robust general feature recognition system.

One of the biggest problems with many solutions to the SLAM problem is the absence of any well integrated task performance system. After all, the reason living things explore and map their environments is so that they can efficiently move around in it, whether an ant or a human. There has been some limited presentation in the literature of maps created by a SLAM process being used to help a robot navigate to goals, such as the museum robot Minerva. However most of these results have been generated from almost decoupled systems — one to generate a high resolution occupancy map, and one to process the occupancy map and generate a route to the goal. Apart from relying on a high fidelity map, these approaches also bear little plausibility with respect to what happens in nature. Animals are not equipped with high resolution range sensors, and generally navigate using a varying mixture of visual

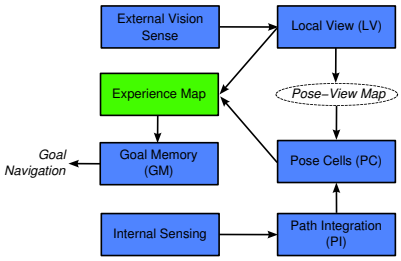


Figure 2: RatSLAM structure

and idiothetic cues. More biologically plausible goal navigation techniques have been developed and tested [Gaussier and Zrehen, 1994], but these are generally created without the focus of robust performance in large scale complex environments and do not achieve comparable results to the mathematical methods.

3 RatSLAM

The RatSLAM system is based on a system of neural networks inspired by the rodent hippocampal complex. These neural networks learn a distributed representation of the environment. In this section we start by describing the system architecture, before moving on to discuss the limitations that spurred the development of the experience mapping technique.

3.1 Overall System

The architecture of the RatSLAM system is shown in Figure 2. 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 representation that is associated with the currently active pose cells. The association between local view and pose is stored in weighted connections between the respective neural networks. A familiar visual scene causes activity to be injected into the particular pose cells linked to the corresponding local view, enabling the robot to re-localise based on familiar visual input.

3.2 Pose Cells

The pose cells are implemented as a competitive attractor network, a neural network that converges to a stable pattern of activity across its inputs. Each unit in the network excites units close to itself and inhibits 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 the point of injection. Activity injected far away from the activity packet will create another packet that competes with the original. If enough activity is injected, the new packet can 'win'

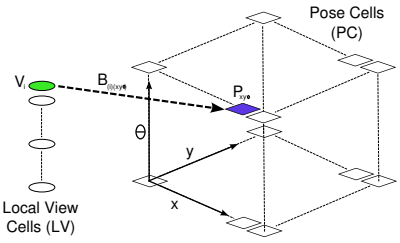


Figure 3: The pose cell and local view cell structures. Associative links are learnt between an activated local view cell V_i and an activated pose cell $P_{xy\theta}$. Active local view cells inject energy into the pose cells through these links.

and the old packet will disappear. The pose cells are arranged in an (x, y, θ) pattern, allowing the robot to simultaneously represent multiple pose estimates in x, y and θ (Figure 3). For indoor experiments each pose cell initially represents approximately $0.25m \times 0.25m$ in area and 10 degrees in bearing.

3.3 Path Integration

The path integration process updates the pose activity packet(s) by injecting activity into the pose cell network based on robot wheel encoder velocities. The process need not necessarily be Cartesian, although the implementation described here is roughly Cartesian to aid in visualization and weight assignment. Initially movement of energy in the pose cell matrix is influenced entirely by the path integration process, but as the robot learns more of the environment's visual appearance, the local view system starts to become more dominant.

3.4 Local View

The local view module is a collection of neural units that represent what the robot sees through the on-board camera in a form that can be associated with the pose cell representation. Camera output is processed through a sum of absolute differences view matcher, that classifies scenes as new or already seen. The local view structure consists of a one-dimensional array of neural units, each of which responds to a different visual scene. Exploration of large indoor environments typically recruits a few thousand neural units responding to distinct views. The details of the vision system architecture are described at length in an earlier paper [Prasser *et al.*, 2004].

The local view units are associated with the pose cell units using Hebbian learning; weights between concurrently active units are increased in strength. As the weights develop, they inject activity from the local view units into the corresponding pose cell units, so that the robot can recall its pose based on familiar visual input. There are no distinct learning and recall phases — they both occur continuously and concurrently.

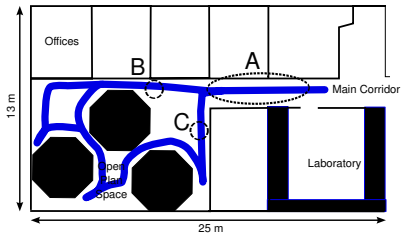
3.5 Spatial Continuity

Our recent work has focused on extensive testing to evaluate the suitability of the RatSLAM system for long term (on the order of days) experiments in larger, more complex environments. This testing has revealed that the original goal memory system, whilst robust to small spatial discontinuities, relied on the RatSLAM environment representation being approximately Cartesian. As test duration and environment complexity increased, the world representations became increasingly topological and less Cartesian in nature. Large topological re-localising 'snaps' in the world representation resulted in failure of the goal memory system.

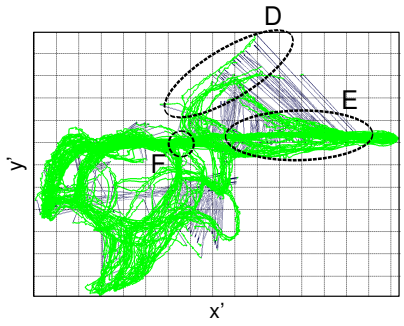
As a robot explores an environment, path integration and visual input drives pose cells to become associated with places and orientations in the environment. Path integration drives local movement of energy around the pose cell matrix. However visual input can cause global re-localisation snaps between cells a long way apart in the pose cell matrix, but which are associated with places close together in the real world. Figure 4(a) shows the testing environment for a 2.5 hour experiment, with the path the robot's exploration algorithm used through the environment shown by the thick solid line. Travelling at an average speed of around 300 mm / second, 2.5 hours was enough time for the robot to thoroughly explore every possible route through the environment.

Figure 4(b) shows the path that the dominant energy packet traced through the pose cell structure during this experiment. The figure is a 2D projection of the 3D pose cell matrix — movement in the third dimension (θ) is not shown in this diagram. The trajectory is shown by the thicker solid lines, with re-localisation snaps shown as thinner straight lines. At the right of the diagram a number of re-localising snaps can be seen where, driven by visual input, the peak energy location in the pose cell network snapped from one location to another. The physical reality represented by these snaps however was a small amount of movement along the main corridor. There is a clear discrepancy between the distance represented in the pose cells and the distance in the physical real world. The same problem exists with orientation — there is a significant orientation difference in the pose cell network before and after the snaps, yet in the physical world the robot had travelled in a straight line, with no orientation change.

The vast majority of conventional SLAM techniques solve this problem by forcing a single representation of the environment. However there is evidence in the biological literature that rats do not necessarily form one coherent representation of the world, especially for complex, visually ambiguous environments [Skaggs and McNaughton, 1998]. Experiments have suggested the possibility of rats switching between multiple represen-



(a) Robot's actual path through the office environment.



(b) Path of maximum energy peak through the (x', y') plane of the pose cell matrix. Each grid square is 4 by 4 pose cells.

Figure 4: Office floor environment.

tations of the same environment as they move around, although the basis for the switching is unclear.

3.6 Multiple Representations

One of the phenomena that can occur in RatSLAM involves more than one group of pose cells, which are from separate parts of the pose cell matrix, representing the same physical location in the environment. This is apparent when looking at the representation of the main corridor by the pose cells in Figure 4(b). The circled areas D and E show straight paths in *different* areas of the pose cells which represent travel along the same physical corridor, labelled A in Figure 4(a). The large number of re-localising snaps shows frequent switching between alternate representations of the corridor.

For most SLAM systems this would be an unacceptable phenomenon and would cause catastrophic failure when the representation was used to perform goal navigation. The initial goal memory implementation described in [Milford *et al.*, 2005] broke down in complex environments because it was based in the pose cell network co-ordinate space and could not deal with large jumps around the pose cell matrix. Selecting a goal location was also difficult for a human user, for in order to specify a goal location the user had to mark *all* the representations of that goal location in the pose cell matrix. Evidence that rodents can form multiple representations of their environment [Skaggs and McNaughton, 1998] suggested that the goal memory implementation rather than the pose cell world representation was at fault. Unfortunately, unlike research on place and head direction cells [Taube *et al.*, 1990; Arleo *et al.*, 2001], there has been little light shed on the exact mechanism rats use to navigate to goals.

3.7 Hash Collisions

The reverse phenomenon to multiple representations can also occur — multiple physical places can be associated with the same pose cells. At point F in Figure 4(b) the same area of pose cells represents two distinct places in the environment, labelled B and C in Figure 4(a). This phenomenon is similar to that of hash collisions when using a hash table. This means that when the pose cells in that area of the matrix are active, there is ambiguity regarding the corresponding robot pose in the physical world.

4 Experience Mapping

The experience mapping technique was developed as a solution to these problems. It solves or avoids the spatial continuity, hash collision and multiple representation issues and produces human-friendly world representations; a human can easily associate parts of the map with places in the physical world. Experience mapping is implemented in a module that builds on top of the world representations in the pose cells and local view cells. This method allows the production of *spatially continuous* representations, which in any local area closely represent the Cartesian arrangement of the environment. The method involves continuous correction of this experience map based on information about the local relative position and orientation of connected experiences.

4.1 Experience Generation

To produce a spatially continuous world representation, output from the pose cells and local view cells is used to create a map made up of robot experiences. A robot experience is a snapshot of the robot's state at a point in time, containing information about the pose and visual scene associated with the experience, as well as odometric information about transitions between it and other connected experiences.

Each experience's relationship to the pose cell and local view networks can be described by 4 state variables:

$$(x', y', \theta', V) \tag{1}$$

where *x', y',* and θ' are the three state variables that describe the location of the cells within the pose cell matrix associated with the experience, and V describes the visual scene associated with the experience. Figure 5 shows the conceptual representation of an experience.

Each experience has a *zone of influence*. This zone of influence describes the value range within each of the 4 state variables when this experience is activated. In the *(x', y')* pose cell plane this zone describes a circular area — the experience has the highest energy level when the maximally activated pose cell is at the centre of this area. This relationship is described by Equations 2, 3, and 4:

$$r = \sqrt{(x'_{pc} - x'_i)^2 + (y'_{pc} - y'_i)^2} \tag{2}$$

$$r_{ratio} = \frac{r}{r_{influence}} \tag{3}$$

$$E_{x'y'} = \max(1.0 - r_{ratio}, 0) \tag{4}$$

where r is the distance in the (x', y') plane of the pose cells from the point of maximal energy to the position associated with the experience, (x'_{pc}, y'_{pc}) are the co-ordinates in the pose cell *(x', y')* plane of the point of maximal energy, (x'_i, y'_i) are the co-ordinates in the *(x', y')* plane of the cells associated with the experience, and E_{xy} is the energy component contribution.

In the θ' dimension, the zone of influence describes a range of orientations within the pose cells that the experience is associated with. This zone is centered around the experience's preferred orientation — it is most energized when the maximally activated pose cell's orientation matches this preferred orientation. As the peak pose cell rotates away from the experience's preferred orientation, the energy level of the experience will decrease. Equations 5, 6, and 7 describe the angular zone of influence:

$$\Delta\theta' = |\theta'_{pc} - \theta'_i| \tag{5}$$

$$\theta'_{ratio} = \frac{\Delta\theta'}{\theta_{influence}} \tag{6}$$

$$E_{\theta'} = \max(1.0 - \theta'_{ratio}, 0) \tag{7}$$

where $\Delta\theta'$ is the minimum angular difference between the most likely pose cell orientation θ'_{pc} and the pose cell orientation θ'_i associated with the experience, and $E_{\theta'}$ is the angular energy contribution.

For the V variable which describes the visual scene, the zone of influence is discrete — each experience's zone includes exactly one visual scene. The visual scene acts like a switch for the experience, switching it on or off.

$$E_V = \begin{cases} 0 & \text{if} \quad V \neq V_i; \\ 1 & \text{if} \quad V = V_i; \end{cases} \tag{8}$$

where V is the current visual scene, V_i is the visual scene associate with experience i, and E_V is the visual scene energy component.

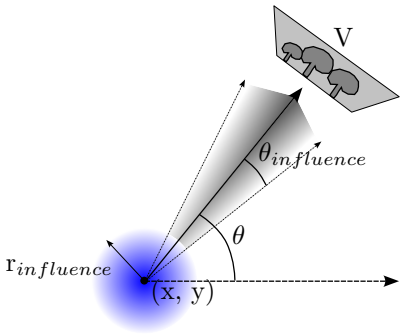


Figure 5: Basic experience

Together Equations 4, 7, and 8 give the total energy level of an experience, E_i:

$$E_i = E_V \times (E_{xy} + E_\theta) \tag{9}$$

As the robot moves around a novel environment, it needs to generate experiences to form a representation of the world. Learning of new experiences is triggered not only by exploring in new areas of an environment, but also by visual changes in areas the robot has already explored. In addition, if there are multiple paths through the pose cells that represent traversing the same corridor, the system will learn multiple sets of experiences, each set associated with one of the paths through the pose cells. This by itself would lead to the goal memory problems dealing with multiple representations, however this problem is avoided through map correction, which is discussed in Section 4.3.

RatSLAM continuously checks whether given the current visual scene and pose cell activity, it is in a previously *experienced* place. If none of the current experiences have a positive energy level, a new experience is learnt, as shown in Equations 10 and 11.

$$E_{max} = \max(\mathbf{E}) \tag{10}$$

where $\mathbf{E}$ is the vector of experience energy levels, and E_{max} is the maximum experience energy level.

$$N_e = \begin{cases} N_e & \text{if} \quad E_{max} > 0; \\ N_e + 1 & \text{if} \quad E_{max} \leqslant 0; \end{cases} \tag{11}$$

where N_e is the number of experiences.

4.2 Experience Transitions

As the robot moves around the environment, the system also learns experience *transitions*. These transitions represent the *physical* movement of the robot in the world as it moves from one experience to another. A transition from one experience to another is represented by three variables θ_{ij}, ϕ_{ij}, and d_{ij}, as shown in Figure 6.

Two angular variables are required to fully describe an experience transition. θ_{ij} describes the difference

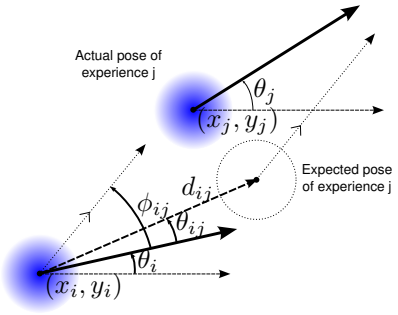


Figure 6: An experience transition. Shaded circles and bold arrows show the actual pose of the experiences. The dotted circle and arrow shows the expected pose of experience j based on the transition information.

between the absolute orientation of the first experience θ_i and the angle to the location of the second experience (x_j, y_j) within the experience map co-ordinate space. ϕ_{ij} however describes the angular difference between the orientation of the two experiences, without any regards to their relative (x, y) location. d_{ij} describes the translational distance between experiences i and j.

When there is a large re-localising snap in the pose cell matrix, the experience transition preserves information about the physical movement of the robot through that snap. This information is used to perform map correction that helps generate a spatially continuous map, as described further in Section 4.3.

4.3 On-line Map Correction

By introducing the visual scene information as a fourth state variable of the experiences we avoid the hash collision problem described in Section 3.7. However, there is still the problem of multiple representations of the same physical location in the environment. This becomes quite apparent when closing the loop — any SLAM system that takes into account ambiguity in the environment cannot instantaneously re-localise at the moment it starts on the second loop of an environment. Rather, it takes some time before enough sensory input is received to confirm that it is back at the start of the second loop. Conventional SLAM systems using EM techniques then do a global map correction that effectively coalesces the multiple representations of the start of the loop into a single one.

Rather than attempting to combine multiple representations, the experience mapping technique *overlaps* them within the experience map's own *(x, y, θ)* co-ordinate space through a process of map correction (not to be confused with the *(x', y', θ')* co-ordinates that describe the locations of the pose cells associated with the experience). This preserves the multiple representations and the information they possess about the environment but creates a human-friendly map where these representations of the same physical place overlap.

The first experience's pose within the experience map co-ordinate space is initialised to an arbitrary location and orientation. The pose of each subsequent experience within this experience map co-ordinate space is initialized using the pose of the last activated experience and the relative odometric information about the transition. When re-localising after significant time spent in a novel part of the environment, the robot will snap from the new experience it has most recently learned to an old, familiar experience. This will result in a large discrepancy between the transition's relative odometric information and the difference between the two experiences *(x, y, θ)* co-ordinates.

The map can be made to be spatially continuous by minimizing these discrepancies. This can be achieved by consolidating the expected pose of each experience based on relative odometric information and their current pose. Equations 12, 13, and 14 describe the update of the physical pose state of each experience.

$$\Delta\theta_i = \alpha_\theta \left[\sum_{j=1}^{N_{from}} \left(\theta_j - \theta_i - \phi_{ij}\right) + \sum_{k=1}^{N_{to}} \left(\theta_k + \phi_{ki} - \theta_i\right) \right] \quad (12)$$

where $\Delta\theta_i$ is the overall change in orientation of experience i, α_θ is the angular learning rate, N_{from} is the number of links from experience i to other experiences, θ_j is the angle of the linked to experience j, θ_i is the current absolute angular orientation of experience i, ϕ_{ij} is the expected angular rotation from experience i to experience j, and N_{to} is the number of links from other experiences to experience i.

$$\begin{aligned} \Delta x_i = \alpha_d \Bigg[& \sum_{j=1}^{N_{from}} \left(x_j - d_{ij}\cos(\theta_i + \theta_{ij}) - x_i\right) \\ & + \sum_{k=1}^{N_{to}} \left(x_k + d_{ki}\cos(\theta_k + \theta_{ki}) - x_i\right) \Bigg] \end{aligned} \quad (13)$$

where Δx_i is the overall change in x co-ordinate of experience i, α_d is the translational learning rate, x_j is the x co-ordinate value of the linked to experience j, d_{ij} is the odometric translational distance between experience i and j, θ_{ij} is the relative angular *orientation* from experience i to j, and x_i is the current x co-ordinate value of experience i.

Table 1: Experience map learning rates

Learning Rate Variable	Value	Description
α_θ	0.5	Orientation update
α_d	0.5	Translational update

$$\Delta y_i = \alpha_d \Bigg[\sum_{j=1}^{N_{from}} \big(y_j - d_{ij} \sin(\theta_i + \theta_{ij}) - y_i\big) + \\ + \sum_{k=1}^{N_{to}} \big(y_k + d_{ki} \sin(\theta_k + \theta_{ki}) - y_k\big) \Bigg] \quad (14)$$

where the naming convention is as for Equation 13, with sin instead of cos.

4.4 Map Learning Rates and Computability

The experience map is subject to the same constraints of any network style learning system — appropriate learning rates must be used to balance rapid convergence with instability. The nature of the network is such that the vast majority of inter-experience links have a high degree of consistency between their relative odometric information and their (x, y, θ) state, with only significant discrepancies whenever there is a re-localising snap. This allows a high learning rate to be used when compared with a self organizing network that has *all* its values randomly initialized. Table 1 shows the learning rates used to update each experience's pose state. Larger learning rates were found to result in network instability.

Experience map computation scales directly with the number of experiences in the map. To achieve appropriate coverage of an environment the number of experiences must be approximately proportional to the area of the environment. The bulk of the computational resources is shared by the experience mapping technique and the pose cell matrix, which computationally also scales directly with the area of the environment. On a 1.1 GHz laptop RatSLAM mapped at real time speed a 43 metre by 13 metre building floor.

5 Simulation Experiments

Simulation allows us to perform a large number of experiments quickly in complex environments which are not necessarily feasible to set up in reality. In Section 6 we describe a real world experiment which validates the results obtained in simulation.

5.1 Simulation Description

We used a simulator developed in our laboratory that can load simple object maps of environments designed in a world editor. Lengths and orientations of walls are specified. Noisy, ambiguous visual information is created using an algorithm that creates visual scene numbers based on the simulated robot's pose. False positive matches can occur — at random times the simulated robot will falsely recognise a familiar scene when in a completely novel part of the environment. Simulations were performed on a 1.1 GHz Pentium III laptop at real time speed. Extensive experimentation in the past both in simulation and the real world has confirmed that the performance in simulation is indicative of real world results.

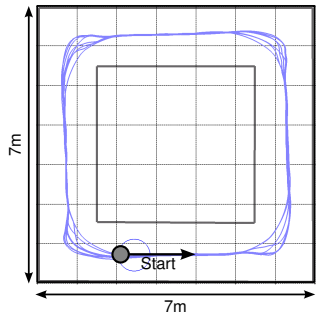
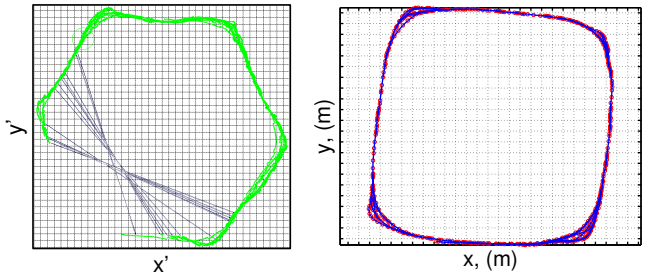


Figure 7: Small loop environment



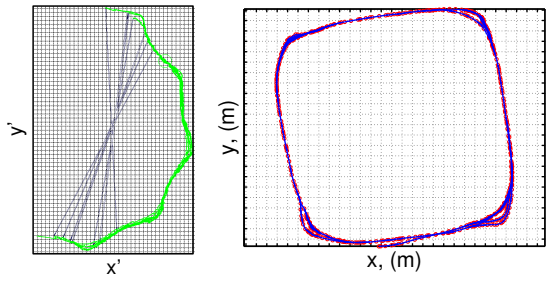
(a) Pose cell map (b) Experience map

Figure 8: Loop closing under a 1.5 deg/sec odometric bias. The grid squares are 250mm by 250mm.

5.2 Closing Simple Loops Under Odometric Error

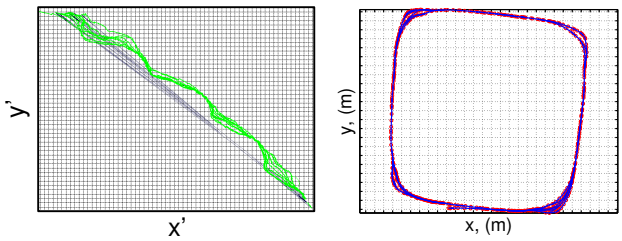
RatSLAM can consistently close loops of varying complexity and length in a range of indoor and outdoor environments. This loop closing is achieved through re-localisation snaps in the pose cell matrix, driven by visual input from familiar scenes. These snaps involve the peak energy location in the pose cell matrix snapping from one location to another. This set of simulations involved 15 minute tests in a square corridor loop environment as shown in Figure 7. The robot's starting location and orientation are shown by a small circle and arrow at the bottom left of the figure.

Since the total loop distance was quite short (about 18



(a) Pose cell map (b) Experience map

Figure 9: Loop closing under a 3.0 deg/sec odometric bias.



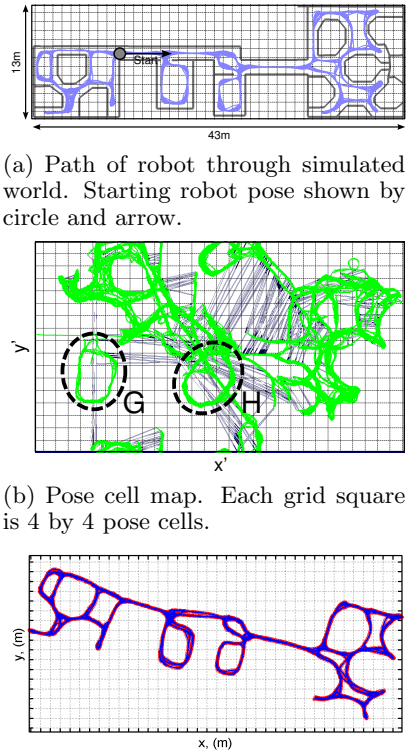
(a) Pose cell map (b) Experience map

Figure 10: Loop closing under a 6.0 deg/sec odometric bias.

metres), there was only a small amount of accumulated odometric error after one loop was completed. In order to demonstrate the spatial continuity of the experience map, a range of odometric biases were introduced. These biases allowed us to show more clearly the difference between the pose cell and experience map world representations.

Figure 8(a) shows the trajectory of the energy peak through the (x', y') co-ordinate space of the pose cell matrix, when a constant odometric bias of 1.5 degrees per second was introduced. The effect of the bias is clear — since movement of energy around the pose cell matrix depends on the path integration module, the 90 degree corners in the real world environment are realised as significantly less than 90 degree turns in the pose cell matrix. Straight paths are also represented as slightly curved routes in the pose cells. There is a region of re-localisation snaps at the beginning of the loop, where visual input has caused the location of peak activity to snap from one location to another. These snaps are represented by the thin straight lines.

In contrast the experience mapping technique has produced a continuous, spatially representative map, as shown in Figure 8(b). The separation of the experiences before and after the re-localisation snap is indicative of the physical movement of the robot rather than the movement of the peak energy packet through the pose cell matrix.



(a) Path of robot through simulated world. Starting robot pose shown by circle and arrow.

(b) Pose cell map. Each grid square is 4 by 4 pose cells.

(c) Experience map. Each grid square represents 1 square metre.

Figure 11: Full floor environment.

Figures 9 and 10 show the pose cell and experience map world representations for odometric biases of 3 deg/sec and 6 deg/sec. The increased path integration error is apparent in the pose cell matrix representations, especially in Figure 10 where the trajectory through the pose cells is almost a straight line. However, the experience maps for all three cases are consistent and representative of the actual physical environment. The only significant variation that occurs is in their overall orientation. This is a result of no experiences being globally anchored — all experiences are free to move and rotate based on the network dynamics, so the final stable map can vary in its position and orientation, but is always internally consistent.

5.3 Full Floor Office Environment

As the environment becomes larger and more complex, the discontinuity of the pose cell trajectories and the frequency of re-localising snaps increases. Figure 11(a) shows the robot's path through a simulated environment based on the complete floor of our office building. This environment contains loops of varying sizes and shapes, as well as long corridor sections and open areas.

The trajectory of the energy peak through the (x', y') co-ordinate space of the pose cells is shown in Figure

Figure 12: Photo of open plan office environment.

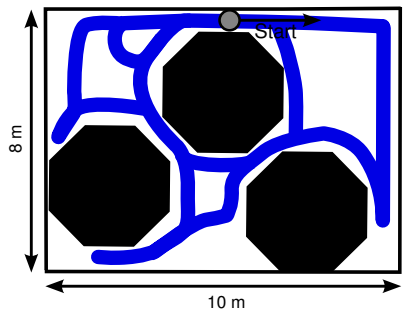
11(b). There is a significant amount of re-localisation present as well as multiple representations of places in the environment, such as the two loops at G and H that represent laps of the same room. The experience map (Figure 11(c)) however closely matches the actual environment. All the multiple representations have been grouped into overlapping areas.

6 Real Robot Experiments

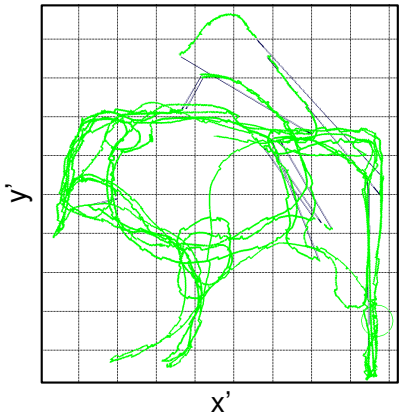
To validate the simulation results we tested the experience mapping technique on a Pioneer 2DXE mobile robot equipped with a 50 degree field of view forward facing camera, a scanning laser and wheel encoders. Some visual processing was performed on the robot's on-board 400 MHz Athlon K6 processor, and the information wirelessly transmitted to a 1.1 GHz Pentium III laptop where the rest of the RatSLAM system resides, including the experience mapping component. System iteration and information updating were performed at an average speed of 7 Hz.

The testing environment for the experiment was an $80m^2$ area of open plan office space in our building, as shown in Figures 12 and 13(a). The environment consisted of a number of desk cloisters, chairs, fridges, office doors and other objects. The robot was set loose for 15 minutes in this environment using its own exploration algorithm, which attempts to learn all the possible routes through the environment.

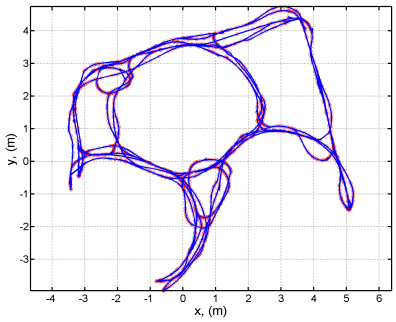
Figure 13(b) shows the pose cell trajectory for the experiment. There are many re-localising snaps and multiple representations present. Figure 13(c) shows the corresponding experience map for this environment. The re-localising snaps are gone, and the multiple representations have been grouped into overlapping areas in the map. The map is human-friendly — a person can easily connect places in the map to places in the physical environment.



(a) Path of robot through open plan office space.



(b) Pose cell map. Each grid square is 4 by 4 pose cells.



(c) Experience map. Each grid square represents 1 square metre.

Figure 13: Open plan environment

7 Discussion

Local areas of the experience maps preserve the Cartesian properties of the area of the environment they represent. This correlation is preserved through minimisation of the discrepancies between the relative poses of the experiences and the odometric information about transitions between the experiences. This means that there is no guarantee that on a global scale the map will be completely Cartesian — for instance straight corridors may be slightly curved in the experience map. In certain situations, for example when the robot is in a very long narrow corridor, the experience map of the forward and backward paths may not overlap. As external observers we know that it is the same corridor going either way, but from the perspective of a robot with a limited field of view and appearance based SLAM technique, it has no information linking the two directions apart from odometric information. Since a

long time may pass between traversing the corridor one way and returning back the other way, the experience map representations of both directions may not overlap exactly.

One obvious solution to this problem is for the exploration algorithm to encourage 'turn arounds' in long narrow environments. By turning around and travelling back the other way, a set of odometric transition links are learnt between the experiences representing either direction of travel along the corridor. The map correction algorithm then overlaps the two representations since it now has odometric information linking them. We are currently working on behavioural based solutions to this problem.

The visual component of an experience is dependent on the lighting conditions of the environment. The indoor environments tested in have so far had relatively constant illumination. However, in experiments on an outdoor robot tractor we have tested a histogram based vision system which relies on only the hue and saturation components of HSV colour space, thereby introducing some robustness to light intensity variations over time [Prasser *et al.*, 2005].

Future work will investigate a goal memory system based on the experience map. Spatially continuous maps are ideal for goal directed navigation, because in any local area the map closely represents the geometric arrangement of the environment. Since the map groups together multiple representations, goals can be specified by a single mouse click rather than by manually picking all the alternate representations in the pose cell matrix. By incorporating time durations into the inter-experience link transitions, the quickest route to any location in the map can be found. The experience map provides both the shortest continuous global route to a goal as well as locally accurate spatial representations, which can be used to make decisions at multiple-choice locations in the environment.

8 Conclusion

This paper has described the methodology and integration of a new technique known as experience mapping into the RatSLAM system. Results were presented for a range of simulated environments and a medium scale real world environment. These results demonstrate that the experience mapping technique can process the world representations generated by RatSLAM to create on-line a map that is spatially continuous and that groups multiple representations into overlapping areas. The continuity and local Cartesian properties of the experience maps ensure they are human-friendly and are suitable for goal directed navigation, regardless of the topological nature and discontinuity of the underlying RatSLAM world representations.

References

[Arleo *et al.*, 2001] Angelo Arleo, F. Smeraldi, S. Hug, and W Gerstner. Place cells and spatial navigation based on vision, path integration, and reinforcement learning. *Advances in Neural Information Processing Systems*, 13, 2001.

[Gaussier and Zrehen, 1994] Philippe Gaussier and Stephane Zrehen. Navigating with an animal brain: A neural network for landmark identification and navigation. *Intelligent Vehicles*, pages 399–404, 1994.

[Grisetti *et al.*, 2005] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. Improving grid based SLAM with Rao Blackwellized particle filters by adaptive proposals and selective resampling. In *International Conference on Robotics and Automation*, Barcelona, Spain, 2005.

[Milford *et al.*, 2004] M.J. Milford, Gordon Wyeth, and David Prasser. Simultaneous localization and mapping from natural landmarks using ratslam. In *Australasian Conference on Robotics and Automation*, Canberra, Australia, 2004.

[Milford *et al.*, 2005] M.J. Milford, Gordon Wyeth, and David Prasser. Efficient goal directed navigation using RatSLAM. In *International Conference on Robotics and Automation*, Barcelona, Spain, 2005.

[Prasser *et al.*, 2004] David Prasser, Gordon Wyeth, and M.J. Milford. Biologically inspired visual landmark processing for simultaneous localization and mapping. In *International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004.

[Prasser *et al.*, 2005] D. Prasser, Michael Milford, and Gordon Wyeth. Outdoor simultaneous localisation and mapping using RatSLAM. In *International Conference on Field and Service Robotics*, Port Douglas, Australia, 2005.

[Skaggs and McNaughton, 1998] WE Skaggs and BL McNaughton. Spatial firing properties of hippocampal CA1 populations in an environment containing two visually identical regions. *Journal of Neuroscience*, 18:8455–8466, 1998.

[Taube *et al.*, 1990] J.S. Taube, R.U. Muller, and J.B. Ranck, Jr. Head direction cells recorded from the postsubiculum in freely moving rats. I. Description and quantitative analysis. *Journal of Neuroscience*, 10:420–435, 1990.