Featureless Vehicle-Based Visual SLAM with a Consumer Camera

Michael Milford[*†], Gordon Wyeth[†]
[*]Queensland Brain Institute
[†]School of Information Technology and Electrical Engineering
The University of Queensland
St Lucia Queensland 4072 Australia
{milford, wyeth@} itee.uq.edu.au

Abstract

The Simultaneous Localisation And Mapping (SLAM) problem 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. We present an alternative approach to the leading existing techniques, which extracts approximate rotational and translation velocity information from a vehicle-mounted consumer camera, without tracking landmarks. When coupled with an existing SLAM system, the vision module is able to map a 45 metre long indoor loop and a 1.6 km long outdoor road loop, without any parameter or system adjustment between tests. The work serves as a promising pilot study into ground-based vision-only SLAM, with minimal geometric interpretation of the environment.

1 Introduction

One of the major problems facing autonomous mobile robots is the Simultaneous Localisation And Mapping (SLAM) problem. The core SLAM problem is the requirement that a robot, starting in an unknown environment, explore in order to learn the environment (map), while simultaneously using that map to keep track of the robot's position (localise) within the environment. SLAM is, strictly speaking, a problem, although throughout the literature it is also used as a description of a process that a robot performs in solving the problem [Thrun, 2002].

There has been extensive research into the SLAM problem over the past two decades. In recent years many different solutions to the SLAM problem have been demonstrated both in indoor and outdoor environments [Dissanayake *et al.,* 2001, Kuipers *et al.*, 2004, Montemerlo *et al.*, 2002, Montemerlo *et al.*, 2003, Newman *et al.*, 2003, Thrun, 2000, Grisetti *et al.*, 2005]. 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 [Clemente *et al.*, 2007, Davison *et al.*, 2007, Davison *et al.*, 2005, Porta *et al.*, 2005, Sim *et al.*, 2005, Cuperlier *et al.*, 2005]. Vision sensors are attractive for many reasons, such as their low cost, passive sensing, and compactness. Furthermore, there is the ever present reality that humans and many animals appear to navigate effectively in large and complex environments using vision as their primary sensor. Some of the more promising results have involved stereo camera setups [Porta *et al.*, 2005] or the use of sophisticated algorithms which recover the 3D trajectory of an unconstrained camera through the environment [Clemente *et al.*, 2007, Davison *et al.*, 2007, Davison *et al.*, 2005].

In this paper we focus on the simpler problem of performing ground-based SLAM using a single consumer level camera, mounted on a vehicle constrained in its movement by its wheel arrangement (i.e. a car). The aim was to determine what mapping performance could be obtained from a relatively simple and non-environment specific (i.e. not looking for trees or doorways) image processing regime if coupled with an already competent SLAM algorithm [Milford, 2008, in press, Milford *et al.*, 2006]. More specifically, in this paper we present straightforward methods for extracting a vehicle's angular velocity and an abstract representation of translational speed in order to perform path integration. In addition, we present a scene learning and recognition method. These vision processing techniques are integrated with the existing SLAM system and tested experimentally over a 45 metre long loop of an indoor environment and a 1.6 km long loop of an outdoor environment.

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

2 Vision System

The camera used for this work was the built-in *iSight* camera on an Apple *Macbook* notebook computer (Figure 1). 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 colour. The use of this particular camera was motivated by its status as a cheap consumer rather than high-end camera, and also because of its impressive autoexposure capabilities in outdoor environments with extreme variations in illumination.

Figure 1 – Built-in *iSight* video camera on Apple *Macbook*. This camera was the sole source of sensory information for all experiments. The laptop was mounted on a chair for the indoor experiments and on a car for the outdoor experiments.

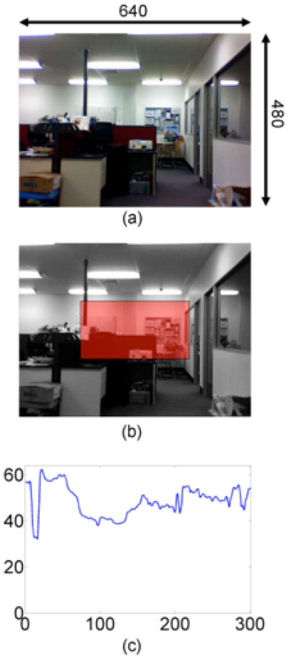


Figure 2 – Image processing stages. The original colour image (a) is converted to grayscale (b), before being cropped and converted into a column intensity graph (c).

Images were captured at 25 frames per second, but 2 in every 3 frames were dropped, resulting in a frame rate of 8.3 frames per second. The colour images (Figure 2a) were first converted to greyscale images, before being cropped to a 300×160 pixel sub window (Figure 2b). Each pixel column was then summed and normalized to form a one-dimensional array (Figure 2c). Cropping the image removes much of the ground plane and increases the geometric relevance 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.

2.1 Extracting Rotation

Rotation information is extracted by comparing consecutive image arrays. Figure 3a-b shows two consecutive images and their associated image arrays (Figure 3c). 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:

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

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. Figure 3d 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 30 pixels to the left. The pixel shift is multiplied by an empirically determined gain constant α to convert it into an approximate angular shift $\Delta\theta$:

$$\Delta\theta = \alpha\left(\arg\min f(s)\right) \quad (2)$$

To ensure that there was sufficient overlap between images, $\Delta\theta$ was only calculated for $|s| < w - 10$.

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 corridors. In such situations, travelling 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 behaviour, or an iterative estimation process for translation and rotation speeds.

2.2 Extracting Speed

Extracting absolute speed from a single camera without any initialization, known landmark sizes or camera elevation information is an impossible task. The speed extraction system presented in this paper was loosely inspired instead by how bees use optical flow to perform path integration. Speeds are 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 simple 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:

$$v = \frac{1}{w-|s_m|}\left(\sum_{n=1}^{w-|s_m|}\left|I^{k+1}_{n+\max(s_m,0)} - I^{k}_{n-\min(s_m,0)}\right|\right) \qquad (3)$$

where

$$s_m = \arg\min f(s) \qquad (4)$$

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

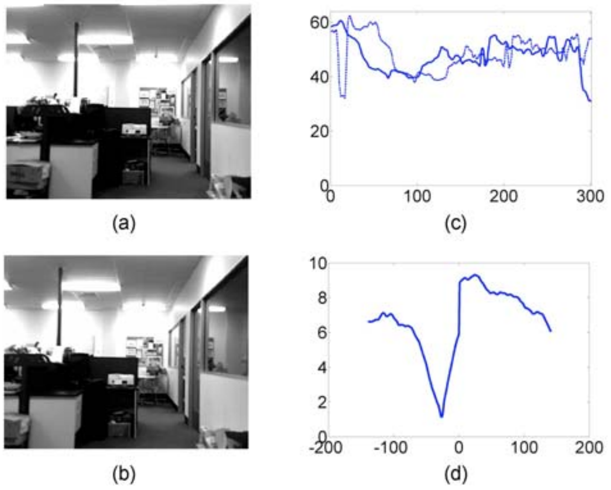


Figure 3 – Rotation information was calculated by comparing consecutive image arrays and calculating the pixel shift of the best match. (a) First image. (b) Second image. (c) Image arrays corresponding to (a) (dotted line) and (b) (solid line). (d) Graph showing adjusted image array differences for shifts in their relative positions. The best match occurs for an image 1 shift of about 30 pixels to the left.

2.3 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 2. 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 $f(k)$:

$$f(k) = \frac{1}{w-|s|}\left(\sum_{n=1}^{w-|s|}\left|I_{n+\max(s,0)} - I^{k}_{n-\min(s,0)}\right|\right) \qquad (5)$$

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.

3 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 [Milford *et al.*, 2004] and [Milford *et al.*, 2006].

Figure 4 shows the core structure of the RatSLAM system. The robot's pose is represented by activity in a competitive attractor neural network called the pose cells. Wheel encoder information is used to perform path integration by appropriately shifting the current pose cell activity. Activity can wrap in all three directions in the pose cell matrix. Vision information is converted into a local view (LV) representation (the image array templates) 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.

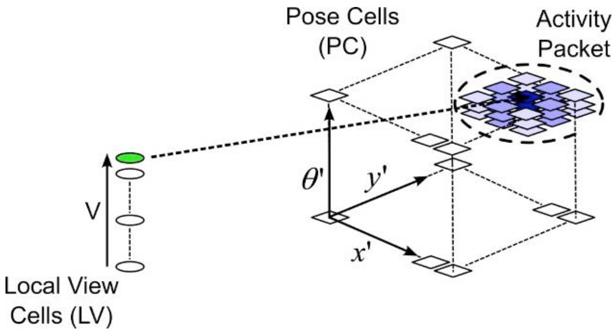


Figure 4 - The core RatSLAM pose cell and local view cell networks.

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 behavioural, temporal, and spatial information in the form of inter-experience links. In effect, experiences represent distinct contextual memories of the environment. Figure 5 shows the relationship between the experience map and the core RatSLAM representations.

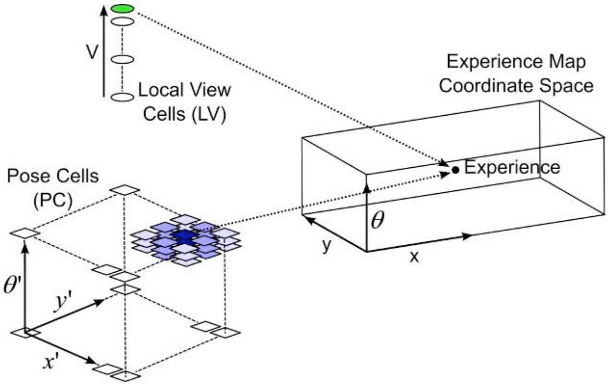


Figure 5 - An experience is associated with certain pose and local view cells, but exists within the experience map's own coordinate space.

3.1 Experiences

Experiences have an activity level that is dependent on how close the activity peaks in the pose cells and local view cells are to the cells associated with the experience. The component of activity determined by the pose cell network activity, $E_{x'y'\theta'}$, is given by:

$$E_{x'y'\theta'} = \begin{cases} 0 & \text{if } r_r' > 1; \\ 0 & \text{if } \theta_r' > 1; \\ 2 - r_r' - \theta_r' & \text{otherwise} \end{cases} \qquad (6)$$

$$r_r' = \frac{\sqrt{\left(x_{pc}' - x_i'\right)^2 + \left(y_{pc}' - y_i'\right)^2}}{r_a} \qquad (7)$$

$$\theta_r' = \frac{\left|\theta_{pc}' - \theta_i'\right|}{\theta_a} \qquad (8)$$

where x'_{pc}, y'_{pc}, and θ'_{pc} are the coordinates in the pose cell matrix of the dominant activity packet, x'_i, y'_i, and θ'_i are the coordinates of the pose cells associated with experience *i*, r_a is the zone constant for the (x', y') plane, and θ_a is the zone constant for the θ' dimension. The visual scene V_i switches an experience on or off:

$$E_i = \begin{cases} 0 & \text{if } V_{curr} \neq V_i; \\ E_{x'y'\theta'} & \text{if } V_{curr} = V_i \end{cases} \qquad (9)$$

where V_{curr} is the current visual scene, and V_i is the visual scene associated with experience *i*. The most active experience is known as the peak experience. Learning of new experiences is triggered by the peak experience's activity level dropping below a threshold value.

3.2 Experience Transitions

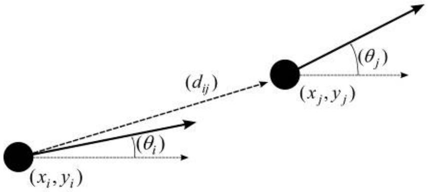


Figure 6 - Links between experiences store several types of information, including odometric information about the robot's movement during the transition.

Inter-experience links store temporal, behavioural, and odometric information about the robot's movement between experiences. Figure 6 shows a transition from experience *i* to experience *j*. The physical movement of the robot during this transition is given by:

$$d\mathbf{p}_{ij} = \mathbf{p}_j - \mathbf{p}_i = \begin{pmatrix} \theta_j \\ x_j \\ y_j \end{pmatrix} - \begin{pmatrix} \theta_i \\ x_i \\ y_i \end{pmatrix} = \begin{pmatrix} d\theta_{ij} \\ dx_{ij} \\ dy_{ij} \end{pmatrix} \qquad (10)$$

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 [Milford, 2008, in press]:

$$d\mathbf{p}_{ij}^{new} = A.d\mathbf{p}_{ij}^{old} + B.d\mathbf{p}_{ij}^{curr} \qquad (11)$$

where $$A = \begin{pmatrix} 1/2 & 0 & 0 \\ 0 & d_s.\cos d\theta & -d_s.\sin d\theta \\ 0 & d_s.\sin d\theta & d_s.\cos d\theta \end{pmatrix},$$

$$B = \begin{pmatrix} 1/2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$d\theta = \frac{1}{2}\left[\tan^{-1}\left(\frac{dy_{ij}^{curr}}{dx_{ij}^{curr}}\right) - \tan^{-1}\left(\frac{dy_{ij}^{old}}{dx_{ij}^{old}}\right)\right], \text{ and}$$

$$d_s = \left(d_{ij}^{curr} + d_{ij}^{old}\right) / \left(2d_{ij}^{old}\right).$$

3.3 Map Correction

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

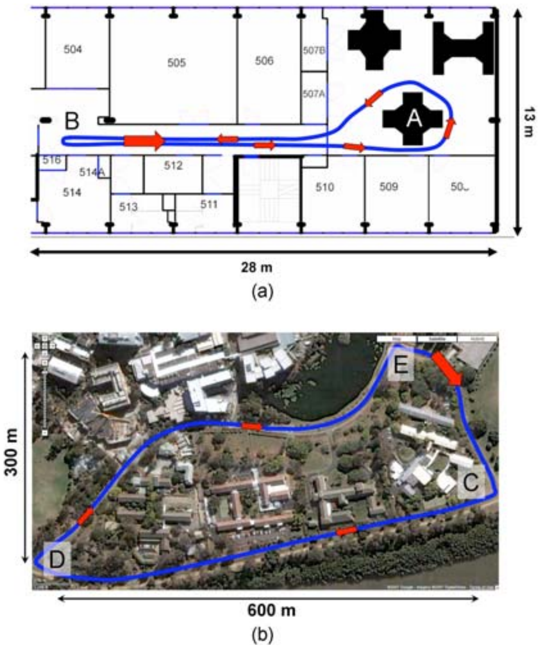
$$\Delta\mathbf{p}_i = \alpha\left[\sum_{j=1}^{N_f}\left(\mathbf{p}_j - \mathbf{p}_i - d\mathbf{p}_{ij}\right) + \sum_{k=1}^{N_t}\left(\mathbf{p}_k - \mathbf{p}_i - d\mathbf{p}_{ki}\right)\right] \qquad (12)$$

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*. The experience map is subject to the same constraints of any network style learning system – appropriate learning rates must be used to balance rapid convergence with instability.

4 Experimental Setup

Experiments were performed in the two environments shown in Figure 7. The indoor environment was part of the floor of an office style building. A Macbook notebook was positioned on a roller office chair, facing 'forwards' with neutral pitch, and pushed around two loops of the environment along the trajectory shown in Figure 7a. The chair was moved as if it were a wheelchair style robot – it could move forwards, rotate on the spot, but could not move sideways parallel to the camera lens. The controller attempted to move the chair at a constant speed, but had to slow down to make turns such as at point B. The length of a single loop was approximately 45 m. The two loops took 140 seconds to traverse.

The outdoor environment consisted of part of the University of Queensland campus. A Macbook was mounted on the front bonnet of a car facing forwards and with neutral pitch. The car was driven at approximately constant speed around two loops of the environment. The weather was generally sunny although the sun was briefly obscured at times during the experiment by clouds. The length of a single loop was approximately 1.6 km. The two loops took approximately 460 seconds to traverse. The image data from both experiments was saved to disk, and replayed at real-time speed to the vision system, combined with the RatSLAM system. The size of the pose cell matrix was 60×60×36 cells. Most importantly, *no parameters were changed between the two environments*.



(a)

(b)

Figure 7 – (a) Indoor and (b) outdoor test environments with robot trajectories shown. The large arrow shows the starting location and direction, with smaller arrows showing the direction of travel. © Google Maps®.

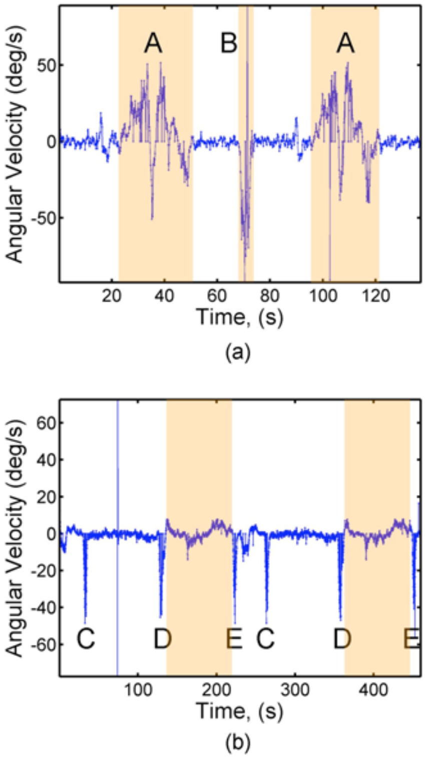
5 Results

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 two test environments.

5.1 Angular Velocity Extraction

Figure 8 shows the rotational speeds in each environment as calculated by the image array matching algorithm. The movement around the table cluster A in Figure 7a can be clearly seen in the two highlighted regions labelled A in Figure 8a. The sharp turn at the end of the corridor at B in Figure 7a can be seen at the point labelled B in Figure 8b. There are a couple of probably erroneous angular velocity values, one at 70 seconds and one at 103 seconds.

In the outdoor environment, the sharp turns at points D, E, and F in Figure 7b are clearly represented in the angular velocity graph in Figure 8b. Of perhaps more importance is the more subtle angular velocity detection corresponding to the road section between points D and E. The angular velocity is initially positive just after D, gradually becomes negative as the car rounds the bend halfway between D and E, then becomes positive again in the final road section leading up to E. These estimated velocities, at least qualitatively, reflect the actual angular velocities that would be expected driving along this stretch of road. The spike around 80 seconds is due to the vision system not correctly handling a short computer lag, which resulted in no images being recorded for several seconds.



(a)

(b)

Figure 8 – Unfiltered rotational speed calculated from consecutive image array matching, for (a) the indoor experiment and (b) the outdoor experiment.

5.2 Speed Extraction

The ‘speed’ of the camera through the environment as calculated by the vision system is shown in Figure 11. No units are shown along the vertical axes, although in strict terms the speed is measured in terms of the average difference between image array intensity values for consecutive images. For both environments, the signal is quite noisy. There is much potential for improvements to the speed detection system, such as employing some form of temporal filter that considers more than just the current and immediate last image (which represent a time period of only 0.12 seconds. Other possible solutions are discussed in Section 6. However, even with this noisy signal it is possible to form coherent maps, as is shown in Section 5.4.

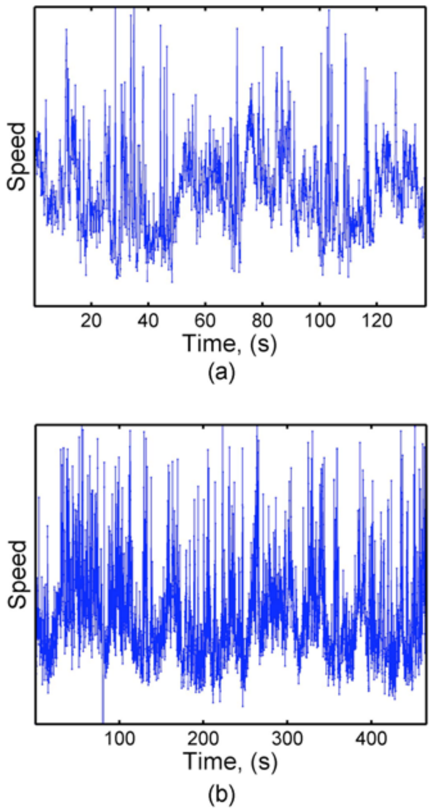


Figure 9 – Unfiltered robot ‘speed’ calculated from image change gradients, for (a) the indoor experiment and (b) and the outdoor experiment.

5.3 Template Learning and Recall

Figure 11 shows the performance of the template learning and recall system in the two environments. In the indoor environment, the system learned 159 templates during the first loop and an additional 28 templates during the second loop. In the outdoor environment the system learned 324 templates during the first loop and an additional 112 templates during the second loop. The inferior recognition performance in the outdoor environment during the second loop was probably due to the more challenging perceptual conditions. Even during the short experiment, illumination conditions varied significantly and traffic was encountered throughout much of the circuit (Figure 10).

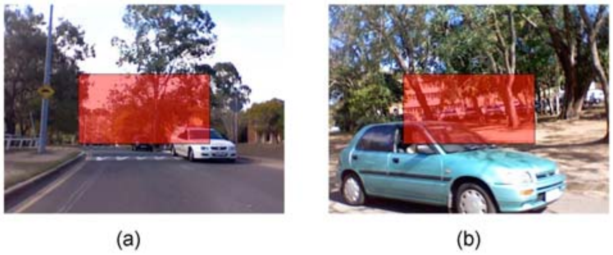


Figure 10 – Some example capture images from the outdoor environment. The traffic encountered around the loop caused some problems for the template learning / recognition system.

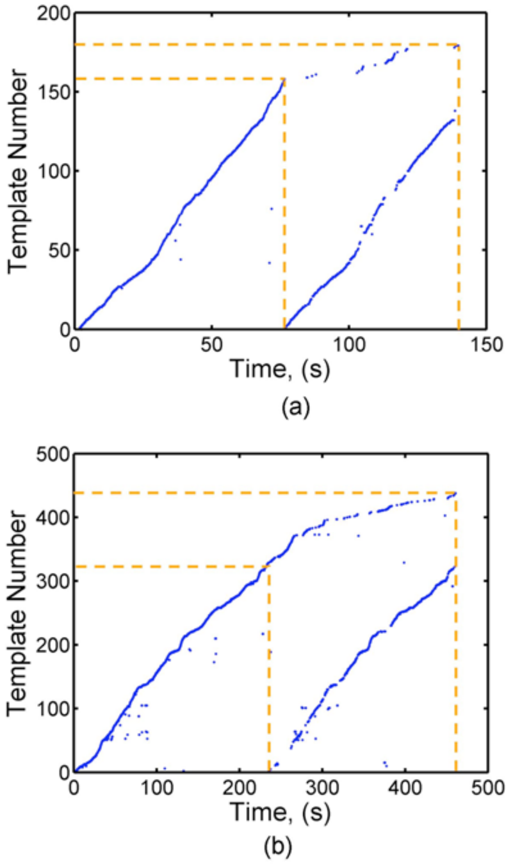


Figure 11 – Visual template learning and recognition in the (a) indoor and (b) outdoor environments.

5.4 Mapping

The experience maps created by the RatSLAM system and experience mapping algorithm are shown in Figure 12. For reference, the ground truth trajectories of the robot through the environments are also shown. For the indoor environment, ground truth was obtained by tracing out the observed path of the camera on an accurate floorplan. For the outdoor environment, the ground truth trajectory was obtained by tracing the path of the road through the aerial photo shown in Figure 7b. The red crosses show the actual camera locations and locations as estimated by the navigation system at various times through the experiment. Because there was no absolute scale for the experience maps, they are manually scaled to facilitate comparison with the ground truth trajectories.

The experience maps closely resemble the actual path of the camera through the environment, although they are not identical. In the indoor environment, because only a forward facing camera was used, there was no information to explicitly bind together the forward and reverse paths through the corridor. Instead, the experience mapping algorithm positioned the paths based on the odometric information obtained from the sequence of images, resulting in a slight alignment error. This problem could be solved by adding a backwards facing camera, or a panoramic camera, as in the original outdoor RatSLAM experiments [Prasser *et al.*, 2005]. In the outdoor environment, the map also closely resembles the actual camera trajectory, although it is slightly warped in places.

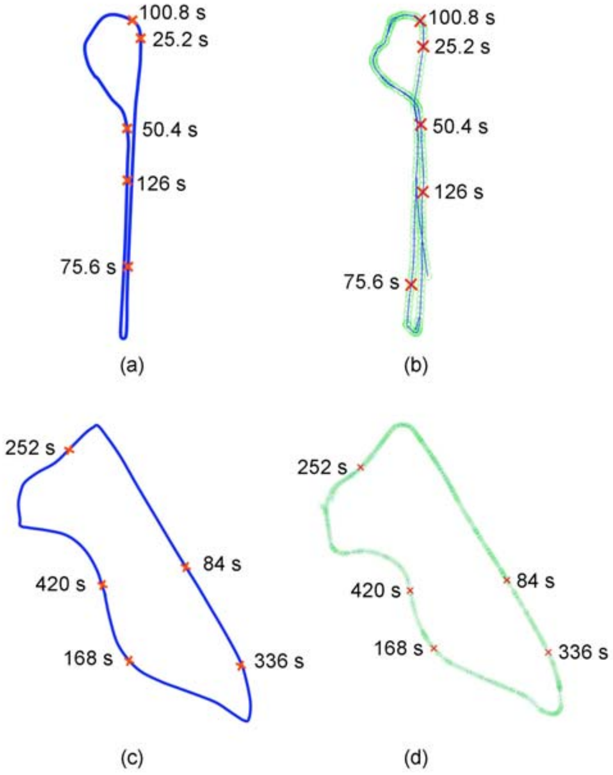


Figure 12 – Ground truth trajectories and corresponding experience maps, for the (a-b) indoor environment and (c-d) outdoor environment. Crosses show the (a, c) actual and (b, d) estimated locations at five times during the experiments.

6 Conclusion

Although preliminary in nature, the results presented in this paper have demonstrated the potential for mapping ground-based environments with only 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, zero 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 coherent and representative maps.

As this was an initial investigative study, there are many areas where future work may be productive. 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 corridor or road, solving slight alignment problems such as in Figure 12b. 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. 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 yield 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 [Milford *et al.*, 2006]. It is interesting to note that animals such as bees may not have a means of measuring absolute speed [Srinivasan *et al.*, 2000].

Most of the areas highlighted for possible future work are brought about by attempting to predict the problems that will be encountered during longer experiments, in larger, more challenging environments. One of the most profitable areas of future work will be to test and develop the vision system in such conditions. These experiments will quickly determine the scalability of the approach to visual SLAM presented in this paper.

References

[S. Thrun, 2002] S. Thrun, "Robotic Mapping: A Survey," in *Exploring Artificial Intelligence in the New Millennium*: Morgan Kaufmann, 2002.

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

[B. Kuipers, *et al.*, 2004] B. Kuipers, *et al.*, "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.

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

[M. Montemerlo, *et al.*, 2003] M. Montemerlo, *et al.*, "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.

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

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

[G. Grisetti, *et al.*, 2005] G. Grisetti, *et al.*, "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.

[L. A. Clemente, *et al.*, 2007] L. A. Clemente, *et al.*, "Mapping Large Loops with a Single Hand-Held Camera," presented at the Robotics: Science and Systems, Atlanta, United States, 2007.

[A. J. Davison, *et al.*, 2007] A. J. Davison, *et al.*, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1052-1067, 2007.

[A. J. Davison, *et al.*, 2005] A. J. Davison, *et al.*, "Vision-Based SLAM for a Humanoid Robot," presented at the International Conference on Robotics and Automation, Barcelona, Spain, 2005.

[J. M. Porta, *et al.*, 2005] J. M. Porta, *et al.*, "Active Appearance-Based Robot Localization Using Stereo Vision," *Autonomous Robots*, vol. 18, pp. 59-80, 2005.

[R. Sim, *et al.*, 2005] R. Sim, *et al.*, "Vision-based SLAM using the Rao-Blackwellised Particle Filter," presented at the International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, 2005.

[N. Cuperlier, *et al.*, 2005] N. Cuperlier, *et al.*, "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.

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

[M. J. Milford, *et al.*, 2006] M. J. Milford, *et al.*, "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.

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

[D. Prasser, *et al.*, 2005] D. Prasser, *et al.*, "Outdoor Simultaneous Localisation and Mapping using RatSLAM," presented at the International Conference on Field and Service Robotics, Port Douglas, Australia, 2005.

[M. V. Srinivasan, *et al.*, 2000] M. V. Srinivasan, *et al.*, "Honeybee Navigation: Nature and Calibration of the "Odometer"," *Science*, vol. 287, pp. 851-853, 2000.