Single Camera Vision-Only SLAM on a Suburban Road Network

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

Abstract—**Simultaneous Localization And Mapping (SLAM) is one of the major challenges in mobile robotics. Probabilistic techniques using high-end range finding devices are well established in the field, but recent work has investigated vision-only approaches. This paper presents a method for generating approximate rotational and translation velocity information from a single vehicle-mounted consumer camera, without the computationally expensive process of tracking landmarks. The method is tested by employing it to provide the odometric and visual information for the RatSLAM system while mapping a complex suburban road network. RatSLAM generates a coherent map of the environment during an 18 km long trip through suburban traffic at speeds of up to 60 km/hr. This result demonstrates the potential of ground-based vision-only SLAM using low cost sensing and computational hardware.**

I. Introduction

SLAM (Simultaneous Localization And Mapping) is the problem that a robot, starting in an unknown environment, must learn a map of the environment, while simultaneously using that map to localize. In recent years many different solutions to the SLAM problem have been demonstrated, both in indoor and outdoor environments [1-6]. However, many of these mapping systems rely on accurate range-finding sensors, which are expensive and can be bulky, such as the well known *SICK* laser scanner.

Some of the recent work in this field has investigated the possibility of discarding range sensors and using only vision sensors [7-11]. Vision sensors are attractive for many reasons, such as their low cost, passive sensing, and compactness. Furthermore, humans and many animals appear to navigate successfully in complex environments using vision as their primary sensor, suggesting vision-only navigation can be effective. Some of the more promising visual mapping approaches have involved stereo camera setups [9] or the use of sophisticated algorithms which recover the 3D trajectory of an unconstrained camera through the environment [7, 8].

In this paper we focus on the problem of performing ground-based SLAM using a single consumer level camera, mounted on a road vehicle. The aim was to determine what mapping performance could be obtained without locating or tracking environmental features (such as trees or road edges) or image features (such as corners or SIFT features). We present methods for extracting a vehicle's angular velocity and an abstract representation of translational speed, without geometric interpretation of the environment, as well as a method for scene learning and recognition. The methods are demonstrated as a visual odometry and scene learning system for RatSLAM, a biologically inspired SLAM system [12, 13]. The system is tested experimentally on an 18 km car journey through a complex suburban road network.

The paper proceeds as follows. Section 2 briefly describes the SLAM system which was coupled with the vision system. Section 3 presents the vision system, including the methods for extraction of angular velocity, speed and the scene learning system. Section 4 describes the test environment and experimental procedure. The performance of each visual processing method is presented in Section 5, along with the map produced by the combined system, before the paper concludes in Section 6.

II. RatSLAM

Although it is not the focus of this paper, for the purposes of self-containment this section briefly presents the SLAM system, known as RatSLAM, which was coupled with the vision system. A more detailed description can be found in [14] and [13]. RatSLAM is the result of a series of performance-driven adaptations and extensions of computational models of mapping processes in the rodent brain [15-20].

Fig. 1 shows the core components of the RatSLAM system. The robot's pose is represented by activity in a competitive attractor neural network called the pose cells. Self-motion information is used to perform path integration by appropriately shifting the current pose cell activity. Activity can wrap in all three directions in the pose cell matrix. Vision information is converted into a local view

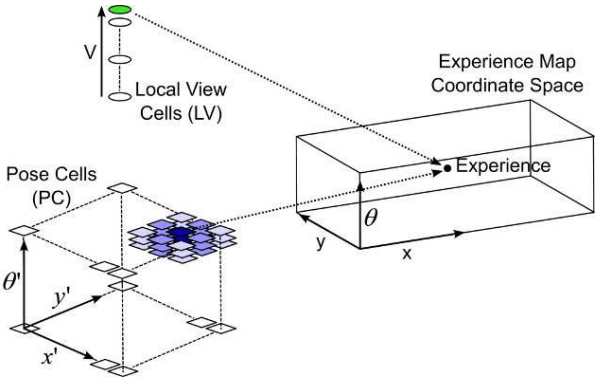


Fig. 1. An experience is associated with certain pose and local view cells, but exists within the experience map's own coordinate space.

Manuscript received September 14, 2007. This work was supported in part by an Australian Research Council Discovery Project Grant and National Health and Medical Research Council Grant.

M. J. Milford is with the Queensland Brain Institute and the School of Information Technology and Electrical Engineering, The University of Queensland, St Lucia, QLD 4072 Australia (phone: +61 7 3365 3770; fax: +61 7 3365 4999; 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, QLD 4072 Australia (e-mail: wyeth@itee.uq.edu.au)

(LV) representation (the image array templates described in Section III-c) that is associated with the currently active pose cells. If familiar, the current visual scene also causes activity to be injected into the particular pose cells associated with the currently active local view cells.

A. Experiences

The activity in the pose cells is converted into a usable map by an algorithm known as the *experience mapping* algorithm. The premise of the experience mapping algorithm is the creation and maintenance of a collection of experiences and inter-experience links. The algorithm creates experiences to represent certain states of activity in the pose cell and local view networks. The algorithm also learns behavioral, temporal, and spatial information in the form of inter-experience links. In effect, experiences represent distinct contextual memories of the environment, with the links representing the movement behavior and journey time required to move between experiences. The approach is similar to graphical SLAM techniques [21, 22], although it forms part of a mapping system, rather than a stand-alone mapping algorithm.

B. Experience Transitions

Inter-experience links store temporal, behavioural, and odometric information about the robot or vehicle movement between experiences. Fig. 2 shows a transition from experience i to experience j. The physical movement during this transition is given by:

$$d\mathbf{p}_{ij} = \mathbf{p}_j - \mathbf{p}_i = \begin{pmatrix} \theta_j \\ x_j \\ y_j \end{pmatrix} - \begin{pmatrix} \theta_i \\ x_i \\ y_i \end{pmatrix} = \begin{pmatrix} d\theta_{ij} \\ dx_{ij} \\ dy_{ij} \end{pmatrix} \quad (1)$$

where $d\mathbf{p}_{ij}$ is a vector describing the position and orientation of experience j relative to experience i. Repeated transitions between experiences result in an averaging of the odometric information:

$$d\mathbf{p}_{ij}^{new} = A.d\mathbf{p}_{ij}^{old} + B.d\mathbf{p}_{ij}^{curr} \quad (2)$$

where A and B are two geometric transform matrices, given in [13].

C. Map Correction

Discrepancies between a transition's odometric information and the linked experiences' (x, y, θ) coordinates are minimized through a process of map correction:

$$\Delta\mathbf{p}_i = \alpha\left[\sum_{j=1}^{N_f}\left(\mathbf{p}_j - \mathbf{p}_i - d\mathbf{p}_{ij}\right) + \sum_{k=1}^{N_t}\left(\mathbf{p}_k - \mathbf{p}_i - d\mathbf{p}_{ki}\right)\right] \quad (3)$$

where α is a learning rate constant, N_f is the number of links from experience i to other experiences, and N_t is the number of links from other experiences to experience i. Extensive experimentation in a range of environments has determined that an α value of 0.5 provides rapid map convergence without introducing instabilities.

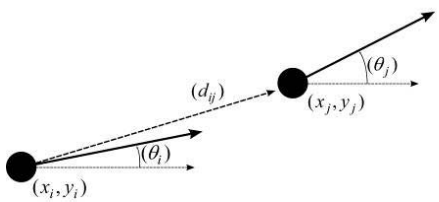


Fig. 2. Links between experiences store several types of information, including odometric information about the robot's movement during the transition.

III. Vision System

The camera used for this work was the built-in *iSight* camera on an Apple *Macbook* notebook computer (Fig. 3). The built-in iSight is similar to the more common external Apple iSight cameras, but uses a USB 2.0 rather than FireWire interface, is fixed-focus and uses an active pixel sensor rather than charge-coupled device (CCD). The camera's resolution is 640×480 pixels and it is capable of 30 frames per second in 24 bit color. The use of this particular camera was motivated by its obvious low cost, demonstrating that the system can function without precision components and can consequently be readily deployed in a range of applications.

Images were captured at an effective frame rate of 8.0 frames per second. The colour images (Fig. 4a) were first converted to greyscale images, before being cropped to a 300×160 pixel sub window (Fig. 4b). Each pixel column was then summed and normalized to form a one-dimensional array (Fig. 4c). Cropping the image removes much of the ground plane and increases the geometric validity of summing pixel columns. These image arrays formed the basic abstract image representation from which vehicle rotation and speed was extracted. They were also used as the basis for the image template learning component.

Fig. 3. Built-in iSight video camera on an Apple Macbook. This camera was the sole source of sensory information for all experiments. The laptop was mounted on the roof of a car in a forward facing, neutral pitch position.

A. Extracting Rotation

Rotation information is extracted by comparing consecutive image arrays. Fig. 5a-b shows two consecutive

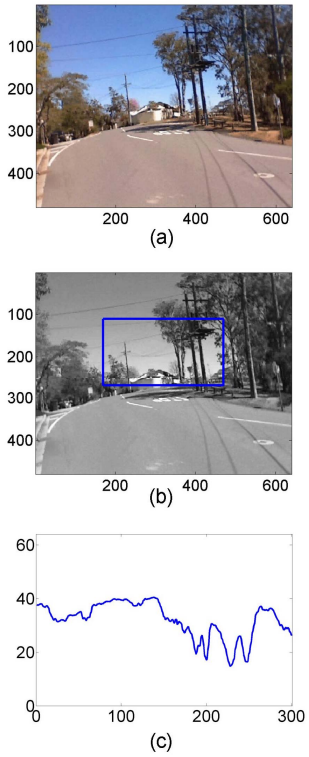


Fig. 4. Image processing stages. The original 640×480 pixel color image (a) is converted to grayscale (b), before being cropped to a 300×160 pixel image, and then converted into a column intensity graph (c).

images, with their associated image arrays shown in Fig. 5c. The comparison between images is performed by calculating the average absolute intensity difference between the two image arrays, *f(s)*, as they are shifted relative to each other:

$$f(s)=\frac{1}{w-|s|}\left(\sum_{n=1}^{w-|s|}\left|I^{k+1}_{n+\max(s,0)}-I^{k}_{n-\min(s,0)}\right|\right) \qquad (4)$$

where I is the image array intensity values of the k^{th} and $k^{th}+1$ images, s is the image array shift, and w is the image width. Fig. 5d shows the average image array intensity differences for shifts of the first image array (dotted line). The best match for these two images is obtained for a shift of about 70 pixels to the right. The pixel shift is multiplied by an empirically determined gain constant, λ, to convert it into an approximate angular shift $\Delta\theta$:

$$\Delta\theta=\lambda(\arg\min f(s)) \qquad (5)$$

To ensure that there was sufficient overlap between images, $\Delta\theta$ was only calculated for $|s| < w - 30$.

The rotation calculation relies on a few assumptions, first and foremost that the camera is forward facing. The camera platform must also be constrained in its movement like a car or wheelchair style robot – the system cannot handle translation parallel to the camera lens plane. In addition, part of the reason for cropping the raw camera images is to reduce the effective field of view of the camera. A small field of view in a forward facing camera reduces the effect on image change of proximal walls in narrow environments

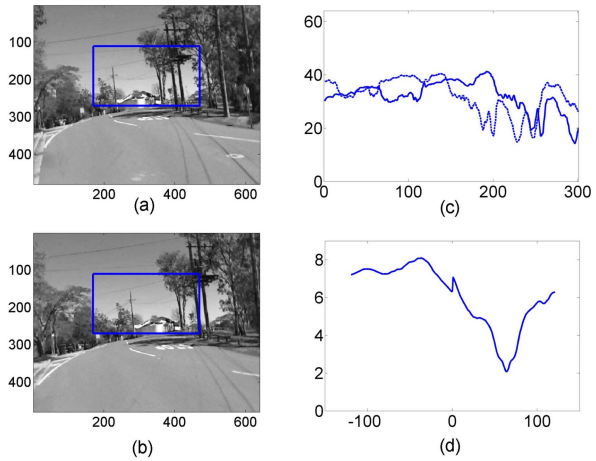


Fig. 5. Rotation information was calculated by comparing consecutive image arrays and calculating the pixel shift of the best match. (a) First image. (b) Second image. (c) Image arrays corresponding to (a) (dotted line) and (b) (solid line). (d) Graph showing adjusted image array differences for shifts in their relative positions. The best match occurs for an image 1 shift of about 70 pixels to the right.

such as corridors. In such situations, traveling along a corridor more closely to one wall than the other introduces the extra challenge of extremely different rates of change in the left and right side of the image, even though the camera is moving in a straight line. An alternative solution would be to use a bee-like optically driven centre-line following movement behavior, or an iterative estimation process for translation and rotation speeds.

B. Extracting Speed

Extracting absolute speed from a single camera without any initialization, known landmark sizes or camera elevation information is a very difficult, if not impossible challenge. The speed extraction system presented in this paper was loosely inspired instead by how bees use optical flow to perform path integration. 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 quite representative of the environment.

The rate of image change v is obtained by calculating the average image array intensity differences for the best rotation match s_m of the current and last image:

$$v=\frac{1}{w-|s_m|}\left(\sum_{n=1}^{w-|s_m|}\left|I^{k+1}_{n+\max(s_m,0)}-I^{k}_{n-\min(s_m,0)}\right|\right) \qquad (6)$$

where

$$s_m=\arg\min f(s) \qquad (7)$$

By calculating the image difference using the best matched image arrays, the effect of rotation is mostly removed from the speed calculation.

C. Template Learning

Any path integration process, whether based on wheel encoder counts or optical flow, is subject to the accumulation of error over time. To overcome this limitation, a navigation system must be able to recognize familiar places using its sensory information. To achieve this capability, we use the image arrays as the basis for a visual template learning system. Images that are deemed sufficiently novel are added to the system's repository of stored image array templates.

Each new image is converted into an image array as described at the start of Section III. This image array I is then compared with all the image array templates I_k stored in the repository, to yield a vector of array differences $f(k)$:

$$f(k) = \frac{1}{w-|s|}\left(\sum_{n=1}^{w-|s|} \left| I_{n+\max(s,0)} - I^{k}_{n-\min(s,0)} \right| \right) \quad (8)$$

If the minimum difference exceeds a threshold value, the new image is added to the repository. Otherwise, the best match existing image array is used as the current template. The s range can be varied depending on the desired rotational generalisation of the system.

IV. Experimental Setup

The visual processing techniques described in Section III were tested in a large suburban road network, shown in Fig. 6. This environment contains approximately 9 km of road, and is quite irregular in overall layout. Elevation varies by 50 m. The road network consists of many loops of varying size and shape, and includes 24 intersections. A Macbook laptop with built-in iSight camera was mounted on the roof of a car, facing forwards and with neutral pitch. The car was driven around the road network such that every street was visited at least once, and most were visited multiple times. Total driving time was just over 30 minutes; the vehicle travelled a distance of approximately 18 km, at speeds of up to 60 km/hr. Images were obtained at a rate of 8 frames per second from the camera, saved to disk, and then replayed to the vision module at real-time speed, combined with the RatSLAM system. Computation was performed on a standard desktop PC, and could be run online on the notebook. The experiment was performed in the late morning of a weekday. Light traffic was encountered throughout the journey, and lighting conditions varied.

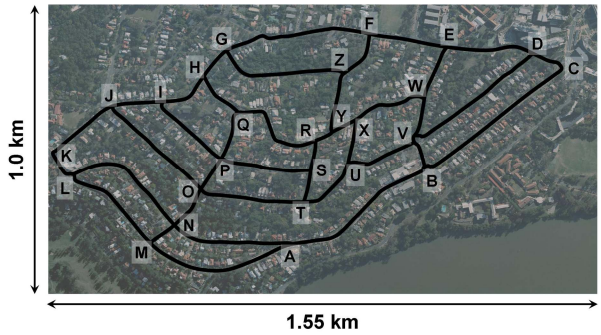


Fig. 6. The test environment was a complex suburban road network. The vehicle started at A and visited every road at least once and many multiple times, traveling 18 km over half an hour, at speeds of up to 60 km/hr. Image © Sensis Pty Ltd.

V. Results

This section presents the performance of the template matching process, the angular velocity and speed extraction methods, and the overall system's mapping performance in the test environment. Angular velocity and speed calculations are shown for illustrative sections of the environment.

A. Angular Velocity Extraction

Fig. 7 shows the angular velocities of the vehicle as calculated by the image array matching algorithm for two sections of road. Fig. 7b shows the estimated vehicle angular velocity over a period of 50 seconds corresponding to the vehicle movement shown in Fig. 7a. A moderate right turn and sharp left turn are clearly represented by a small negative angular velocity peak and large positive angular velocity peak. Fig. 7c shows the trajectory of the vehicle as calculated using only visual odometry.

The estimated angular velocities as the car travelled through the roundabout shown in Fig. 7d are shown in Fig. 7e. The two periods of small positive angular velocities correspond to the entry and exit of the car from the roundabout. The period of larger negative angular velocities between them corresponds to the car rounding the central circular island in the roundabout. Fig. 7f shows the trajectory of the vehicle as calculated using only visual odometry.

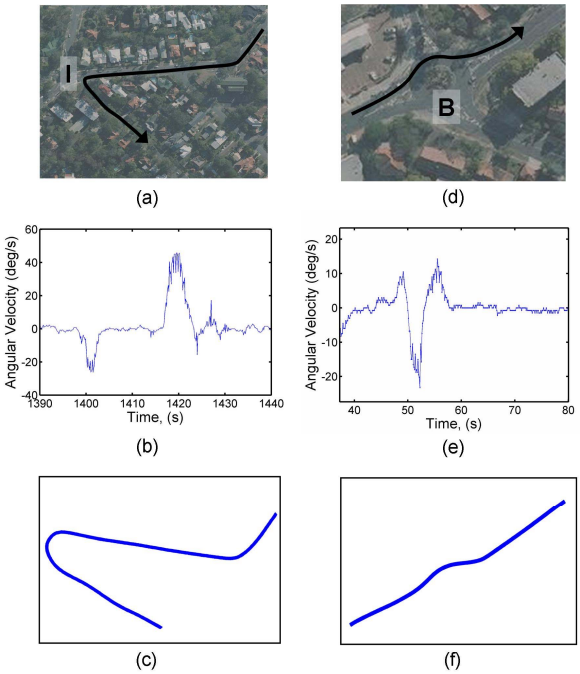
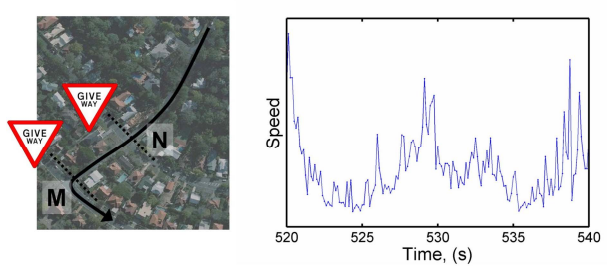


Fig. 7. Unfiltered vehicle angular velocity calculated from image change rates. (a) Vehicle trajectory through a series of two turns and (b) corresponding calculated velocities. (c) Calculated vehicle trajectory using only visual odometry. (d) Vehicle trajectory through a roundabout and (e) corresponding calculated angular velocities. (f) Calculated vehicle trajectory using only visual odometry.



(a) (b)

Fig. 8. (a) Vehicle trajectory through two give way signs and (b) the unfiltered calculated vehicle 'speed', showing the vehicle slowing before each intersection.

B. Speed Extraction

The 'speed' of the camera through road section ONM as calculated by the vision system is shown in Fig. 8b. No units are shown along the vertical axis, although in strict terms the speed is measured in terms of the average difference between image array intensity values for consecutive images. The speed signal is noisier than the angular velocity signal shown in Fig. 7b and Fig. 7e, but the deceleration before each intersection can clearly be seen.

C. Template Learning and Recall

Fig. 9 shows the performance of the template learning and recall system. During the half hour experiment, the vision system learned 3000 templates (Fig. 9a). The system was able to recognize familiar sections of the environment, as shown during the time periods 310 to 350 s and 400 to 470 s in Fig. 9b. During repeated traverses of familiar road sections, very few new templates were learned, as shown by the periodic 'flattening' of the graph in Fig. 9a. There were a couple of occasions when the vision system displayed poor recognition of a familiar road section; these were caused in one case by the appearance of a large truck in a narrow road section, and in the other case by sunlight reflecting from a group of parked cars directly into the camera lens.

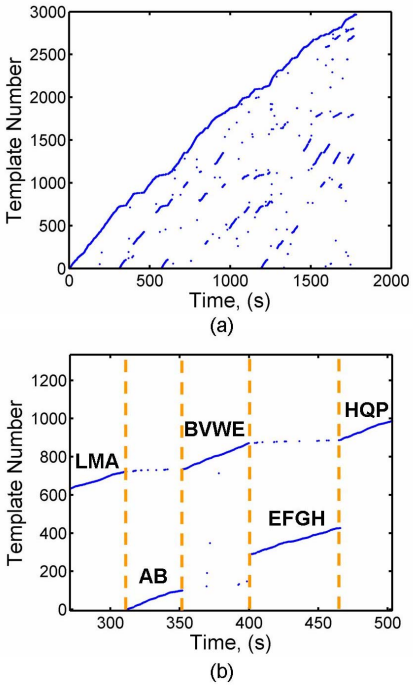


Fig. 9. Visual template learning and recognition over the (a) entire experiment and (b) for the section LMABVWEFGHQP. During this section the vehicle is passing through sections AB and EFGH for the second time. The vision system is able to recall the visual templates learned during the first pass.

D. Mapping

The experience map created by the RatSLAM system and experience mapping algorithm is shown in Fig. 10b. For reference, the ground truth trajectory of the robot through the environment is also shown. The ground truth trajectory was obtained by tracing the path of the road through the aerial photo shown in Fig. 6. The accompanying video shows the entire visual sequence and the simultaneous construction of the experience map.

The experience map closely resembles the actual path of the camera through the environment, although it is not identical. In particular, parts of the map are warped. The warping is due to the lack of an absolute measure of speed – in narrow, visually rich sections of road the appearance of the environment changes more quickly for a given vehicle speed than in open, visually bland areas. For example, road section RQ is slightly stretched in the experience map because the vehicle was traveling through a narrow lane with a high perceptual speed.

VI. Conclusion

The results presented in this paper have demonstrated the potential for mapping ground-based environments with only

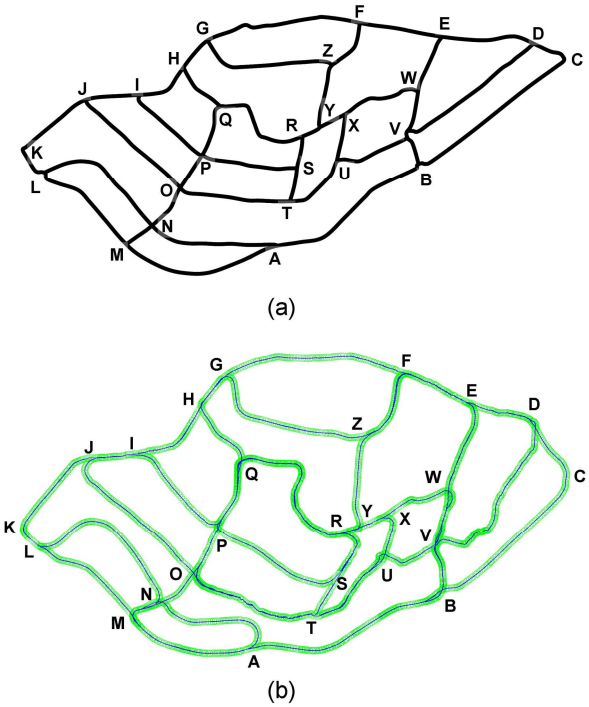


Fig. 10. (a) Ground truth trajectory and (b) corresponding experience map. The map is topologically correct, although warped in places, due to the use of perceptual rather than absolute speed in path integration.

a single, consumer camera, without geometric interpretation of the environment. It appears that, as expected, quite good rotational information can be extracted from visual sequences with a forward facing, neutral pitch camera. In the short term it also seems possible to learn and recognize visual scenes in an outdoor environment, although dynamic objects such as cars can disrupt performance. Extracting translation speed was more challenging, and only an abstract representation of speed was obtained. However, the mapping system was still able to generate a coherent and representative map.

A better measure of translational velocity may be obtained by employing some form of ground texture or feature tracking. Having a measure of physical speed will probably produce maps that more closely resemble the physical layout of the environment. However, it may turn out that using a perceptual speed measure yields maps that are more useful for a robot performing autonomous navigation. In previous work using the RatSLAM system, navigation performance did not seem to be affected by warping of the map [13]. It is interesting to note that animals such as bees may not have a means of measuring absolute speed [23], yet still navigate successfully in large environments. Although it could not be tested in this case, based on past experience [13], it is likely that the generated experience map could be used for effective navigation.

A. Future Work

By extending the vision system to use multiple cameras or a panoramic camera, it will be possible to explicitly link trajectories in opposing directions such as along a road or corridor. The crude summation of vertical pixel columns could be replaced by a processing step that takes into account the camera mounting geometry and optical characteristics. Methods for handling dynamic objects such as cars, and day-night lighting changes, are currently under investigation. Longer experiments, in larger, more challenging environments are to be performed, including active navigation experiments. These experiments will determine the scalability in space and time of the approach to visual SLAM presented in this paper.

REFERENCES

[1] 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 Transactions on Robotics and Automation*, vol. 17, pp. 229-241, 2001.

[2] B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, and F. Savelli, "Local Metrical and Global Topological Maps in the Hybrid Spatial Semantic Hierarchy," presented at the International Conference on Robotics and Automation, New Orleans, USA, 2004.

[3] 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 International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 2003.

[4] P. M. Newman and J. J. Leonard, "Consistent, Convergent, and Constant-time SLAM," presented at the International Joint Conference on Artificial Intelligence, Acapulco, Mexico, 2003.

[5] S. Thrun, "Probabilistic Algorithms in Robotics," in *AI Magazine*, vol. 21. Pittsburgh: Carnegie Mellon University, 2000, pp. 93-109.

[6] G. Grisetti, C. Stachniss, and W. Burgard, "Improving Grid Based SLAM with Rao Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling," presented at the International Conference on Robotics and Automation, Barcelona, Spain, 2005.

[7] 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 Robotics: Science and Systems, Atlanta, United States, 2007.

[8] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1052-1067, 2007.

[9] J. M. Porta, J. J. Verbeek, and B. Krose, "Active Appearance-Based Robot Localization Using Stereo Vision," *Autonomous Robots*, vol. 18, pp. 59-80, 2005.

[10] R. Sim, P. Elinas, M. Griffin, and J. J. Little, "Vision-based SLAM using the Rao-Blackwellised Particle Filter," presented at the International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, 2005.

[11] N. Cuperlier, M. Quoy, P. Gaussier, and C. Giovanangelli, "Navigation and planning in an unknown environment using vision and a cognitive map," presented at the International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, 2005.

[12] M. J. Milford, *Robot Navigation From Nature*. Berlin-Heidelberg: Springer-Verlag, 2008, in press.

[13] 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 International Conference on Robots and Intelligent Systems, Beijing, China, 2006.

[14] 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.

[15] 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.

[16] 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.

[17] 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," *Network: Computation in Neural Systems*, vol. 13, pp. 217-242, 2002.

[18] B. Browning, "Biologically Plausible Spatial Navigation for a Mobile Robot," PhD, Computer Science and Electrical Engineering, University of Queensland, Brisbane, 2000.

[19] D. Redish, A. Elga, and D. Touretzky, "A coupled attractor model of the rodent head direction system," *Network: Computation in Neural Systems*, vol. 7, pp. 671-685, 1996.

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

[21] J. Folkesson and H. Christensen, "Graphical SLAM - a self-correcting map," presented at the International Conference on Robotics and Automation, New Orleans, United States, 2004.

[22] S. Thrun and M. Montemerlo, "The GraphSLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures," *The International Journal of Robotics Research*, vol. 25, pp. 403-429, 2006.

[23] M. V. Srinivasan, S. Zhang, M. Altwein, and J. Tautz, "Honeybee Navigation: Nature and Calibration of the "Odometer"," *Science*, vol. 287, pp. 851-853, 2000.