Mapping a Suburb With a Single Camera Using a Biologically Inspired SLAM System

Michael J. Milford, *Member, IEEE*, and Gordon F. Wyeth, *Member, IEEE*

***Abstract*—This paper describes a biologically inspired approach to vision-only simultaneous localization and mapping (SLAM) on ground-based platforms. The core SLAM system, dubbed RatSLAM, is based on computational models of the rodent hippocampus, and is coupled with a lightweight vision system that provides odometry and appearance information. RatSLAM builds a map in an online manner, driving loop closure and relocalization through sequences of familiar visual scenes. Visual ambiguity is managed by maintaining multiple competing vehicle pose estimates, while cumulative errors in odometry are corrected after loop closure by a map correction algorithm. We demonstrate the mapping performance of the system on a 66 km car journey through a complex suburban road network. Using only a web camera operating at 10 Hz, RatSLAM generates a coherent map of the entire environment at real-time speed, correctly closing more than 51 loops of up to 5 km in length.**

***Index Terms*—Bio-inspired robotics, monocular vision simultaneous localization and mapping (SLAM).**

I. Introduction

BIOLOGICAL systems solve the well-known robotics problem of simultaneous localization and mapping (SLAM) on a daily basis, with methods that are apparently robust, flexible, and well integrated into the creatures' sensory and behavioral systems. Introspectively, we know that looking at a single photograph is often enough to allow us to recall the location from which the photograph was taken, showing the strength of visual cues for localization. Simple mammals and even insects have shown this ability to store and organize visual cues, so that a single visual cue, or a brief sequence of cues, can globally relocalize the animal. Biology, it would therefore seem, has a great deal to teach us in the problem of visual SLAM.

Rodents, in particular, have been extensively studied with regard to their ability to store and recall visual cues. Most of these studies have been focused on the particular region of the brain in and around the hippocampus. While rodents also rely heavily on olfaction and whisking, most experiments are designed to remove these cues and to focus on the rodent's behavior in a controlled visually cued environment. External observations of rodent behavior are accompanied by neural recordings, which has led to the discovery of cells with strikingly clear correlates to mapping tasks such as *place* cells, *head direction* cells, and *grid* cells [1]–[3]. From a robotics point of view, this information is very helpful in designing a visual SLAM system. On the other hand, the data from the rat hippocampus is based on very sparse neural readings, and much conjecture exists on the exact computation that goes on. Furthermore, the experiments are conducted in environments that roboticists would consider trivially simple.

In this paper, we ask the question: Can computational models of rodent hippocampus perform as well as the state-of-the-art algorithms in visual SLAM? To address this question, we set a challenge for our biologically inspired SLAM system to map an entire suburb from a single webcam mounted on a car, at real-time speed. This is a challenge that would test the leading probabilistic approaches to visual SLAM, and so makes for an interesting comparison. In performing this challenge, the biologically based system reveals its strengths and weaknesses, offering insight into the lessons to be learnt from biology for visual SLAM systems.

The paper proceeds with a brief review of the state of the art in visual SLAM systems, paying particular attention to the common principles that underlie best practice and the standard of performance. The paper then reviews the current understanding of spatial encoding in rodent brains, bringing out the similarities and differences between the natural and artificial systems. A description of one of the standard computational methods used in the literature for modeling rodent hippocampus follows. The paper then details the RatSLAM system used in this study, showing the details of the computation required, and how the system can be driven from a single webcam. The results of the study are presented, revealing the benefits and drawbacks of a biologically inspired approach.

The paper draws upon work previously published in conference proceedings [4]–[6]. We also publish for the first time new

Manuscript received December 14, 2007; revised July 2, 2008. First published September 19, 2008; current version published October 31, 2008. This paper was recommended for publication by Associate Editor A. Davison and Editor L. Parker upon evaluation of the reviewers' comments. This work was supported in part by the Australian Research Council under a Discovery Project Grant and the Thinking Systems Initiative.

M. J. Milford is with the Queensland Brain Institute, The University of Queensland, St Lucia, Queensland 4072, Australia, and also with the School of Information Technology and Electrical Engineering, The University of Queensland, St Lucia, Queensland 4072, Australia (e-mail: milford@itee.uq.edu.au).

G. F. Wyeth is with the School of Information Technology and Electrical Engineering, The University of Queensland, St Lucia, Queensland 4072, Australia (e-mail: wyeth@itee.uq.edu.au).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org. provided by the author. This material includes one video demonstrating the RatSLAM system mapping the St Lucia suburb, provided in two different formats. It shows the formation of the experience map over the entire experiment at 1x to 50x real-time speed. The visual imagery obtained from the camera and used by the mapping system is also shown in synchrony with the mapping process, as well as a static aerial photo of the suburb. The size of the video is 22 MB in stlucia_wmv9.avi format and 35 MB in stlucia_mpeg4.avi format. stlucia_wmv9.avi is playable in Windows Media Player versions 9 and above. stlucia_mpeg4.avi uses the xvid MPEG4 codec (1.1.2 or later). Contact milford@itee.uq.edu.au for further questions about this work.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2008.2004520

results showing online mapping and localization during a 66 km journey through a suburban road network.

II. State of the Art in Visual SLAM

Most of the current work in visual SLAM builds on the probabilistic techniques developed with active sensors such as laser range-finders and sonar. Typical approaches maintain a probabilistic representation of both the robot's pose and the locations of features or landmarks in the environment. During self-motion, the pose probability distribution is updated using odometry and a motion model of the robot. Observations of landmarks are used to update the robot pose distribution as well as the landmark location distributions. New landmarks are added and given their own distributions. There are a number of now well-established SLAM methods that use probabilistic filters such as extended Kalman filters (EKFs), particle filters, and occupancy grids [7]–[12].

A number of researchers have been making steady advances in visual SLAM. Davison *et al.* [13] have achieved reliable real-time visual SLAM in small environments using an EKF based around real-time structure from motion. Their system, dubbed MonoSLAM, uses reasonable assumptions about the motion dynamics of the camera and an active search for visual features with a single hand-held camera to achieve real-time SLAM. This work has been expanded to outdoor environments [14] by strengthening the data association and adopting a Hierarchical Map approach in order to close a loop 250 m long.

Methods for probabilistic SLAM are not the focus of this paper, but we are particularly interested in the performance of other camera-only SLAM systems in outdoor environments to gauge the standard that we have set for our biologically inspired system. While not strictly a SLAM system (as the system has no self-motion estimate, and consequently, no pose-estimation filter) [15] has demonstrated loop closure over a 1.6 km path length based solely on the appearance of the image. Conversely, [16] showed closure of a 1.4 km long loop, but combined vision with odometry available from wheel rotation measurements to produce the metric level of the map representation, and similarly, the commercial vSLAM system [17] can build a semimetric map by combining odometry with visual scale-invariant feature transform (SIFT) features, although only independent tests published in [18] have shown its effectiveness outdoors and then only over a range of 200 m. The result by Clemente [14] seems to set the benchmark for vision-only SLAM in the outdoors, while the challenge presented for our system (mapping a suburb) would seem beyond even visual SLAM systems augmented with independent odometry.

III. Spatial Encoding in Rodent Brains

Extensive neural recordings from rodents over the past 30 years have shown that rodents maintain a representation of their own pose. Certain cells, such as *place cells*, fire consistently when the rat is at a particular location in the environment, but not elsewhere [2]. Controlled experiments have demonstrated that rodents can update their representation of pose based on estimates of self-motion obtained from copies of motor commands and vestibular information. Rats are also able to update and even "relocalize" their neural estimate of pose using external sensing such as vision, olfaction, and whisking. Both these abilities parallel how robot SLAM systems use odometry and external sensors such as lasers and cameras to update and correct the estimates of robot pose. However, unlike robots that perform SLAM, rats do not build detailed geometrical representations of the environments. Rats rely on the learnt associations between external perception and the pose belief created from integration of self-motion cues.

A. Cell Types

The discoveries of cells with strong spatial characteristics were driven by recordings of neural activity in single (or small groups of) cells as a rodent moved around in a small arena. *Place cells* fire maximally when the rodent is located at a specific location in the environment, and fire to a lesser degree as the rodent moves away from this location. *Head direction* cells fire when the rodent's head is at specific global orientations, but their firing is not correlated to the location of the animal's body [3], [19], [20]. Although the behavior of both place and head direction cells can be modified by many other factors, they can be thought of as somewhat complementary cell types, one providing positional information, the other directional.

Recently, a new type of spatial encoding cell called a *grid cell* was discovered in the entorhinal cortex, an area closely related to the hippocampus proper [1], [21], [22]. Grid cells are most notable as showing place cell-like properties but with multiple firing fields; a single grid cell will fire when the rat is located at any of the vertices of a tessellating hexagonal pattern across the environment. There are also *conjunctive grid cells*-–cells that fire only when the rat is at certain locations *and* facing in a specific orientation.

B. Spatial Encoding Behavior

All of the cell types have two fundamental characteristics: they are anchored to external landmarks and they persist in darkness. Experiments with landmark manipulation show that the rodent brain can use visual sighting of familiar landmarks to correct its pose estimation, performing a similar function to the update process in robot SLAM. As with robots, pose estimates degrade with time in the absence of external cues [23], suggesting similar computation to the prediction process in robot SLAM. However, there is no neural data from rats with respect to loop closure and larger environments, due in part to the limitations of recording techniques and the behavioral range of the caged laboratory rats. It is interesting to note that the discovery of grid cells, hailed as one of the most significant discoveries in neuroscience of the past 30 years [24], only came about when the rat's arena size was increased to a mere 1.0 × 1.0 m [1].

IV. Continuous Attractor Networks

Continuous attractor networks (CANs) are often used to model the behavior of place, head direction, and grid cells,

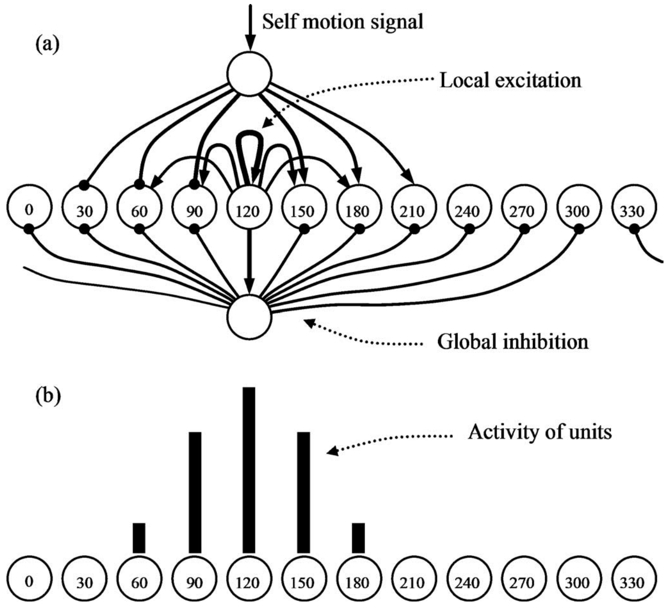


Fig. 1. (a) Excitatory (arrows), inhibitory (round), and self-motion connections for a continuous attractor network representation of head direction cells. (b) A stable activity packet centered at 120°.

especially in robotic studies [25]–[29]. A CAN is a type of neural network that has an array of neural units with weighted connections. The neural units compute activity by summing the activity from other neural units through the weighted connections. CANs have many recurrent connections and behave differently to the better known multilayer feed-forward neural networks. The recurrent connections cause the network to converge over time to certain states (attractors) in the absence of external input.

A CAN is used to represent the robot's pose in the study presented in this paper. Its properties are significantly different from the usual probabilistic representations found in SLAM algorithms, as the descriptions in this section show.

A. Attractor Dynamics

The properties of a CAN can best be explained in the context of a head direction network, in which the activity of neural units encodes the one-dimensional state variable of the rodent's head orientation. Fig. 1(a) shows a ring of head direction units unraveled into a single line—the units at each end are connected to those at the opposite end. Each unit excites itself and units near itself (using the excitatory connections shown as arrows), while inhibiting all other cells (using the round-shaped inhibitory connections). In the absence of the self-motion signal, the combination of local excitation and global inhibition leads to the formation of a single stable peak, as in Fig. 1(b). Many varied configurations of units and weights have been used to achieve this behavior [25]–[29]. Note that the weights in Fig. 1 are shown only for the unit representing a 120 degree head direction; the weights are repeated for each unit.

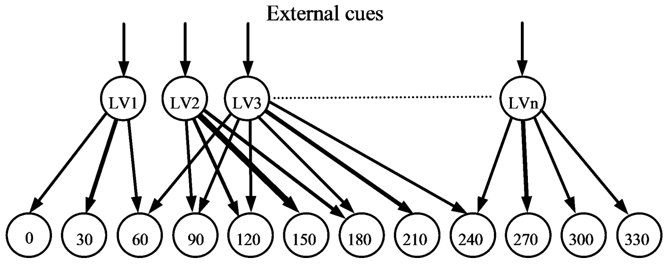


Fig. 2. Connections for calibration of head direction from local view. The connections shown here operate in parallel to the connections shown in Fig. 1(a).

B. Path Integration

The activity in the network can be shifted by signals representing self-motion in the process of path integration. Fig. 1(a) illustrates how a positive self-motion signal excites units representing a larger angle of head direction, while inhibiting units representing the lesser angle. Zhang [28] shows that, in certain conditions, the peak can be moved without deformation using weights that are the derivative of the local excitatory weights used to form the attractor peak.

Naturally, errors in path integration accumulate over time. Unlike probabilistic SLAM, path integration with a CAN does *not* carry a representation of the uncertainty accumulated over time, as seen, for example, in the spread of particles in the prediction cycle of a particle filter. The width and height of the activity packet in a CAN stays constant under path integration.

C. Local View Calibration

To maintain a consistent representation, the head direction cell firing direction can be reset by environmental cues [20], [23]. One of the methods postulated for this mechanism is that the head direction cells are also excited by connections from *local view cells* that represent the presence of cues at specific locations with respect to the pose of the rodent's head [30]. For example, a local view cell might represent a snapshot of the environment as a rat runs along a wall. In order to learn the association between the snapshot of the environment and the current head direction, coactivated local view and head direction cells form stronger connections through a process such as Hebbian learning. When the visual scene is encountered again, the active local view cells inject activity through the learnt connections into the head direction cells associated with that scene. Fig. 2 shows an example network after some learning has taken place, forming connections of varying strengths from local view cells to head direction cells.

The injected activity from the local view cells is filtered by the attractor dynamics of the network. If activity from the local view cells is injected into an inactive region of the network (as it is during loop closure correction), the global inhibition connections tend to suppress the new packet, while the local excitatory connections sustain the old packet. Only a well-ordered and coherent sequence of local view activity can cause a large change of position of the activity packet in the network. The attractor dynamics create a filter that rejects spurious or ambiguous information from the local view cells.

There are some very clear distinctions between the process of local view calibration and the apparently analogous measurement update cycle in a SLAM algorithm. In a probabilistic filter, the observation model computes the likelihood of the current sensor reading when the robot pose and landmark locations are known. In RatSLAM, there is no notion of landmark locations; only statistics of views are gathered with respect to robot pose. Local view calibration does not compute the likelihood of a view for a pose, but rather generates the associated poses for a given view. Errors in association are filtered by the attractor dynamics.

D. Extending to Two Dimensions

The one-dimensional head direction model can be extended to form a two-dimensional continuous attractor model for place. A two-dimensional CAN is modeled as a sheet of units, again with local excitation and global inhibition to form a spontaneous peak of activity. The peak can be moved by a mechanism analogous to the one-dimensional path integration system and reset by learnt associations with local view cells.

The wrap-around circular nature of the head direction network has also inspired a solution to make a two-dimensional network suitable for use over large areas [29]. If the network has boundaries, then the representation will fail when the rodent leaves the area covered by the network. However, if the network is configured so that each edge of the network is connected to the opposite side (forming a single toroidal surface), then the representation continues, with the reuse of cells in a tessellated pattern across the environment.

E. Robotic Implementations

The strong spatial characteristics of head direction and place cells have led to a small number of computational models that have actually been implemented on mobile robots [31]–[33]. The purpose of most of these studies has been to test the model's fidelity to biological results, rather than to create a mapping system with the best possible performance. In addition, most of these studies predate the discovery of grid cells in entorhinal cortex, which have provided a rich new source of biological inspiration for roboticists.

The following section outlines our implementation of a simplified model of rodent hippocampus on a robot. Some of the work, especially in the early stages, drew heavily upon these pioneering robot studies with models of hippocampus, especially [31], [32]. However, in contrast to other robot implementations, our focus is very much on achieving the best possible robot mapping and localization performance, rather than strong biological plausibility.

V. RatSLAM

In our previous work, we have developed a system called RatSLAM that draws upon the current understanding of spatial encoding in rat brains to perform real-time learning and recall of semimetric maps on real robots [6], [34], [35]. In this section, we describe the components of the RatSLAM system, and how they have been adapted to suit the problem of large-scale mapping and localization for a vision-only system.

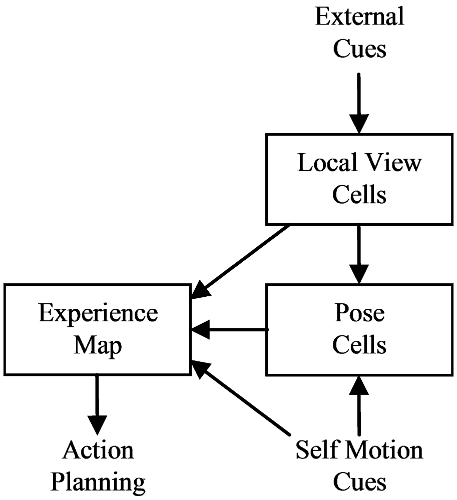


Fig. 3. Broad connectivity between functional regions in the RatSLAM system.

A. Overview

The RatSLAM model, as presented here, is the result of extensive development with a focus on mapping performance achieved in real-time on robot hardware. Consequently, a number of simplifications and additions have been made to more biologically plausible models, while still retaining the core concept of spatially selective cells that respond to internal and external sensory information. To prevent any nomenclature confusion between the biological and robotic system components, the RatSLAM components bear different names to their biological counterparts. The components and their organization are shown in Fig. 3.

The *pose cells* form the core of the RatSLAM system. The pose cells represent the three degree of freedom (DoF) pose of the robot using a three-dimensional version of the CAN described in the previous section. Each face of the three-dimensional pose cell structure is connected to the opposite face with wraparound connections, as shown in Fig. 4. In large environments during self-motion, activity wraps around the pose cell structure many times, resulting in firing fields (the locations in the environment at which the cell fires) that form a tessellating pattern similar to that of grid cells. Consequently, an individual pose cell can become associated with many different robot poses in physical space.

Activity in the pose cells is updated by self-motion cues, and calibrated by local view. The self-motion cues are used to drive path integration in the pose cells, while the external cues trigger local view cells that are associated with pose cells through associative learning. In this study, both the local view and the self-motion cues are generated from camera images.

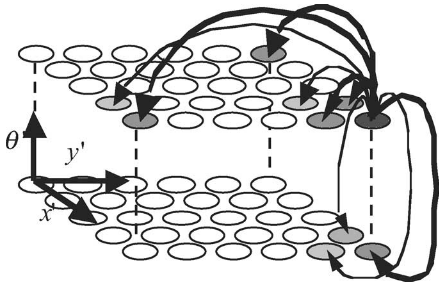


Fig. 4. Continuous attractor network representation of the pose cells, with wraparound excitatory connections (arrows) in all three dimensions creating a stable activity packet (cell shading).

The role of the *experience map* is to organize the flow of information from the pose cells, the local view cells, and the self-motion cues into a set of related spatial experiences. An individual experience in the experience map is defined by the conjunction of the pattern of activity in the pose cells (the *pose code*) and the pattern of activity in the local view cells (the *local view code*). When either or both of the pose or local view codes change, a new experience is formed, and is linked to the previous experience by a transition that encodes the distance found from self-motion cues. New experiences and transitions will continue to form as the robot enters new areas. During loop closure, when the robot revisits a known area, the pose and local view codes are the same as in a previous experience. The experience mapping algorithm performs map correction by aligning the experience locations and orientations in the experience map with the transition information stored between the experiences.

The principle purpose of the experience map is to form a representation that is suitable for action planning [6], [36], and the strength of the representation is that it maintains strong topological coherence. However, it can also be used to form a semimetric map by plotting the experience positions and transitions between experiences. The key results in this paper are shown as plots of the experience map at the end of the experiment. It is important to note that the maps are topologically connected using the transitions between experiences, and that the metric location of experiences is not a measure of connectedness.

The following sections give more details for the computation involved in the pose cells and experience map. Section VI describes how a single camera can generate the self-motion cues and the activity in the local view cells.

B. Pose Cells

The activity in the pose cells is described by the pose cell activity matrix P, and is updated by the attractor dynamics, path integration, and local view processes. The pose cells are arranged in three dimensions with two dimensions (x', y') representing a manifold on absolute place (x, y) and one dimension θ' as a manifold on absolute head direction θ (see Fig. 4).

1) Attractor Dynamics: A three-dimensional Gaussian distribution is used to create the excitatory weight matrix $\varepsilon_{a,b,c}$, where the indexes a, b, and c represent the distances between units in x', y' and θ' coordinates, respectively. The distribution is calculated by

$$\varepsilon_{a,b,c} = e^{-\left(a^2+b^2\right)/k_p} e^{-c^2/k_d} \tag{1}$$

where k_p and k_d are the width constants for place and direction, respectively. The change of activity in a pose cell due to local excitation is given by[1]

$$\Delta P_{x',y',\theta'} = \sum_{i=0}^{(n_{x'}-1)} \sum_{j=0}^{(n_{y'}-1)} \sum_{k=0}^{(n_{\theta'}-1)} P_{i,j,k}\varepsilon_{a,b,c} \tag{2}$$

where $n_{x'}$, $n_{y'}$, $n_{\theta'}$ are the three dimensions of the pose cell matrix in (x', y', θ') space. The calculation of the excitatory weight matrix indexes cause the excitatory connections to connect across opposite faces; thus

$$\begin{aligned} a &= (x' - i)(\mathrm{mod} n_{x'}) \\ b &= (y' - j)(\mathrm{mod} n_{y'}) \\ c &= (\theta' - k)(\mathrm{mod} n_{\theta'}) \end{aligned} \tag{3}$$

The computation of (2) across all pose cells is potentially expensive, as it represents a circular convolution of two three-dimensional matrices. Significant speedups are achieved by exploiting the sparseness of the pose cell activity matrix; for the parameters used in this paper, typically around 1% of cells have nonzero values.

Each cell also inhibits nearby cells using an inhibitory form of the excitatory weight matrix, with the same parameter values, but negative weights. By performing inhibition after excitation (rather than concurrently), and adding slight global inhibition, the symmetry of the excitatory and inhibitory weights leads to suitable network dynamics, without using traditional Mexican hat connectivity [37]. Consequently, the network is easier to work with, not requiring separate tuning of different excitatory and inhibitory weight profiles. The slight global inhibition is applied equally across all cells, with both inhibition processes given by

$$\Delta P_{x',y',\theta'} = \sum_{i=0}^{n_{x'}} \sum_{j=0}^{n_{y'}} \sum_{k=0}^{n_{\theta'}} P_{i,j,k}\psi_{a,b,c} - \varphi \tag{4}$$

where $\psi_{a,b,c}$ is the inhibitory weight matrix and φ controls the level of global inhibition. All values in P are then limited to nonnegative values and normalized.

Without external input, the activity in the pose cell matrix converges over several iterations to a single ellipsoidal volume of activity, with all other units inactive. The next two sections briefly describe how activity can shift under a process of path integration, and how activity can be introduced by local view calibration.

2) Path Integration: Rather than computing weighted connections to perform path integration, RatSLAM increases both the speed and accuracy of integrating odometric updates by making an appropriately displaced copy of the activity packet [38]. This approach sacrifices biological fidelity, but computes

[1] These indexes must be set to $n - 1$ so that (3) will work.

faster, has no scaling problems, and does not require particularly high or regular update rates. Note that unlike probabilistic SLAM, there is no notion of increasing uncertainty in the path-integration process; the size of the packet does not change under path integration. Path integration for the studies presented in this paper is based on visual odometry signals that are described in Section VI.

3) Local View Calibration: The accumulated error in the path-integration process is reset by learning associations between pose cell activity and local view cell activity, while simultaneously recalling prior associations. The local view is represented as a vector V, with each element of the vector representing the activity of a local view cell. A local view cell is active if the current view of the environment is sufficiently similar to the previously seen view of the environment associated with that cell.

The learnt connections between the local view vector and the three-dimensional pose cell matrix are stored in the connection matrix β. Learning follows a modified version of Hebb's law where the connection between local view cell V_i and pose cell $P_{x', y', \theta'}$ is given by

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

which is applied across all local view cells and pose cells that are active. The change in pose cell activity under calibration is given by

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

where the δ constant determines the strength of visual calibration, and n_{act} is the number of active local view cells. The method for computing the activity in the local view cell vector is given in Section VI.

C. Experience Map

An experience map is a fine-grained topological map composed of many individual experiences, e, connected by transitions, t. Each experience e_i is defined by its associated union of pose code P^i and local view code V^i, where code refers to the pattern of activity in a cell group. The experience is positioned at position $\mathbf{p}^i$ in experience space—a space that is a useful manifold to the real world. An experience can then be defined as a 3-tuple

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

The first experience is created at an arbitrary starting point, and subsequent experiences build out from the first experience over transitions.

1) Experience Creation: When the pose code or local view code is sufficiently different from a stored experience, a new experience is created. The pose and local view codes of existing experiences are compared to the current pose and local view code through a score metric S:

$$S = \mu_p \left|P^i - P\right| + \mu_v \left|V^i - V\right| \tag{8}$$

where μ_p and μ_v weight the respective contributions of pose and local view codes to the matching score. When the score for all current experiences exceeds a threshold $S_{\max}$, a new experience is created, with an associated transition. The transition t_{ij} stores the change in position as measured from odometry

$$t_{ij} = \left\{\Delta\mathbf{p}^{ij}\right\} \tag{9}$$

where $\Delta\mathbf{p}^{ij}$ is the change in the vehicle's pose according to odometry. t_{ij} forms the link between previous experience e_i and new experience e_j such that

$$e_j = \{P^j, V^j, \mathbf{p}^i + \Delta\mathbf{p}^{ij}\}. \tag{10}$$

Note that this equation holds only at experience creation; $\mathbf{p}^j$ is likely to change under loop closure.

2) Loop Closure: There is no explicit loop detection; rather, loop closure occurs when the pose code and local view code after a change in experience sufficiently match a stored experience. When this occurs, it is highly unlikely that the summed change in position of the transitions leading to the experience at closure will match up to the same position. To move toward a match, the positions of all experiences are updated using

$$\Delta\mathbf{p}^i = \alpha \left[\sum_{j=1}^{N_f} (\mathbf{p}^j - \mathbf{p}^i - \Delta\mathbf{p}^{ij}) + \sum_{k=1}^{N_t} (\mathbf{p}^k - \mathbf{p}^i - \Delta\mathbf{p}^{ki})\right] \tag{11}$$

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. In these experiments, α is set to 0.5 (larger values can lead to map instability). The map update process occurs continually, but is most apparent during loop closures.

3) Reading the Experience Map: A visual readout of the experience map can be obtained by plotting the positions of the experiences joined by their respective transitions. In other work, we have stored behavioral and temporal data in the transition tuple to aid path planning [6], [38], and frequency of use data in both the experience and transition tuples to aid map maintenance [38].

VI. Vision System

The RatSLAM system requires two outputs from a vision-processing system: self-motion information and recognition of familiar visual scenes. Unlike the mapping and localization system, the vision system used in this paper has no biological inspiration. Rather than trying to adapt the models of rodent vision to function in a vastly different context (both in terms of environment and sensors), we use a lightweight method for visual odometry and image matching. The visual odometry and image matching methods are noisy and prone to error and by no means state of the art. There are proven methods for achieving more reliable and accurate odometry [39], and for matching in image space [40]. However, the vision system presented here computes in real-time on standard desktop or laptop computing hardware, even when sharing resources with the mapping system. As the results will show, the mapping and

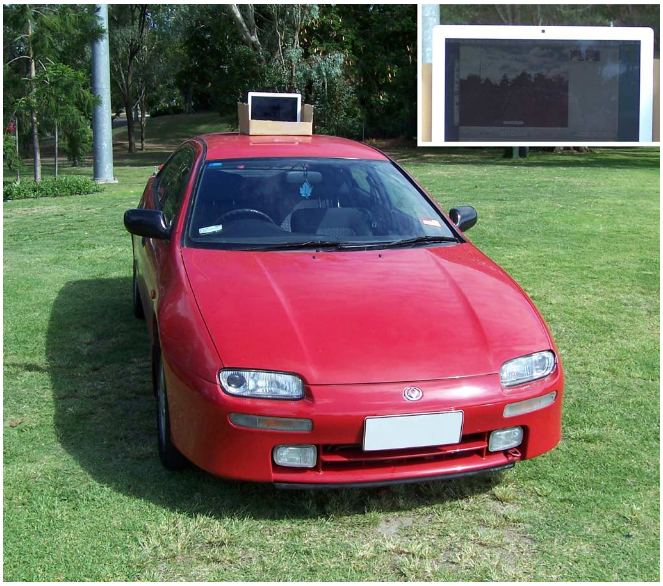


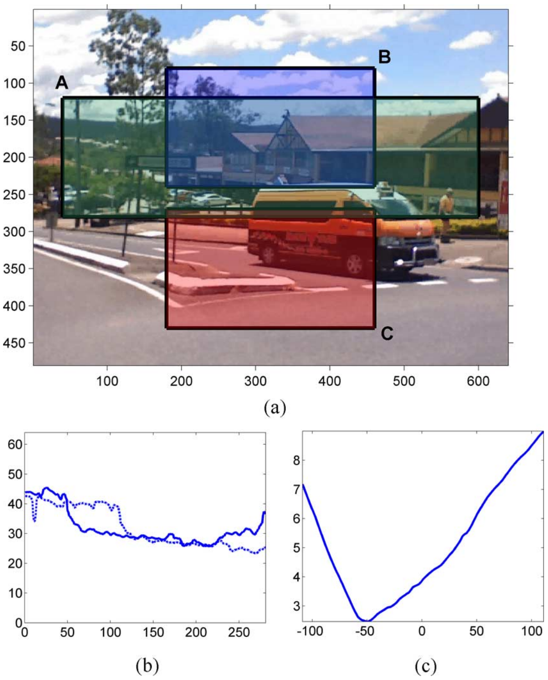
Fig. 5. Apple Macbook with built-in iSight video camera (inset) mounted on the test vehicle. This camera was the sole source of sensory information for the mapping system. The laptop was mounted on the roof of the car in a forward facing, zero pitch position.

localization method is tolerant of noise in both odometry and scene matching.

A. Image Acquisition

The camera used for this research was the built-in *iSight* camera on an Apple *Macbook* notebook computer (see Fig. 5). The camera is fixed-focus and uses an active pixel sensor rather than charge-coupled device (CCD). Grayscale images were captured at a frame rate of 10.0 frames per second at a resolution of 640 × 480 pixels. Images were cropped into three regions for use by the rotational odometry, translational odometry, and image matching modules [see Fig. 6(a)]. Each image region served a different purpose—image matching was performed off region A, which was more likely to contain useful landmark information while generally removing redundant information from the ground and sky. Region B provided a higher proportion of distal cues that could be tracked for vehicle rotation detection. Region C contained primarily ground plane information, allowing the calculation of translational speed without the errors introduced by optical flow discrepancies between narrow wooded tracks and large open multilane roads.

The vision algorithms use a scanline intensity profile formed from the subimages (much like the profile used in Carnegie Mellon University's (CMU) rapidly adapting lateral position handler (RALPH) visual steering system [41]). The scanline intensity profile is a one-dimensional vector formed by summing the intensity values in each pixel column, and then normalizing the vector. This profile is used to estimate the rotation and forward speed between images for odometry, and to compare the current image with previously seen images to perform local view calibration.



(a)

(b)

(c)

Fig. 6. All rotational, speed, and scene information was extracted from grayscale 640 × 480 pixel images (shown in color here for clarity). (a) Based on the ground plane assumption, upper image regions were used for scene recognition (A) and rotation detection (B) as they were more likely to contain distinct visual information and distal cues. Lower ground regions were less distinctive, but provided a more consistent speed estimate across constricted and open areas. (b) Image arrays corresponding to two consecutive scanlines. (c) Graph showing adjusted scanline differences for shifts in their relative positions. The best match occurs for a shift of about −50 pixels to the left.

B. Estimating Rotation

Rotation information is estimated by comparing consecutive image arrays. Fig. 6(b) shows the scanline intensity profiles from two consecutive images. The comparison between profiles is performed by calculating the average absolute intensity difference between the two scanline intensity profiles $f(s)$ as they are shifted relative to each other

$$f\left(s, I^j, I^k\right) = \frac{1}{w - |s|}\left(\sum_{n=1}^{w-|s|}\left|I^j_{n+\max(s,0)} - I^k_{n-\min(s,0)}\right|\right) \tag{12}$$

where I^j and I^k are the scanline intensity profiles to be compared, s is the profile shift, and w is the image width. Fig. 6(c) shows the average absolute intensity differences for shifts of the first image array (dotted line). The pixel shift s_m in consective images I^j and I^k is the value of s that minimizes $f()$ for those two profiles

$$s_m = \underset{s\in[\rho-w,w-\rho]}{\arg\min}\ f\left(s, I^j, I^k\right) \tag{13}$$

where the offset ρ ensures that there is sufficient overlap between the profiles. In the experiments that follow $w = 280$ and $\rho = 70$. For the two example images in Fig. 6, s_m is found at a rotation of -50 pixels. The pixel shift is multiplied by the gain constant, σ, (which can be calculated either empirically or from the camera's intrinsic parameters) to convert the estimated shift in pixels into an angular shift $\Delta\theta$

$$\Delta\theta = \sigma s_m . \tag{14}$$

The rotation calculation relies on a few assumptions, first and foremost that the camera is forward facing with respect to motion, and that the movement of the camera has little or no translational movement parallel to the camera sensor plane. Mounting the camera on the roof of a road vehicle satisfies these constraints.

C. Estimating Speed

Speeds are estimated based on the rate of image change, and represent movement speed through perceptual space rather than physical space. As can be seen later in the results section, when coupled with an appropriate mapping algorithm, this approach can yield environment maps that are strongly representative of the environment. The speed measure v is obtained from the filtered average absolute intensity difference between consecutive scanline intensity profiles at the best match for rotation

$$v = \min\left[v_{\mathrm{cal}} f(s_m, I^j, I^{j-1}), v_{\max}\right] \tag{15}$$

where v_{cal} is an empirically determined constant that converts the perceptual speed into a physical speed, and $v_{\max}$ is a maximum velocity threshold. By calculating the image difference using the best matched scanline intensity profiles, the effect of rotation is mostly removed from the speed calculation. The threshold $v_{\max}$ ensured that spuriously high image differences were not used. Large image differences could be caused by sudden illumination changes such as when traveling uphill facing directly into the sun.

D. Local View Cell Calculation

A local view cell is active if the current view of the environment is sufficiently similar to the previously seen view of the environment associated with that cell. In the implementation described in this paper, the current view is compared to previous views using the same scanline intensity profile that formed the basis of odometry. Previously seen scanline intensity profiles are stored as view templates with each template paired to a local view cell. The current scanline intensity profile is compared to the stored templates, and if a match is found, the local view cell associated with the matching stored template is made active. If there is no match to a previously seem template, then a new template is stored and a new local view cell created that is associated with the new template, as shown in Fig. 7.

The comparison between the current profile and the templates is performed using the average absolute intensity difference function in (12). The comparison is performed over a small range of pixel offsets ψ to provide some generalization in rotation to the matches. The best match is found as

$$k_m = \arg\min_k f\left(s, I^j, I^k\right), \qquad s \in [-\psi, \psi] \tag{16}$$

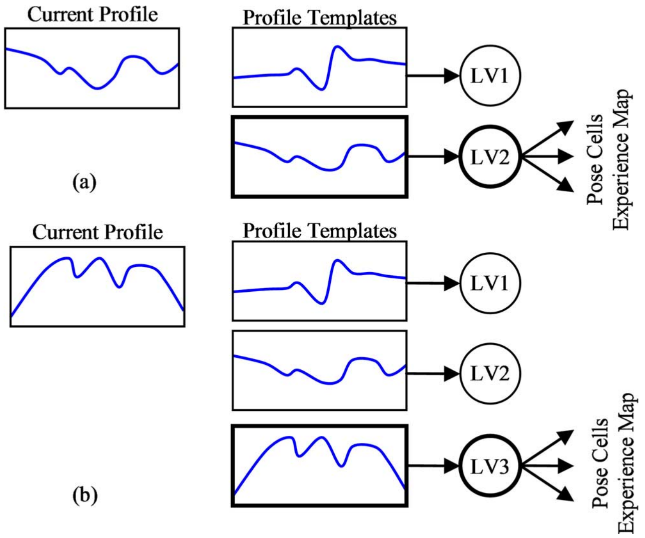


Fig. 7. Template matching system for local view calculation. (a) Where the current profile sufficiently matches a previously stored template, the associated local view cell is activated. (b) For a novel profile, a new template and local view cell is created and activated.

where I^j is the current profile and I^k are the stored profiles. The quality of the match d is calculated by

$$d = \min_{s \in [-\psi, \psi]} f\left(s, I^j, I^{k_m}\right) \tag{17}$$

and tested against a threshold d_m to determine whether the profile is sufficiently similar to the template or whether a new template need be created, and k_m updated accordingly. The local view vector is then set

$$V_i = \begin{cases} d_m - d_i, & d_i \le d_m \\ 0, & d_i > d_m \end{cases} \qquad \forall i . \tag{18}$$

VII. Experimental Setup

The challenge set for the system was to map the entire suburb of St Lucia in Brisbane, Australia, from a laptop's built-in webcam mounted on the roof of a car. St Lucia is a challenging environment to map with many contrasting visual conditions: busy multilane roads, quiet back streets, wide open campus boulevards, road construction work, tight leafy lanes, monotonous suburban housing, highly varied shopping districts, steep hills, and flat river roads. The maximum dimensions of the road network were approximately 3.0 km in the east–west direction and 1.6 km in the north–south direction. The environment is more than five times the area of the environment mapped in [42] using laser detection and ranging (LADAR) scans, which, according to the researchers, was the largest environment mapped by a single laser scanner as of 2007.

The dataset was gathered on a typical fine spring Brisbane day starting in the late morning. During test set acquisition, the vehicle was driven 66 km over 100 min at typical driving speeds up to the speed limit of 16.7 m/s (60 km/h). The car was driven around the road network such that every street was visited at least once, and most were visited multiple times. In past work, a panoramic camera has been used to bind forward

TABLE I
POSE CELL PARAMETERS

Parameter	Value
$n_{x'}, n_{y'}, n_{\theta'}$	30 × 30 × 36
Nominal pose cell size	10 m × 10 m × 10°
k_p	7 cells (70 m)
k_d	7 cells (70°)
Φ	0.00002

TABLE II
LOCAL VIEW CELL PARAMETERS

Parameter	Value
λ	0.25
d_m	3.2
A	0.5
δ	0.4

TABLE III
EXPERIENCE MAP PARAMETERS

Parameter	Value
α	0.5
μ_p	0.5
μ_v	0.5
S_{max}	1

TABLE IV
OTHER PARAMETERS

Parameter	Value
σ	0.0828 degrees/pixel
v_{cal}	13.2 m/s
v_{max}	18.5 m/s

and reverse traverses of paths [35]; in this experiment, roads were only driven in one direction. The road network consists of 51 inner loops of varying size and shape, and includes over 80 intersections including some large roundabouts. Images were obtained at a rate of ten frames per second from the camera, and saved to disk as a movie.

A. Parameter Setting

The equations and algorithms that drive the RatSLAM mapping and vision-processing system contain several parameters, which are given in Tables I–IV along with their values. A brief summary of the significance, selection, and sensitivity of these parameters is given here.

Most system parameters were left at values that had been found through extensive tuning to give good performance in the wide range of experiments performed previously in indoor and outdoor environments [6], [34], [35]. All the experience map parameters remained unchanged, as did the pose cell weight variances and local view to pose cell learning rate. The nominal pose cell dimensions, which had previously been set at 0.25 m × 0.25 m × 10° for indoor environments, and 2 m × 2 m × 10° for small outdoor environments, were changed to 10 m × 10 m × 10°. These values ensured that at typical car speeds, activity in the pose cells shifted at a similar rate to that in previous work, removing the need to tune any other pose cell related parameters.

System performance was, however, critically dependent on the tuning of one parameter. The maximum template distance parameter, d_m, required tuning with the new vision system in order to provide a suitable balance of visual discrimination and generalization. Having too high a value would result in over generalization, meaning unexplored parts of the environment might generate too many false positive matches to already explored regions. Conversely, too low a value would result in the system being unable to recognize familiar places if there had been slight visual changes. The value was determined by taking the most visually challenging repeated section of road—in this case, an uphill section where the sun was shining directly into the camera on one pass—and gradually increasing the value until the system could recognize enough images to relocalize to that section correctly. This tuning procedure also had the effect of introducing some over generalization in other sections of the environment, but this had no significant negative effect on mapping performance.

Some other parameters had minor effects on mapping performance. The maximum velocity threshold $v_{\max}$ was introduced to avoid unreasonably large-speed estimates that could occur in periods of extreme image variation from frame to frame. An example of this occurred near the end of the experiment while driving through a forest, with the camera moving many times a second between being in dark shadows to having the sun shining directly into the lens [see Fig. 8(e) and (f)]. Without this threshold, such sections of environment could become so enlarged by spuriously high-speed estimates that the mapped road sections would expand to overlap other roads. However, while overlapping in the (x, y) experience map space, these mapped road sections would not be connected by any experience links. The RatSLAM navigation system [6] would still be able to plan and execute paths to locations, because paths are planned based on experience connectivity, rather than just map layout in the (x, y) plane. Overlap would, however, adversely affect both the map's usability by humans and the ability of the navigation system to perform shortcuts.

VIII. RESULTS

In this section, the overall mapping results are presented, as well as the output from the various system components. Videos showing the St Lucia environment being mapped accompany this paper, and are also available on the Web in the directory: http://ratslam.itee.uq.edu.au, as well as source code for the RatSLAM mapping system.

A. Experience Map

The experience map created by the mapping and localization system is shown in Fig. 9(b). For reference, an aerial photo of the test environment is also shown [see Fig. 9(a)], with the thick

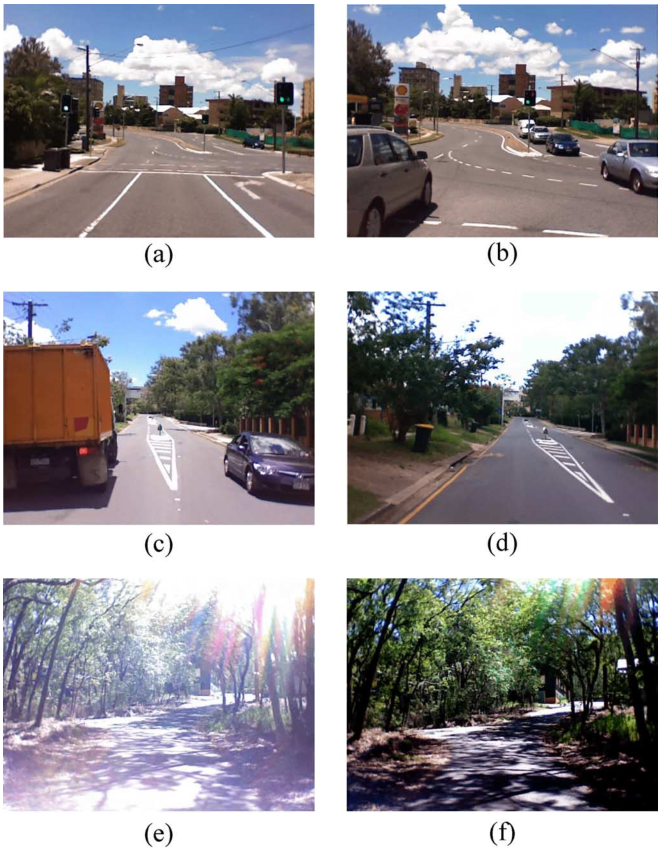


Fig. 8. Paired camera snapshots showing sections of the environment and the variation in their visual appearance during the experiment. (a) and (b) Traffic at a five-lane section of road. (c) and (d) Traffic and changing illumination. (e) and (f) Sun flare and changing illumination on a narrow track through dense forest. During the experiment the sun moved in and out of clouds, clouds formed, moved, changed shape, and disappeared, and shadows changed as the sun moved in the sky.

black line indicating the roads that were driven along during the experiment. The map contains 12 881 individual experiences, and 14 485 transitions between experiences. The map captures the overall layout of the road network, as well as finer details such as curves, corners, and intersections. Because the mapping system only had a perceptual rather than absolute measure of speed, the map is not geometrically consistent on a global scale, nor could it be expected to be. At one location $(-750, -250)$, the map splits into two representations of the same stretch of road, caused by very severe illumination variations hindering rerecognition of the stretch of road.

B. Map Correction

The experience map correction process continually rearranges the locations and orientations of experiences in order to match the transition information stored in the links between them. A measure of the map's stability and convergence can be gained by examining the average difference between the odometric information stored in the transitions and the actual experience locations within the map. Fig. 10 shows this value plotted over the entire duration of the experiment. The large initial loop closures can be clearly seen as spikes that then quickly decrease. After 1700 s, the map is already quite stable, and reaches a completely stable configuration by about 5000 s, before the experiment has finished. The steady-state value of 1.2-m indicates that on average, there is a 1.2-m discrepancy between the locations of two linked experiences and their relative locations, as dictated by the odometric information stored in the transition(s) that link them together.

C. Angular Velocity Estimates

Fig. 11(a) shows the vehicle's angular velocity as calculated by the vision-processing system for the winding section of road at the top left corner of the test environment. The plot captures most of the turns accurately although there are some erroneous spikes (at 1418 s, for example). Fig. 11(b) shows the trajectory of the vehicle as calculated using only the visual odometry information (solid line), without any map correction, as compared with the ground truth trajectory (dashed line). While the vehicle's rotation is accurately represented, the calculated vehicle speed for this section of road is significantly higher than in reality.

D. Translational Speed Estimates

Fig. 12 shows the vehicle's speed as calculated by the vision system, for a section of road during which the car was traveling at a constant speed. The speed signal was noticeably inferior to the angular velocity signal in two respects: the signal was noisier, with calculated speeds noticeably jumping around even when the vehicle was moving at a constant speed, and there were local gain biases in particularly cluttered or open areas of the environment. It would, however, be difficult to use traditional optical flow techniques given the low frame rate of 10 Hz, the high vehicle speed, the low field of view, and the low quality of the images produced by the Web camera.

E. Relocalization in the Pose Cells

Fig. 13 shows a sequence of snapshots of activity in the pose cell matrix during a typical relocalization event. At 1138 s, a familiar visual scene activates local view cells, which, in turn, activate pose cells through local view—pose links, causing a competing activity packet to appear above the existing activity packet. More familiar visual scenes inject further activity causing the new packet to become slightly dominant by 1140 s. The original packet receives no further visual support, dying out by 1142 s.

F. Local View Cell Activity

The vision system learned 12 844 vision templates during the experiment, as shown in Fig. 14. The graph shows the active local view cells at 100 ms time intervals, with activation level indicated by point size and darkness (larger and darker meaning higher activation). Periods of no new template additions (manifested as a flattening of the graph line) indicate either times when the vehicle was driving through already learned streets, or times when the vehicle was stopped, such as at intersections or traffic lights.

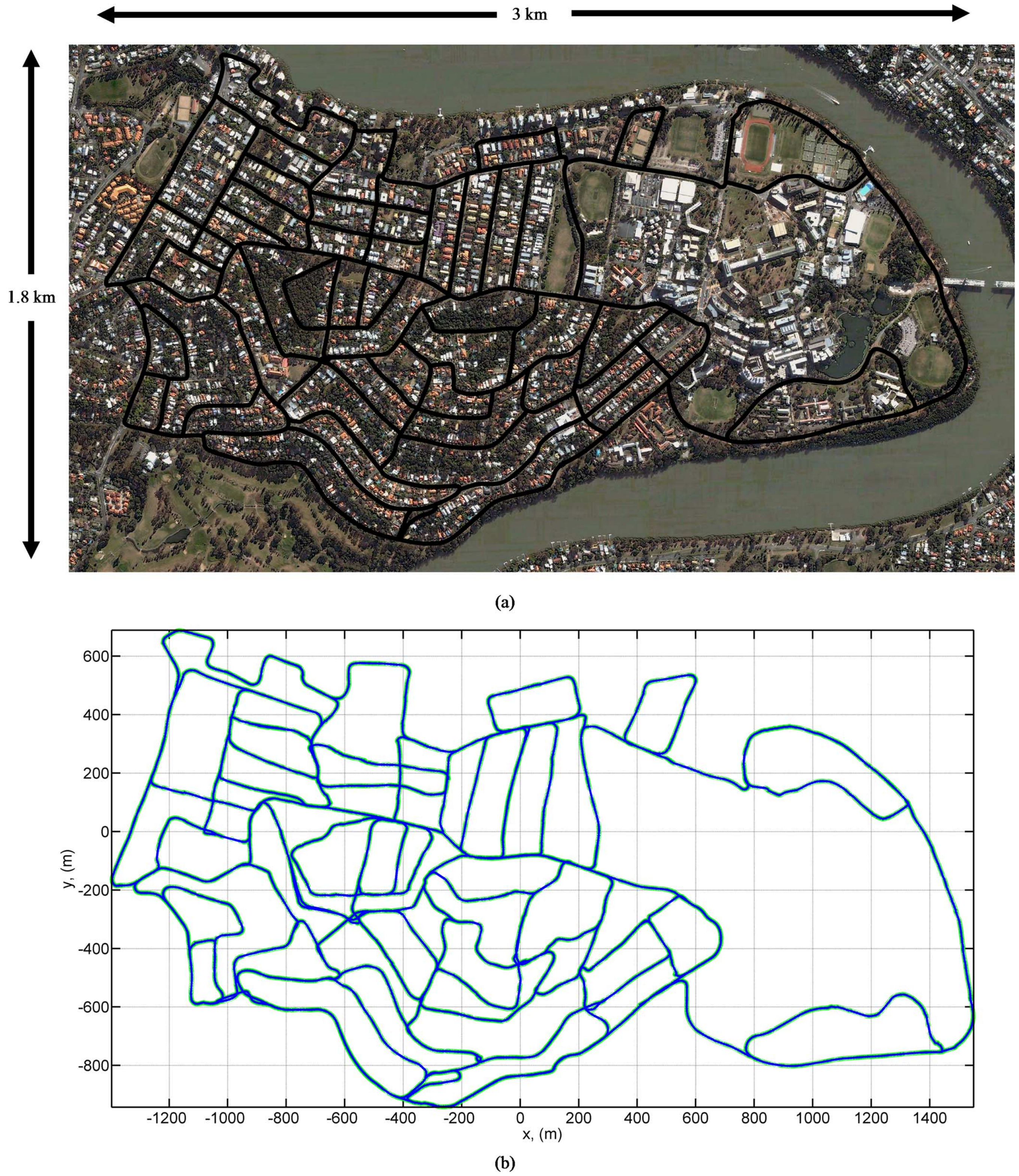


Fig. 9. (a) The test environment was the complete suburb of St Lucia, located in Brisbane, Australia. The area of road mapped measured approximately 3.0 km by 1.6 km. (b) Experience map produced by the mapping system. The map captures the overall layout of the road network as well as finer details such as curves and intersections.

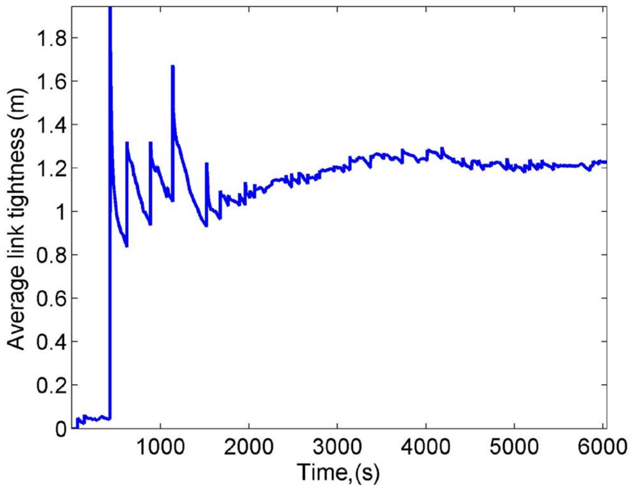
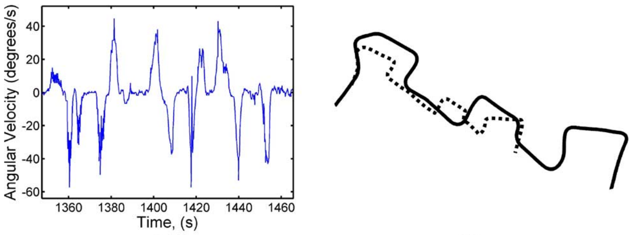


Fig. 10. Average link tightness over the experiment—the average discrepancy between the odometric information stored in the links between each experience and their relative positions. Spikes indicate the large loop closures that occur early in the experiment.



(a) (b)

Fig. 11. (a) Angular velocity calculated by the vision system for a section of winding road. (b) Vehicle trajectory as calculated by the vision odometry system (solid line) and the ground truth trajectory (dashed line). For this segment of road, the vision system captured vehicle rotation adequately, but overestimated vehicle velocity by about a factor of 1.6.

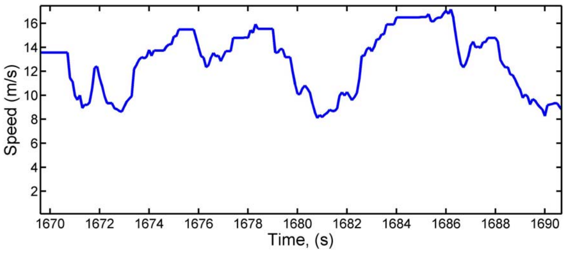
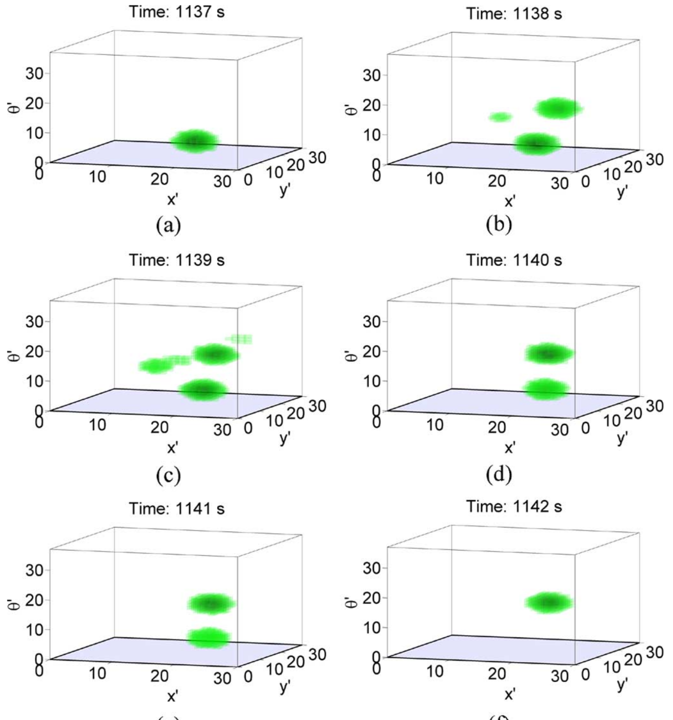


Fig. 12. Vehicle's translational speed, as calculated from the rate of image change for a section of road during which the vehicle traveled at a constant speed. The calculated speed varies by 40%, providing the mapping system with a noisy source of speed information.



(a) (b)

(c) (d)

(e) (f)

Fig. 13. Sequence of snapshots of the pose cell activity during a relocalization event. Over a period of 6 s, the vision system recognizes familiar visual scenes and injects activity into the pose cells, eventually creating a new dominant packet of activity.

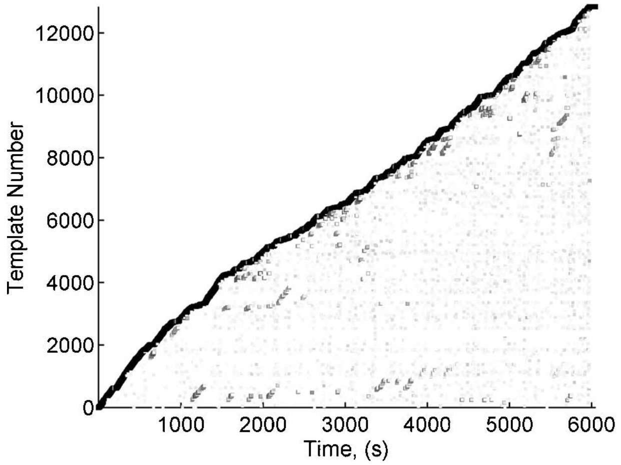


Fig. 14. Active local view cells plotted against time. Flattened sections of the graph represent periods of travel through already learned areas, or times when the vehicle was stopped. Higher activation levels are indicated by darker, larger points.

Fig. 15 shows a subsection of the local view cell plot corresponding to the vehicle's movement around the loop at the bottom right of the environment. During the 1.6 km loop, the vision system learned 415 templates, an average of one every 3.86 m. At 618 s, the vision system started recognizing familiar templates, indicating it was back at the start of the loop. During the subsequent second journey along part of the loop, it learned a further 22 new templates, while recognizing 175 previously learned templates. The average rate of recognition across the entire experiment was approximately 80%.

G. Vision System Ambiguity

To provide an insight into the visual ambiguity of the environment, Fig. 16 shows the activation levels of two local view cells plotted against the location in which they were active. Each local view cell is active in multiple locations, with one active

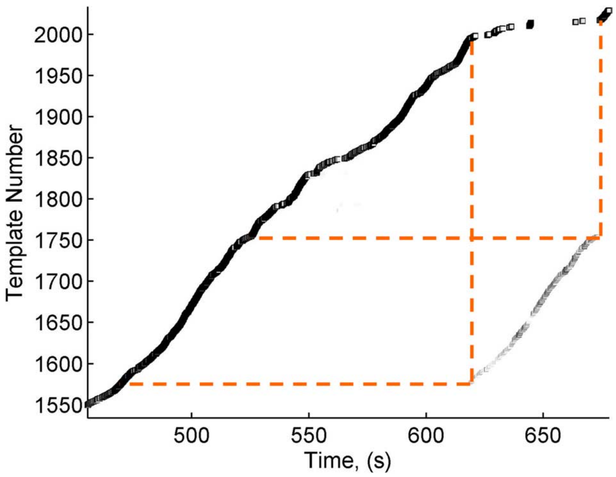


Fig. 15. Local view cell activity during a loop closure event.

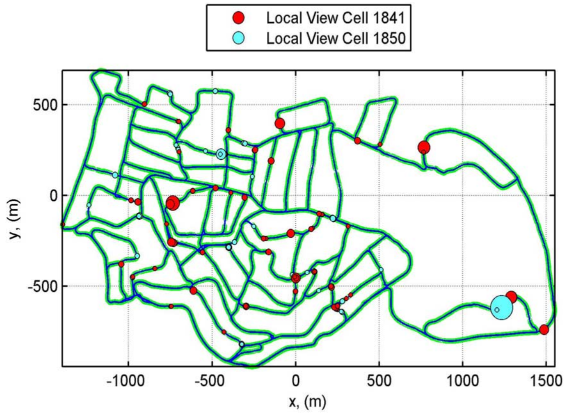


Fig. 16. Many different parts of the environment can appear similar to the vision system. In this plot, the locations in the environment where two local view cells were active are shown, with larger circles indicating higher levels of cell activation at that location.

in 43 distinct locations. The pose cells manage this degree of visual ambiguity by gradually building up pose estimates based on sequences of local view cell activation.

H. Pose Cell Activity

The wrapping connectivity of the pose cell structure, coupled with the large size of the environment, results in individual pose cells firing at multiple distinct locations in the environment. Fig. 17 shows the summed activation levels of the pose cells located in two different (x', y') positions within the pose cell structure, plotted against vehicle location in the environment. At each location in the environment where the cells fire, the activation level increased as the vehicle moved through the cell's firing field toward its center, and decreased as the vehicle left.

I. Global Localization

To test the ability of the system to localize itself within the map from an unknown starting position, 20 relocalization trials were run. The relocalization points were chosen by dividing the journey into 20 segments of 290 s each. This division selected

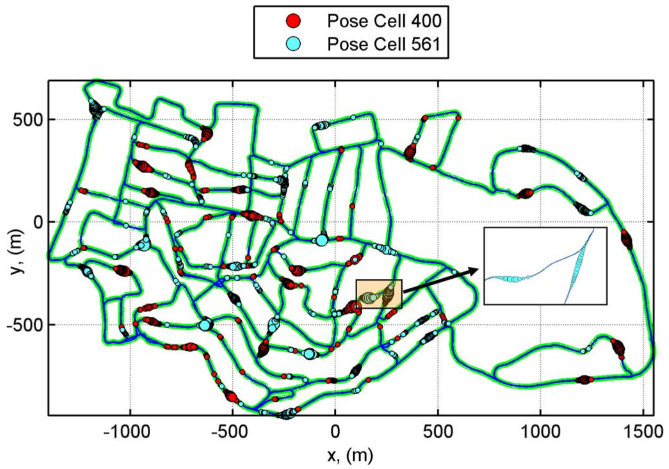


Fig. 17. Due to the wrapping connectivity of the pose cell matrix, each pose cell would fire at multiple locations in the environment. This figure shows the summed activation levels of the pose cells located in two different (x', y') positions within the pose cell structure (positions differentiated by color) plotted against vehicle location in the environment, with larger circles indicating a higher level of activation. The inset highlights two cell firing sequences for cells at one of the (x', y') positions, showing the cell activity level increasing, plateauing, and then, decreasing as the vehicle moved through two of the cells' firing fields.

a wide range of locations in the environment, and also, only included points in time when the vehicle was moving. The mapping system was initialized with a pose estimate of (0, 0, 0) and started at the beginning of each segment. The system was timed to see how long it would take to relocalize to the correct location within the experience map. Relocalization was defined as being achieved when the mapping algorithm selected the correct experience in the experience map.

Over the 20 trials, the average time taken to relocalize was 1.9 s (a maximum travel distance of 32 m), with a maximum relocalization time of 6.5 s and standard deviation of 1.2 s. The minimum relocalization time of 1.2 s (12 frames) represents the fastest relocalization possible in ideal visual conditions with a sequence of strongly activated local view cells all uniquely corresponding to the same environment location. There were no false loop closures. The trials demonstrated the system's ability to deal with perceptual ambiguity within this dataset.

J. Processing Requirements

The mapping system and vision processing module run at 10 Hz on a 2.4 GHz dual core Pentium processor (using only one of the cores). A typical breakdown of the processing time required at each frame at the end of the experiment when computational load was greatest (largest number of templates and experiences) is as given in Table V.

The experience map correction algorithm runs only after all other system modules have finished computation, and uses the remaining time, less a 5 ms buffer. The main use of computational resources in the vision system comes from the template matching algorithm. The current matching algorithm performs a preliminary search that ranks likely matching candidates based on their unrotated profiles, reducing the computation time by an

TABLE V
PROCESSING TIME

System Component	Processing Time
Graphical rendering	10 ms
Vision processing system	28 ms
Pose cell iteration	6 ms
Experience map correction	51 ms
Total	95 ms

order of magnitude, although the initial search time still scales with n, the number of templates. To scale real-time performance to a city-sized environment (100 km^2 and above) will require a selective correction algorithm for the experience map, and a search method for the visual templates that scales with $\log(n)$.

IX. DISCUSSION

This paper has presented a biologically inspired SLAM system known as RatSLAM, and demonstrated its use using only visual sensory information. The system is robust in visually ambiguous environments, and is fast enough to run online on a standard desktop computer or laptop while mapping a large outdoor environment. The mapping experiment demonstrated the system's ability to consistently close large loops with significant path integration errors, and to generate a coherent map by consolidating odometric information with map connectivity requirements.

It is clear that RatSLAM has some superficial similarities to probabilistic SLAM algorithms: the system tracks a belief with regard to pose during motion, and corrects that belief based on observations. However, beyond superficial similarity, RatSLAM has many marked contrasts to existing approaches to SLAM.

A. *Pose Cells*

The CAN that computes the pose cell activity contrasts strongly to the various forms of Bayesian inference engine found at the heart of existing SLAM algorithms. Without local view input, the pose cell activity packet always converges quickly over time to the same shape and activity level. There is little notion of variance in the width of the packet, or confidence in the activity of packet. During path integration, the width and activity level in the packet remain constant; there is no notion of increasing uncertainty in pose. Only during previously seen local view input is there any form of probabilistic representation, as multiple peaks caused by ambiguous visual scenes could be argued to represent multiple probabilistic beliefs. However, under attractor dynamics, there is constant pressure to converge to a single peak, which is not consistent with probabilistic forms of SLAM. Consequentially, RatSLAM behaves very differently from a particle filter during loop closure; where the particle filter relies on its motion model maintaining the particles necessary to close a loop (through the spreading particle cloud sampling the correct hypothesis), RatSLAM will typically have no active pose cells representing the correct vehicle location, at least initially. Instead, the previously seen local view input introduces the multiple pose hypotheses. RatSLAM also does not appear to suffer from the particle depletion problem that occurs with particle filters in nested loop situations [43], since traversing an inner loop has no effect on the system's ability to generate new pose hypotheses for the outer loop.

B. *Experience Map*

The experience map representation shows some apparent visual similarity to graphical SLAM techniques that have been developed in recent years [44]–[47]. The experiences and transitions do share some similarities with the pose nodes and motion arcs in algorithms such as GraphSLAM [46]. However, graphical SLAM techniques also link pose nodes with features that are commonly observable from multiple poses in the map, and include geometric measurement information. RatSLAM associates a single feature with each pose, without any geometric connection to the feature. Information storage and mapping processes are split between the experience map and the local view and pose cell networks, rather than being stored and processed probabilistically all in the one map. Furthermore, the experience map correction process is driven by fixed global map correction parameters, rather than by changing link constraint strengths. Experience transitions do not change strength, but the stiffness of the entire map does increase as it becomes more interlinked. For newly mapped sections of the environment, the map stiffness determines how much the existing map adjusts to incorporate the new section, and how much the section is adjusted to fit the existing map.

C. *Visual Features*

RatSLAM does not track visual features over multiple frames (apart from the frame to frame array matching for odometry), but instead processes the appearance of each frame individually. Appearance-based matching removes the complexity of feature detection, and allows the system to operate at relatively low frame rates and high movement speeds, with little overlap of features from frame to frame. However, the absence of visual features limits geometric interpretation of the environment. The map contains no geometric representations of landmarks, only the connectedness and approximate spatial relationship of places visited in the environment.

D. *Loop Closure*

Reliable loop closure is achieved by only matching experiences that have matching activity states of both the internal pose and local view cell. As a consequence, loop closure is not instantaneous, but rather occurs after a matching pose state is built up as familiar visual scenes activate pose cells. In the experiment, the system was able to rapidly close more than 51 loops, with no false loop closures, and was also able to globally relocalize to an arbitrary starting point in the map, usually within a couple of seconds. There is no difference in difficulty or process between closing a 400-m loop with a cumulative odometric error of 80 m, and a 5000-m loop with a cumulative error of 1200 m, both of which occur in the dataset. Odometric error is dealt with by combining robust loop closure detection with the experience mapping algorithm, which adjusts each loop to distribute odometric error throughout the map.

X. Conclusion

With both its similarities and many differences to other SLAM systems, RatSLAM provides an alternative approach to performing online, vision-only SLAM in large environments. The system not only benefits from some commonalities with traditional SLAM systems but also uses a number of differing approaches to achieve the same overall goal of creating useful maps of real world environments. These differences enabled the system to map an environment that would challenge existing state-of-the-art SLAM systems and to repeatedly and reliably close large loops in the face of significant odometric error and environmental ambiguity.

References

[1] M. Fyhn, S. Molden, M. P. Witter, E. I. Moser, and M.-B. Moser, "Spatial representation in the entorhinal cortex," *Science*, vol. 27, pp. 1258–1264, 2004.

[2] J. O'Keefe and D. H. Conway, "Hippocampal place units in the freely moving rat: Why they fire where they fire," *Exp. Brain Res.*, vol. 31, pp. 573–590, 1978.

[3] J. B. Ranck, Jr., "Head direction cells in the deep cell layer of dorsal presubiculum in freely moving rats," *Soc. Neurosci. Abstracts*, vol. 10, p. 599, 1984.

[4] M. Milford and G. Wyeth, "Featureless vehicle-based visual SLAM with a consumer camera," presented at the Australasian Conf. Robot. Autom., Brisbane, Australia, 2007.

[5] M. Milford and G. Wyeth, "Single camera vision-only SLAM on a suburban road network," presented at the Int. Conf. Robot. Autom., Pasadena, CA, 2008.

[6] M. J. Milford, G. F. Wyeth, and D. P. Prasser, "RatSLAM on the edge: Revealing a coherent representation from an overloaded rat brain," presented at the Int. Conf. Robots Intell. Syst., Beijing, China, 2006.

[7] S. Thrun, "Robotic mapping: A survey," in *Exploring Artificial Intelligence in the New Millennium*. San Mateo, CA: Morgan Kaufmann, 2002.

[8] S. Thrun, "Probabilistic algorithms in robotics," *AI Mag.*, vol. 21, pp. 93–109, 2000.

[9] G. Dissanayake, P. M. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localisation and map building (SLAM) problem," *IEEE Trans. Robot. Autom.*, vol. 17, no. 13, pp. 229–241, Jun. 2001.

[10] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," presented at the AAAI Nat. Conf. Artif. Intell., Edmonton, AB, Canada, 2002.

[11] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," presented at the Int. Joint Conf. Artif. Intell., Acapulco, Mexico, 2003.

[12] S. Thrun, "Exploration and model building in mobile robot domains," presented at the Int. Inst. Electr. Electron. Eng., San Francisco, CA, 1993.

[13] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[14] L. A. Clemente, A. J. Davison, I. D. Reid, J. Neira, and J. D. Tardos, "Mapping large loops with a single hand-held camera," presented at the Robot.: Sci. Syst., Atlanta, GA, 2007.

[15] M. Cummins and P. Newman, "Probabilistic appearance based navigation and loop closing," presented at the Int. Conf. Robot. Autom., Rome, Italy, 2007.

[16] H. Andreasson, T. Duckett, and A. Lilienthal, "Mini-SLAM: Minimalistic visual SLAM in large-scale environments based on a new interpretation of image similarity," presented at the Int. Conf. Robot. Autom., Rome, Italy, 2007.

[17] N. Karlsson, E. di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. E. Munich, "The vSLAM algorithm for robust localization and mapping," presented at the Int. Conf. Robot. Autom., Pasadena, CA, 2005.

[18] Z. Kira, *Evaluation of VSLAM in Outdoor Environments*. Atlanta, GA: Georgia Tech, 2004.

[19] 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," *J. Neurosci.*, vol. 10, pp. 420–435, 1990.

[20] J. S. Taube, R. U. Muller, and J. B. Ranck, Jr., "Head-direction cells recorded from the postsubiculum in freely moving rats. II. Effects of environmental manipulations," *J. Neurosci.*, vol. 10, pp. 436–447, 1990.

[21] 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.

[22] F. Sargolini, M. Fyhn, T. Hafting, B. L. Mcnaughton, M. Witter, M.-B. Moser, and E. I. Moser, "Conjunctive representation of position, direction, and velocity in entorhinal cortex," *Science*, vol. 312, pp. 758–762, 2006.

[23] J. Knierim, H. Kudrimoti, and B. McNaughton, "Place cells, head direction cells, and the learning of landmark stability," *J. Neurosci.*, vol. 15, pp. 1648–1659, 1995.

[24] D. Redish, "Through the Grid, a Window on Cognition," *Sci. American: Mind Matters*, 2007.

[25] D. Redish, A. Elga, and D. Touretzky, "A coupled attractor model of the rodent head direction system," *Netw.: Comput. Neural Syst.*, vol. 7, pp. 671–685, 1996.

[26] 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," *Netw.: Comput. Neural Syst.*, vol. 13, pp. 429–446, 2002.

[27] S. M. Stringer, T. P. Trappenberg, E. T. Rolls, and I. E. T. de Araujo, "Self-organizing continuous attractor networks and path integration: One-dimensional models of head direction cells," *Netw.: Comput. Neural Syst.*, vol. 13, pp. 217–242, 2002.

[28] K. Zhang, "Representation of spatial orientation by the intrinsic dynamics of the head-direction cell ensemble: A theory," *J. Neurosci.*, vol. 16, pp. 2112–2126, 1996.

[29] A. Samsonovich and B. L. McNaughton, "Path integration and cognitive mapping in a continuous attractor neural network model," *J. Neurosci.*, vol. 17, pp. 5900–5920, 1997.

[30] A. Arleo, *Spatial Learning and Navigation in Neuro-mimetic Systems: Modeling the Rat Hippocampus*. Germany: Verlag-dissertation, 2000.

[31] A. Arleo and W. Gerstner, "Spatial cognition and neuro-mimetic navigation: a model of hippocampal place cell activity," *Biol. Cybern.*, vol. 83, pp. 287–299, 2000.

[32] B. Browning, "Biologically plausible spatial navigation for a mobile robot," Ph.D. dissertation, Comput. Sci. Electr. Eng., Univ. Queensland, Brisbane, Australia, 2000.

[33] P. Gaussier, A. Revel, J. P. Banquet, and V. Babeau, "From view cells and place cells to cognitive map learning: processing stages of the hippocampal system," *Biol. Cybern.*, vol. 86, pp. 15–28, 2002.

[34] M. J. Milford, G. Wyeth, and D. Prasser, "RatSLAM: A hippocampal model for simultaneous localization and mapping," presented at the Int. Conf. Robot. Autom., New Orleans, LA, 2004.

[35] D. Prasser, M. Milford, and G. Wyeth, "Outdoor simultaneous localisation and mapping using RatSLAM," presented at the Int. Conf. Field Service Robot., Port Douglas, Australia, 2005.

[36] M. J. Milford, D. Prasser, and G. Wyeth, "Experience mapping: Producing spatially continuous environment representations using RatSLAM," presented at the Australasian Conf. Robot. Autom., Sydney, Australia, 2005.

[37] B. L. McNaughton, F. P. Battaglia, O. Jensen, E. I. Moser, and M. B. Moser, "Path-integration and the neural basis of the cognitive map," *Nature Rev. Neurosci.*, vol. 7, pp. 663–678, 2006.

[38] M. J. Milford, *Robot Navigation from Nature*. vol. 41, Berlin-Heidelberg: Springer-Verlag, 2008.

[39] J. Campbell, "A robust visual odometry and precipice detection system using consumer-grade monocular vision," presented at the Int. Conf. Robot. Autom., Barcelona, Spain, 2005.

[40] P. Newman, D. Cole, and K. Ho, "Outdoor SLAM using visual appearance and laser ranging," presented at the Int. Conf. Robot. Autom., Orlando, FL, 2006.

[41] D. Pomerleau, "Visibility estimation from a moving vehicle using the RALPH vision system," in *Proc. IEEE Conf. Intell. Transp. Syst.*, 1997, pp. 906–911.

[42] M. Bosse and J. Roberts, "Histogram matching and global initialization for laser-only SLAM in large unstructured environments," presented at the Int. Conf. Robot. Autom., Roma, Italy, 2007.

[43] C. Stachniss, G. Grisetti, and W. Burgard, "Recovering particle diversity in a Rao-Blackwellized particle filter for SLAM after actively closing loops," presented at the Int. Conf. Robot. Autom., Barcelona, Spain, 2005.

[44] J. Folkesson and H. Christensen, "Graphical SLAM—A self-correcting map," presented at the Int. Conf. Robot. Autom., New Orleans, LA, 2004.

[45] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," presented at the Int. Conf. Robot. Autom., Orlando, FL, 2006.

[46] S. Thrun and M. Montemerlo, "The GraphSLAM algorithm with applications to large-scale mapping of urban structures," *Int. J. Robot. Res.*, vol. 25, pp. 403–429, 2006.

[47] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Auton. Robots*, vol. 4, pp. 333–349, 1997.

Michael J. Milford (S'06–M'07) was born in Brisbane, Australia, in 1981. He received the B.E. degree in mechanical and space engineering and the Ph.D. degree in electrical engineering from the University of Queensland, Brisbane, in 2002 and 2006, respectively.

He is currently a Research Fellow with the Queensland Brain Institute and the School of Information Technology and Electrical Engineering, University of Queensland. His current research interests include biologically inspired robot mapping and navigation and the neural mechanisms underpinning navigation in animals.

Gordon F. Wyeth (M'03) received the B.E. and Ph.D. degrees in computer systems engineering from the University of Queensland, Brisbane, Australia, in 1989 and 1997, respectively.

He is currently a Senior Lecturer of information technology and electrical engineering at the School of Information Technology and Electrical Engineering, University of Queensland. He is the Co-Director of Mechatronic Engineering. His current research interests include biologically inspired robot systems, developmental robotics, and robot education.

Dr. Wyeth is a Chief Investigator on the Thinking Systems Project, i.e., a joint Australian Research Council/National Health and Medical Research Council (ARC/NHMRC) funded initiative, and has served as the Chief Investigator on several Australian Research Council and industry funded projects. He was the President of the Australian Robotics and Automation Association (2004–2006) and has twice chaired the Australasian Conference on Robotics and Automation. He was three times runner-up in the RoboCup small-size league and has served on various RoboCup committees. He developed RoboCup Junior in Australia and has twice chaired the Australian RoboCup Junior championships.