Towards Lifelong Navigation and Mapping in an Office Environment

Gordon Wyeth[1] and Michael Milford[1,2]

[1] Information Technology and Electrical Engineering, University of Queensland, Australia
wyeth@itee.uq.edu.au, milford@itee.uq.edu.au

[2] Queensland Brain Institute, University of Queensland, Australia

Summary. This paper addresses the challenge of developing robots that map and navigate autonomously in real world, dynamic environments throughout the robot's entire lifetime – the problem of lifelong navigation. Static mapping algorithms can produce highly accurate maps, but have found few applications in real environments that are in constant flux. Environments change in many ways: both rapidly and gradually, transiently and permanently, geometrically and in appearance. This paper demonstrates a biologically inspired navigation algorithm, RatSLAM, that uses principles found in rodent neural circuits. The algorithm is demonstrated in an office delivery challenge where the robot was required to perform mock deliveries to goal locations in two different buildings. The robot successfully completed 1177 out of 1178 navigation trials over 37 hours of around the clock operation spread over 11 days.

1 Introduction

A mobile robot can achieve its goals more efficiently if it can remember the layout of its surroundings, maintain a representation of its own position, and use its memory of the spatial layout to execute paths to goal locations. The problem of getting a robot to build a spatial representation of its world and to localize within that spatial representation has been extensively studied as the problem of Simultaneous Localization and Mapping (SLAM). There are several solutions to the SLAM problem, and many demonstrations of highly accurate mapping ability (such as [4, 13]), yet it is widely recognized that these solutions are not finding their way into practical applications with autonomous robots [14]. The existing solutions are geared towards building maps based on static features, and assume that the mapped features of the surroundings will not change. SLAM maps are not easily updated, and rapidly become out of date as alterations to the robot's surroundings accrue. In visual SLAM, where the robots use cameras to detect features, the problem is exacerbated by constant visual changes caused by variations in lighting. For a map to be useful to a robot, the map must adapt to all kinds of change: rapid and gradual, transient and permanent, visual and functional.

Animals, on the other hand, seem to adapt to changes in spatial layout with ease. The navigation ability of the rodent has been widely studied in biology at both a behavioral and neural level [3]. Rats can forage over ranges of kilometers, remembering sources of food, avoiding dangerous areas, and returning to the nest [5]. The rat navigates equally well through urban back alleys, under houses, through pipes, and across grasslands. All of these environments are constantly changing from the activities of other animals or humans, and from the weather and the season. Biologists inspired by the remarkable navigation performance of the rodent family have made major advances in unraveling the neural circuitry that stores the map, localizes the rat and plans the paths. In this paper, we use that neural circuitry as inspiration for a solution to the problem of lifelong mapping and navigation for a mobile robot.

There have been a number of robot systems built based on the neural circuitry of the rat: some have been designed to test biological ideas (for example, [1]), others have explored improvements for robotic applications (for example, [2]). Our previous work has been focused on the application of the rodent inspired algorithm, RatSLAM, to challenging robot problems. RatSLAM has been demonstrated mapping indoor environments [9], a university campus [11], and an entire suburb [10]. In this paper, we show that the maps built by RatSLAM are well suited to planning for navigation, and readily maintainable in changing environments.

The challenge we set for RatSLAM in this paper was to build a map suitable to plan "deliveries" in two working office environments operating at all times in the 24 hour day/night cycle over a two week period. The robot was given an initial period to autonomously explore the first environment and generate a user readable map. The delivery locations were then marked on that map, and delivery requests randomly generated for over 1000 trials. The robot was then moved (without notice) to a second building where it built a new map, and performed deliveries to new locations. The robot was finally returned to the original building where it autonomously re-localized and performed deliveries to location in the original map. While performing deliveries in both buildings, the robot autonomously found and remembered the location of its charger, and autonomously recharged its batteries as required.

The paper describes the RatSLAM system in the next section, detailing how the system can be used to build maps that remain stable in size and computation requirements over time. After describing the delivery challenge in detail, the paper then shows results that illustrate both reliability and stability of the mapping and navigation system. A more detailed and authoritative version of this study can be found in [8].

2 RatSLAM

RatSLAM has three principal components: the *pose cells* which use odometry to provide a locally consistent spatial reference, the *local view cells* which provide

the interface to the robot's external sensors, and the *experience map* which fuses the information in the pose cells and the local view cells to provide a representation suitable for autonomous navigation. The components of RatSLAM and the interactions of the components are illustrated in Figure 1, and described briefly in the following sections. Further details of the operation of RatSLAM can be found in [8, 10].

2.1 Pose Cells

The pose cells are a three-dimensional Continuous Attractor Network (CAN), a type of neural network that consists of an array of neural units [12]. Unlike other neural networks that operate by changing the value of connections between neural units, the CAN predominantly operates by varying the activity of the neural units while keeping the connection strengths fixed. During operation, the pose cell network will generally have clusters of highly active units: *activity packets*. The active cells of an activity packet provide a representation of the robot's pose that is consistent with the pose cell network's rectangular prism structure, as shown in Figure 1. Each of the three dimensions of the prism corresponds to one of the three spatial dimensions x', y', and θ'. Primed co-ordinates are used as the pose cells' representation of space is heavily distorted and aliased. To interpret the pose of the robot from the pose cells, the activity packet(s) must be transformed into the more useful experience map representation. The purpose of the pose cells and the activity packet is to provide a representation that is readily associated with external perception through the local view.

Attractor Dynamics

An activity packet is self-maintained by fixed local excitatory connections that increase the activity of units that are close in (x', y', θ') space to an active unit. Fixed inhibitory connections suppress the activity of smaller clusters of activity elsewhere in the network. Connections wrap across all six faces of the pose cell network, as shown by the longer red arrows in Figure 1. The change in the cells' activity level ΔP is given by:

$$\Delta P = P * \varepsilon - \varphi \tag{1}$$

where P is the activity matrix of the network, ε is the connection matrix, and * is the convolution operator. As well as the inhibition in the connection matrix, the constant φ creates further global inhibition. At each time step, activation levels in P are restricted to non-negative values and the total activation is normalized.

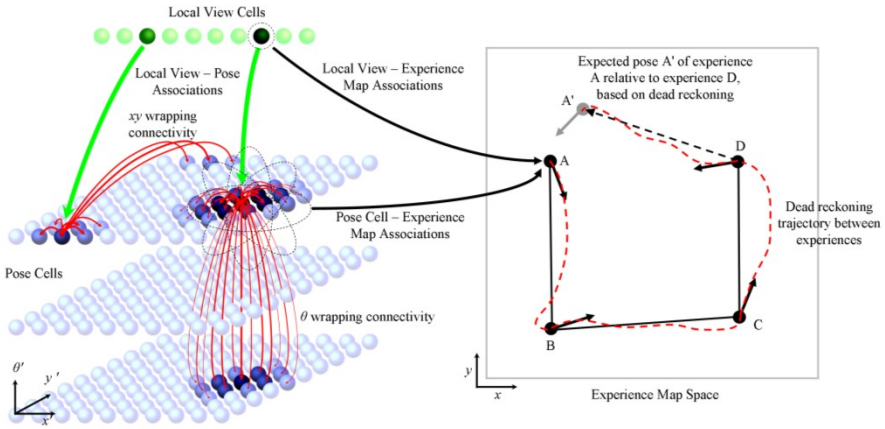

Fig. 1. The RatSLAM system. Each local view cell is associated with a distinct visual scene in the environment, and becomes active when the robot sees that scene. A three-dimensional continuous attractor network forms the pose cells, where active pose cells encode the estimate of the robot's pose. Each pose cell is connected to proximal cells by excitatory and inhibitory connections, with wrapping across all six faces of network. Intermediate layers in the (x', y') plane are not shown. The network connectivity leads to clusters of active cells known as activity packets. Active local view and pose cells drive the creation of experience nodes in the experience map, a semi-metric graphical representation of places in the environment and their interconnectivity.

Path Integration

Path integration involves shifting the activity packet in the pose cell network based on odometry information. At each time step, RatSLAM interprets the odometry information to displace a copy of the current activity state in the pose cell network. The path integration process can cause a cluster of activity in the pose cells to shift off one face of the pose cell structure and wrap around to the other, as is shown in both packets in Figure 1, one of which is wrapping across the θ' boundary, the other across the y' boundary. Recording from a single cell under path integration will create firing fields with rectangular tessellations, similar to the triangular tessellations seen in rodent's grid cells found in entorhinal cortex [6].

2.2 Local View Cells

The local view cells produce a sparse vector based on a classification of the robot's external perception. A local view cell is created every time the robot sees a scene that is distinct from any other scene that the robot has seen before. Each local view cell is paired with a template of its associated distinct scene. If the scene is viewed again then the local view cell will become active.

Scene Recognition

In this system, the external perception for the local view system was driven entirely from panoramic images obtained from a camera facing vertically upwards at a parabolic mirror, mounted at the central rotation axis of the robot. Some typical images are shown in Figure 2. The image is unwrapped to 128 pixels representing 360° in the horizontal dimension and 20 pixels representing 90° in the vertical dimension. Gain and exposure control are applied to the entire image, with patch normalization on the lower half of the image. In order to provide rotationally invariant matching, each row in the image is transformed to a set of Fourier coefficients. Image similarities between the current image and template images are calculated using the multiplication of the Fourier coefficients, which is equivalent to convolution in image space:

$$C = F^{-1}\left[\sum_{y=1}^{h} F\left(\vec{i}_y\right) \cdot F\left(r_y\right)\right] \tag{2}$$

where $F(\ )$ is the Fourier Transform operator and i_y and r_y are the pixels rows at y in the current and template images, respectively. The value of the maximum real correlation coefficient gives the quality of the match m:

$$m = \max\left(\mathrm{Re}\left(C\right)\right) \tag{3}$$

A new image template and local view cell is created if the best match for all current image-template image pairings is below a threshold m_{min}. The match quality scores for each image pair are used to set the activation levels for the corresponding local view cells:

$$V_i = \max(m_i, 0) \quad \forall i \tag{4}$$

Multiple local view cells can be simultaneously active to varying degrees in regions with perceptual aliasing, and there is no competition imposed between the cells. The connections between the local view cells to the pose cells, and the spatio-temporal filtering properties of the pose cells, prevent perceptual aliasing from adversely affecting the representation built in the experience map.

Connecting Local View to Pose

RatSLAM increases the strength of connections between local view cells and pose cells that are active simultaneously. In other words, RatSLAM learns an association between a visual scene and the robot pose. During a loop closure event, the familiar visual scene activates local view cells with learnt connections to the pose cells representing the pose where the visual scene was first encountered. Due to the attractor dynamics of the pose cells, a single visual scene is not enough to force an immediate change of pose; several consecutive and consistent views are required to update the pose. The attractor dynamics temporally and spatially filter

the information from the local view cells, providing rejection of spurious loop closure events.

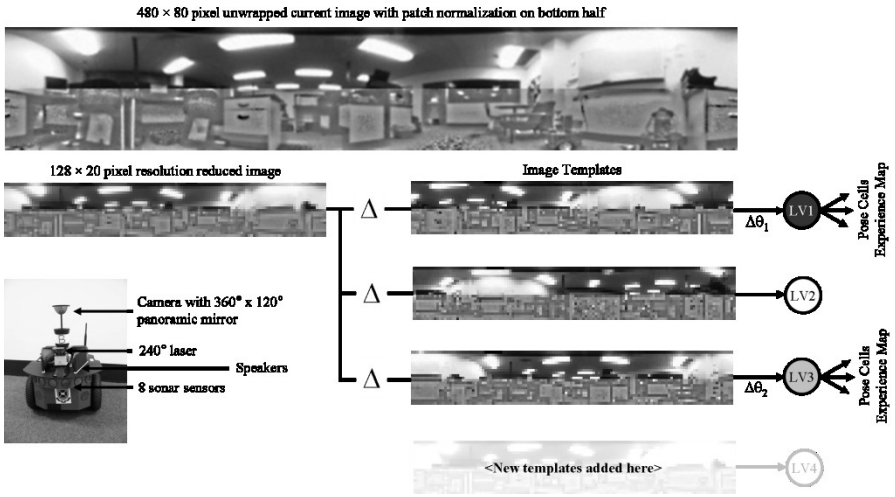

Fig. 2. Vision hardware and vision processing system. A camera and mirror produces panoramas which are unwrapped into 360 degree by 90 degree images. The image is then reduced in resolution, patch normalized to enhance local image contrast and correlated with all template images. Template images that are close matches to the current image activate local view cells, which link to the pose cells and experience map.

The connections between local view cells and pose cells are stored in a connection matrix β, where the connection between local view cell V_i and pose cell $P_{x',y',\theta'}$ is given by:

$$\beta_{i,x',y',\theta'}^{t+1} = \max\left(\beta_{i,x',y',\theta'}^{t}, \lambda V_i P_{x',y',\theta'}\right) \tag{5}$$

where λ is the learning rate. When a familiar visual scene activates a local view cell, the change in pose cell activity, ΔP, is given by:

$$\Delta P_{x',y',\theta'} = \frac{\delta}{n_{act}} \sum_i \beta_{i,x',y',\theta'} V_i \tag{6}$$

where the δ constant determines the influence of visual cues on the robot's pose estimate, normalized by the number of active local view cells n_{act}. Figure 1 represents the moment in time when a strongly active local view cell has injected sufficient activity into the pose cells to cause a shift in the location of the dominant activity packet. The previously dominant activity packet can also be seen, which is less strongly supported by a moderately activated local view cell.

2.3 Experience Mapping

The experience map combines the activity pattern of the pose cells with the activity of the local view cells to create a topologically consistent and semi-metric map. The pose cells' representation of space is distorted and aliased, making it unsuitable for path planning. Often when a loop closure event occurs, the odometric error introduces a discontinuity into the pose cells' representation of space, creating two sets of cells that might represent the same area in space. Similarly, the wrapped connectivity of the pose cell network leads to pose ambiguity, where a pose cell encodes multiple locations in the environment, forming the tessellated firing fields seen in grid cells in the rat. The experience map does not contain the discontinuities and ambiguities of the pose cells.

The experience map contains representations of places combined with views, called experiences, e_i, based on the conjunction of a certain activity state P_i in the pose cells and the local view cells V_i. Links between experiences, l_{ij}, describe the spatio-temporal relationships between places. Each experience is positioned at a location p_i, a position that is constantly updated based on the spatial connectivity constraints imposed by the links. Consequently, the complete state of an experience can be defined as the triple:

$$e_i = \left\{P^i, V^i, \mathbf{p}^i\right\} \tag{7}$$

Figure 1 shows the region of pose cells and the single local view cell associated with the currently active experience A.

Transition links, l_{ij}, encode the change in position, Δp_{ij}, computed directly from odometry, and the elapsed time, Δt_{ij}, since the last experience was active:

$$l^{ij} = \left\{\Delta\mathbf{p}^{ij}, \Delta t^{ij}\right\} \tag{8}$$

where l_{ij} is the link from the previously active experience e_i to the new experience e_j. The temporal information stored in the link provides the travel time between places in the environment. Path planning is achieved by integrating the time values in the transition links starting at the robot's current location to form a temporal map. The fastest path to a goal experience can be computed by performing steepest gradient ascent from the goal experience to the current location.

Creating Experiences and Closing Loops

At each time step, the current pose cell and local view cell states are compared with each experience. If a previously stored experience matches the current state, it is chosen as the 'active' experience, and represents the best estimate of the robot's location within the experience map. If the activity state in the pose cells or local view cells is not sufficiently described by any of the existing experiences, a new experience is created using the current pose and local view cell activity states. The odometry information defines the initial location space of a newly created experience relative to the previous experience:

$$e_j = \{P^j, V^j, \mathbf{p}^i + \Delta\mathbf{p}^{ij}\} \tag{9}$$

When loop closure occurs, the relative position of the two linked experiences in the map will typically not match the odometric transition information between the two, as shown in the discrepancy between experience A and A' in Figure 1. A relaxation method seeks to minimize the discrepancy between odometric transition information and absolute location in experience space, by applying a change in experience location $\Delta\mathbf{p}^i$:

$$\Delta\mathbf{p}^i = \alpha\left[\sum_{j=1}^{N_f}(\mathbf{p}^j - \mathbf{p}^i - \Delta\mathbf{p}^{ij}) + \sum_{k=1}^{N_t}(\mathbf{p}^k - \mathbf{p}^i - \Delta\mathbf{p}^{ki})\right] \tag{10}$$

where α is a correction rate constant, N_f is the number of links from experience e_i to other experiences, and N_t is the number of links from other experiences to experience e_i. Equation 10 is applied iteratively at all times during robot operation; there is no explicit loop closure detection that triggers map correction. The effect of the repeated application of Equation 10 is to move the arrangement of experiences in experience map space incrementally closer to an arrangement that averages out the odometric measurement error around the network of loops.

Experience Pruning

The experience map algorithm will continue to add experiences about the changing state of the world as the robot operates. In this way, the robot constantly updates and maintains its representations of the world. For example, if the robot proceeds along a corridor on one day with an adjoining door open, then the robot will build an experience that captures the open door. If the door is closed on the next day, then a new experience will be constructed with the door closed. The experiences will overlap in the experience map, by virtue of their connectivity to surrounding experiences, effectively mapping the same position with the alternate states of the door. This is not an extension of the existing algorithm; it is a property inherited from the algorithm's biological origins.

The difficulty with using this method of map maintenance is that the robot will remember all experiences from throughout its lifetime, creating an unmanageable list of experiences to search and update. The list of experiences must be pruned to a manageable level in order to attain the goal of lifelong navigation and mapping. Experience pruning consolidates parts of the experience map which exceed a maximum spatial density threshold. In this way the number of experiences grows in proportion to the size of the environment that has been explored, but not with time.

Figure 3 illustrates an experience being pruned from the map. The pruning algorithm uses a grid overlaid on the experience map to tag squares that contain more than one experience. Other methods for choosing the experience to delete were investigated (using measures such as recency and connectivity) but were found to have no effect on performance. Existing transitions to removed expe-

riences are either deleted or updated to link to another experience. The odometric information for the new link is inferred from the current positions of the experiences in experience space, while the temporal information is calculated by dividing the inter-experience distance by the average robot speed. If, after the pruning process, any local view cells no longer define any experiences, the cells and the visual templates associated with them are removed. Any local view – pose cell links from these removed local view cells are also deleted.

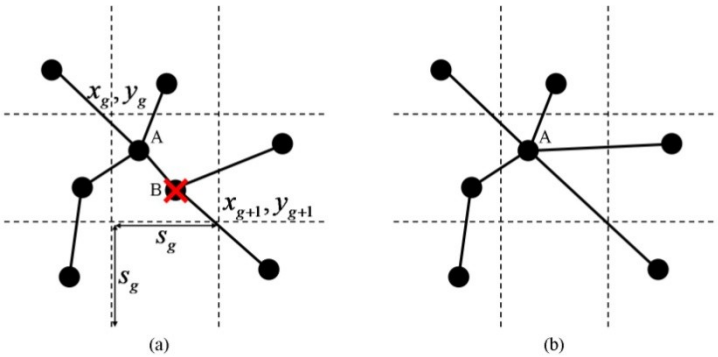

Fig. 3. Experience map pruning. Experiences are removed to maintain a one experience per grid square density. (b) Transitions to or from removed experiences are either deleted or reconnected to remaining experiences.

3 The Office Delivery Challenge

The challenge set for the system was to perform the role of a delivery robot in a real world workplace over a two week period. The workplace consisted of floors in two different buildings at The University of Queensland in Brisbane, Australia, shown in Figure 2. The two buildings, Axon and General Purpose South (GP South), are typical research environments, moderately populated during the day and cleaned by janitors at night. The sizes of the two environments were 43 by 13 metres and 60 by 35 metres. The robot had to navigate through open plan office space, corridors, kitchens and laboratories. The environments were by no means static, with moving people, changing door states, re-arrangement of furniture and equipment, and a range of trolleys that intermittently appeared and disappeared during deliveries, maintenance and cleaning operation. Perceptually the environments also changed, with the most significant example being the day-night time cycles, which had an especially significant impact in areas with many windows.

We used a Pioneer 3 DX robot equipped with a panoramic imaging system, a ring of forward facing sensors, a Hokuyo laser range finder and encoders on the wheels (shown in Figure 4). All computation and logging was run onboard on a 2 GHz single core computer running Windows XP. The robot's typical operation speed was 0.3-0.5 ms^{-1}. The robot operated continuously for two to three hours be-

tween recharging cycles. Due to the need for supervision of the robot during operation (to ensure the integrity of the experiment), the total active time was limited to one to four recharge cycles in a typical day. In order to capture the effect of around-the-clock operation, experiments were conducted across all hours of the day and night.

3.1 Procedure

The experimenter first placed the charging station at a location in the Axon building. The experimenter then positioned the robot at a random location, turned it on, and initiated the software suite. The robot commenced exploration of the environment. After 117 minutes, the experimenter specified six desired delivery locations, by clicking on these locations in the experience map. Detecting the specification of delivery locations, the robot commenced a process of picking a delivery location at random and navigating to it to make a 'mock' delivery. When the robot detected a low battery state, it navigated to the recharging dock and docked to recharge. During re-charging the robot powered down in a fashion that retained the map, but lost localization.

Fig. 4. Photos of the two environments. (a) Cluttered open plan office space in Axon building, note the number of windows. (b) Robotics laboratory in Axon building. (c-e) Corridors in GP South building. Panoramas generated using Autostitch software demo [7].

After 8 days and 1017 delivery trials in the Axon environment, the experimenter turned off the robot and re-deployed it in the GP South environment. The experimenter also placed the charging station in the new environment, and then turned on the robot and initiated the software suite. To take advantage of the wider corridors, the experimenter made a single change to a movement parameter governing the top speed and obstacle clearance; this change was not required but enabled higher speed operation. The robot was not told that it had been put in a new environment, and since it had no path solutions to any existing delivery locations, commenced exploration. After 68 minutes, the experimenter specified five desired delivery locations, by clicking on the experience map. With delivery goals that were now accessible, the robot commenced random delivery navigation. After 54 delivery trials the robot returned to the charger to recharge, after which the robot was turned off and replaced in the original Axon environment, with no notification that it had changed environments. The robot commenced navigation to the original delivery locations. A further 72 delivery trials were conducted in the Axon building, before the experiment ended after 11 days.

4 Results

The results show the performance of the robot in its main task of navigating to delivery locations, and the stability of the performance over the 11 days of the trial. Results were obtained from analysis of more than 9 gigabytes of data including video from the robot, logged every time the robot docked to recharge.

4.1 Delivery Performance

Over the entire experiment, the robot had 1177 successful navigation trials (including navigation for deliveries and for recharging) and only one failure. The robot failed to complete delivery trial number 579, in that it did not get to the delivery location within five minutes. This failure was due to an extended localization failure in the open space in the robotics laboratory pictured in Figure 4(b). The global navigation system provided a local path to the local navigation system which could not be fulfilled, and the robot became ‘stuck’ in the corner, before eventually timing out the delivery attempt and reporting a delivery failure. The robot laboratory was one of the most challenging environments for the mapping system, because robot motion was relatively unconstrained in the large open space, leading to many unique template sequences. The delivery accuracy was worst for the delivery location in this room.

Figure 5 shows the accuracy of the global navigation system with respect to ground truth when making deliveries. The global navigation system uses 0.1 m as the acceptable tolerance for a delivery to be deemed complete. Perfect localization at delivery would be shown as a delivery error of 0.1 m. Goal location 3 was the delivery point located in open space in the robot laboratory.

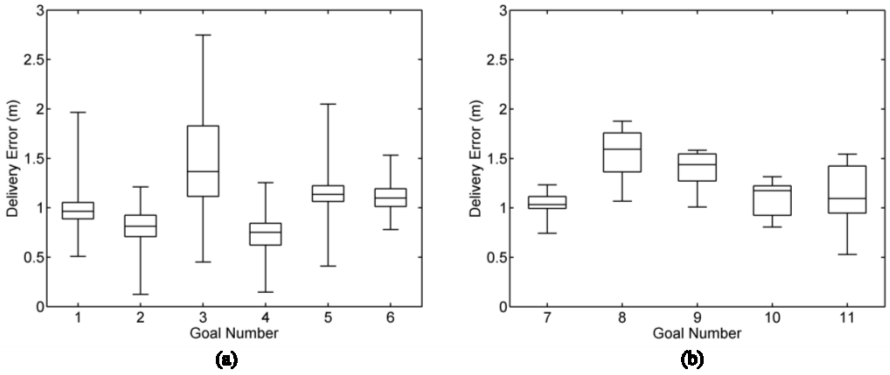

Fig. 5. Accuracy of delivery tasks, shown using a box whisker plot. Whiskers encompass the full range of delivery errors for each goal location.

4.2 Experience Maps

The experience map is the core global representation that forms the basis for navigation performance. In Figure 6, the experiences are plotted as a green circle at the (x, y) coordinates in the stored position **p**. Linked experiences are joined with a blue line. The spatial integrity of the map is important for efficient operation of the pruning algorithm, while topological integrity is important for path planning and navigation. The maps are not strictly accurate in a Cartesian sense, but need not be for effective navigation. For clarity, the map of GP-South in Figure 6(e) has been represented with an offset to the map of Axon. Note that there are no topological connections between the two buildings.

4.3 Stability

In order to address the question of the long term stability of the persistent navigation and mapping system, various measures were assessed to investigate trends in performance over time. The first indicator is the average time taken to navigate to each delivery location, plotted against time as in Figure 7. The delivery duration is dependent on the distance from the robot starting location and the randomly chosen delivery location. Durations are shown as averages over 100 deliveries, or over the period spent in the building whichever is shorter, smoothing the data sufficiently to obtain a measure of whether there is any increasing trend in navigation times as the map evolves. The graph shows that there is no increase in delivery times, even after the robot has mapped another building.

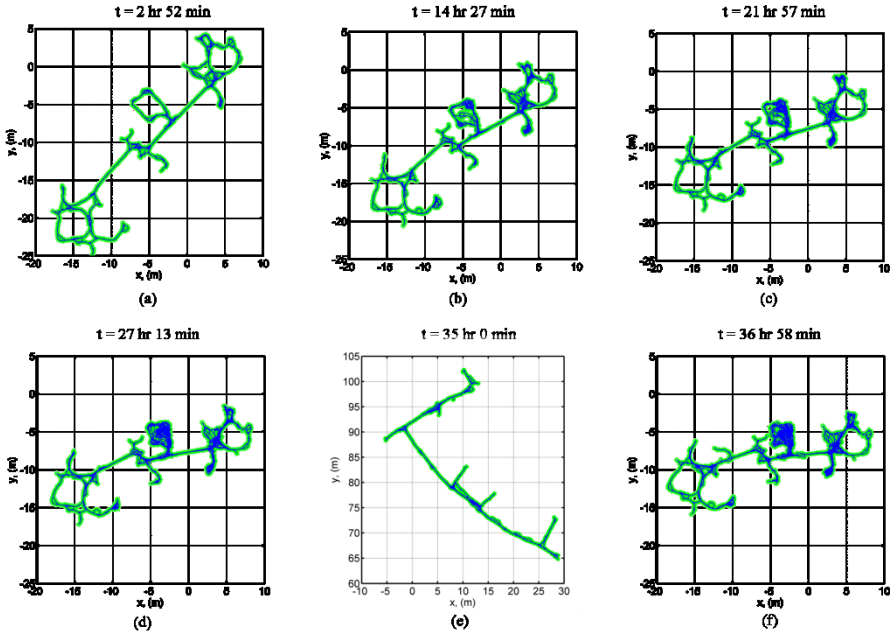

Fig. 6. Experience map progression for Axon and GP South buildings over the entire experiment. For the purposes of presentation, the experience map in the GP South building was initialized with an offset, to keep the maps separate.

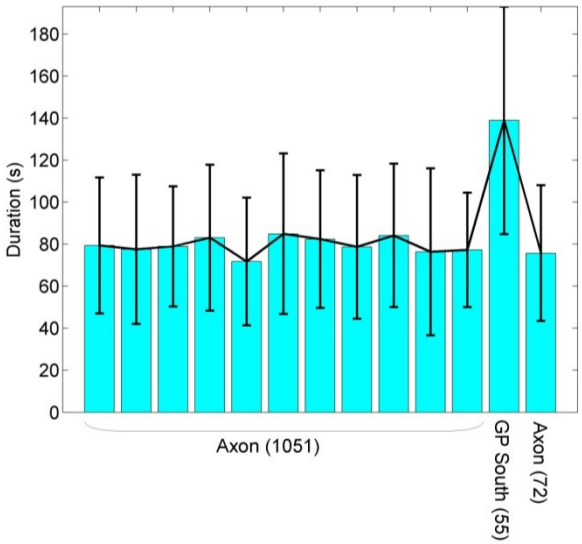

Fig. 7. Navigate-to-goal durations for the initial navigation trials in the Axon building (in groups of 100), General Purpose South building, and the final trials in Axon. Error bars show the standard deviation.

The second indicator of stability used is the number of experiences and visual templates retained by the system over time, shown in Figure 8. The number of experiences and visual templates in use is stable after the first two hours, although the robot does gradually learn extra sections of the environment. In particular, the robotics laboratory (see Figure 4(b)) continued to build new experiences as the robot visited new areas of the room on subsequent visits. The extended coverage of the room is illustrated by the map evolution shown in Figure 6. About 1200 new experiences and 800 new visual templates were learnt in 90 minutes when the robot was moved to GP South, after which the number of experiences reaches a plateau of 2500. When placed back into Axon building at 35 hours, a small number of new experiences and templates are learned.

There were ample computational resources at all times during the eleven day experiment to maintain a 7 Hz rate of vision processing, localization and navigation planning. Spare compute time was used to run the experience map relaxation algorithm in Equation 10, which stabilized at 50 Hz in the initial Axon trials, and dropped to 25 Hz when the extra GP-South trials were included.

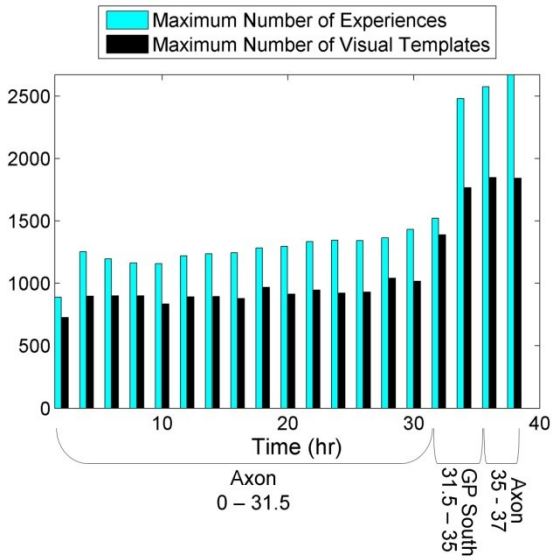

Fig. 8. Graph of the number of experiences and the number of view templates over the entire active duration of the experiment.

5 Conclusions

We presented the RatSLAM algorithm as a possible solution to the challenge of lifelong mapping and navigation. RatSLAM is significantly different to other

SLAM algorithms: it relies on remembering sequences of observed features across a trajectory of poses, rather than geometric optimization of motion and feature measurements. Our biological analogy of memory for places is built in a system inspired by the neural computation believed to take place in the rodent hippocampus. The neural inspiration lends itself readily to a paradigm of learning, remembering and forgetting, rather than geometry, probability and optimization. The flexibility introduced by our paradigm shift enables the creation of maps that can adapt to the constant change found in real environments.

It is important to realize that functional integrity rather than absolute Cartesian accuracy is the key criteria for success in lifelong navigation and mapping. Performance metrics must be principally concerned with the measurement of goal attainment, rather than the accuracy of the underlying map. In this paper, we have presented a series of metrics that are of principal importance to the measurement of success of a lifelong navigation and mapping system. The rate of goal attainment (>99.9%) is the key metric, with metrics showing the stability of the number of retained elements in the map providing evidence that this system could continue to function with high reliability for months or possibly years.

6 References

[1] A. Arleo and W. Gerstner, "Spatial Cognition and Neuro-Mimetic Navigation: A Model of Hippocampal Place Cell Activity," *Biological Cybernetics*, vol. 83, pp. 287-299, 2000.

[2] A. Barrera and A. Weitzenfeld, "Biologically-inspired robot spatial cognition based on rat neurophysiological studies," *Autonomous Robots*, vol. 25, pp. 147-169, 2008.

[3] N. Burgess, J. G. Donnett, K. J. Jeffery, and J. O'Keefe, "Robotic and neuronal simulation of the hippocampus and rat navigation," *Philosophical Transactions of the Royal Society of London B Biological Sciences*, vol. 352, pp. 1535-1543, 1997.

[4] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, pp. 34-46, 2007.

[5] D. E. Davis, J. T. Emlen, and A. W. Stokes, "Studies on Home Range in the Brown Rat," *Journal of Mammalogy*, vol. 29, pp. 207-225, 1948.

[6] T. Hafting, M. Fyhn, S. Molden, M.-B. Moser, and E. I. Moser, "Microstructure of a spatial map in the entorhinal cortex," *Nature*, vol. 11, pp. 801-806, 2005.

[7] http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html.

[8] M.J. Milford and G. Wyeth (2009) “Persistent Navigation and Mapping using a Biologically Inspired SLAM System,” *International Journal of Robotics Research*, in press.

[9] M. J. Milford, G. Wyeth, and D. Prasser, "RatSLAM: A Hippocampal Model for Simultaneous Localization and Mapping," presented at the *International Conference on Robotics and Automation*, New Orleans, USA, 2004.

[10] M. Milford and G. Wyeth, "Mapping a Suburb with a Single Camera using a Biologically Inspired SLAM System," *IEEE Transactions on Robotics*, vol. 24, pp. 1038-1053, 2008.

[11] D. Prasser, M. Milford, and G. Wyeth, "Outdoor Simultaneous Localisation and Mapping using RatSLAM," presented at the *International Conference on Field and Service Robotics*, Port Douglas, Australia, 2005.
[12] S. M. Stringer, E. T. Rolls, T. P. Trappenberg, and I. E. T. de Araujo, "Self-organizing continuous attractor networks and path integration: two-dimensional models of place cells," *Network: Computation in Neural Systems*, vol. 13, pp. 429-446, 2002.
[13] S. Thrun, D. Hahnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, and W. Whittaker, "A system for volumetric robotic mapping of abandoned mines," presented at the *International Conference on Robotics and Automation*, Taipei, Taiwan, 2003.
[14] S. Thrun and J. Leonard, "Simultaneous Localisation and Mapping," *Springer Handbook of Robotics*, Chapter 37, p. 871, 2008.