Aerial SLAM with a Single Camera using Visual Expectation

Michael J. Milford, *Member, IEEE*, Felix Schill, *Member, IEEE*, Peter Corke, *Fellow, IEEE*, Robert Mahony, *Senior Member, IEEE*, Gordon Wyeth, *Member, IEEE*

***Abstract*—Micro aerial vehicles (MAVs) are a rapidly growing area of research and development in robotics. For autonomous robot operations, localization has typically been calculated using GPS, external camera arrays, or onboard range or vision sensing. In cluttered indoor or outdoor environments, onboard sensing is the only viable option. In this paper we present an appearance-based approach to visual SLAM on a flying MAV using only low quality vision. Our approach consists of a visual place recognition algorithm that operates on 1000 pixel images, a lightweight visual odometry algorithm, and a visual expectation algorithm that improves the recall of place sequences and the precision with which they are recalled as the robot flies along a similar path. Using data gathered from outdoor datasets, we show that the system is able to perform visual recognition with low quality, intermittent visual sensory data. By combining the visual algorithms with the RatSLAM system, we also demonstrate how the algorithms enable successful SLAM.**

I. Introduction

Recent advances in the technology behind small flying vehicles, often called micro aerial vehicles (MAVs), have led to a range of research on topics such as surveillance, inspection and search and rescue. Localization is a key requirement for high level autonomous robot operation in these tasks. While GPS has long been used as a localization sensor on both piloted and unpiloted air vehicles, localization using GPS is not possible in indoor or cluttered outdoor environments where GPS is generally not available ("GPS-denied" environments). One solution to operating in GPS-denied environments is to employ artificial landmarks or external camera systems [1, 2], but in many cases, environment modification is impractical or undesirable, restricting applicability.

To avoid the need for GPS or external infrastructure, a number of range sensor-based and camera-based onboard localization algorithms have been developed. Range-based techniques include work by Grzonka et al. [3] and Bachrach et al. [4], who combined onboard laser range finders with a particle filter to perform SLAM on a flying robot. Vision-based approaches such as those by Ahrens [5], Angeli [6] and Blosch [7] have implemented state of the art camera-only SLAM algorithms such as MonoSLAM [8] and PTAM [9] to track features between frames and perform visual SLAM on flying robots. Common to all these vision-based approaches has been continuous access to high quality camera images, and with the exception of a few implementations such as by Courbon et al. [10, 11], the need to track features over many frames.

In this paper, we present an approach to vision-based SLAM on a micro aerial vehicle that requires only low quality visual information and does not rely on tracking features between frames. Our approach is inspired by the observation that a flying robot equipped with appropriate local movement behaviors tends to follow similar "safe" paths through the constraints of a typical GPS-denied environment. We first present a visual place recognition algorithm that uses 1000 pixel images, and a light-weight visual odometry algorithm that calculates self-motion estimates from patch tracking in 240 × 200 pixel images. We then describe a new visual expectation algorithm that enhances recall of familiar places without compromising on precision or accuracy, independently of the mapping backend. Using data gathered from two outdoor datasets, we show that the system is able to perform visual recognition even with low quality and intermittent visual sensory data. By combining the visual processing algorithm with the RatSLAM SLAM system, we also demonstrate how the odometry, recognition and expectation algorithms together enable successful mapping.

The paper proceeds as follows. In Section 2 we give a short overview of vision-based navigation and mapping algorithms for MAVs. Section 3 provides a description of the RatSLAM system and the visual odometry, recognition and expectation algorithms. We describe the experimental setup in Section 4, before presenting the results in Section 5. The paper concludes in Section 6.

II. Vision-Based Mapping and Navigation on MAVs

Recent advances in vision-based SLAM algorithms have led to an increase in the number of vision-based localization systems on flying robots.Ahrens et al. [5] used an off board visual tracking system combined with the visual tracking algorithms of Davison [8] and an onboard camera to perform navigation and localization on a Hummingbird quadrotor over a short distance, with some drift. Angeli et al. [6] performed localization and loop closure on both a blimp and Twinstar MAV equipped with a downwards facing camera. More recently, Blosch et al. [7] used the visual SLAM algorithm of Klein et al. [9] to accurately localize a

This research was supported by the Australian Research Council through discovery grant DP0880509, "Image-based teleoperation of semi-autonomous robotic vehicles" and an Australian Research Council Special Research Initiative on Thinking Systems, TS0669699, to GW.

M.M., P.C. and G.W. are with the School of Engineering Systems at the Queensland University of Technology, QLD, 4001, Australia. Tel: +61 7 3138 9969, e-mail: michael.milford@qut.edu.au.

R.M. and F.S. are with the School of Engineering at the Australian National University, Canberra, ACT, 0200, Australia, e-mail: robert.mahony@anu.edu.au.

Hummingbird quadrotor platform flying a small loop in an indoor environment, using only a single onboard camera. Courbon et al. [10, 11] used a visual path-based approach that stores images of the environment as ordered routes, to achieve mapping and navigation on an X-4-flyer quadrotor in an indoor environment using a single camera.

One common characteristic of almost all flying visual SLAM approaches has been that they have had access to reasonably high quality visual sensory data, suiting the detection and tracking of features over multiple frames. The research described in this paper investigated how simple the visual sensory data could be while still performing visual SLAM on a flying robot.

III. Vision-Based SLAM Components

In this section, we describe the mapping backend RatSLAM, and the visual odometry, visual template and visual expectation algorithms.

A. RatSLAM System

RatSLAM is a robotic visual SLAM system inspired by models of the neural processes underlying navigation in the rodent brain. It has been deployed on a range of robot and vehicle platforms in many different environments [12-14], but always on ground-based robots. RatSLAM consists of three major components – a continuous attractor neural network known as the *pose cells*, a graphical map known as the *experience map*, and a set of *local view cells*. The pose cell network encodes the robot's pose state, and performs the role of filtering self-motion and visual information. The local view cells encode distinct visual scenes or templates – each cell becomes associated with a distinct visual template. The experience map provides a topologically correct and locally metric map of the environment for use in navigation. In this paper, we use the experience map to show the utility of the visual recognition system in performing SLAM. More detailed descriptions of the RatSLAM system can be found in [12, 13].

B. Visual Odometry

A lightweight visual odometry system was implemented using patch tracking of two fixed patch locations, shown in Fig. 1a. The flyer (described in Section IVa) was treated as an approximately non-holonomic vehicle, with patch A used to track vehicle yaw, and patch B used to track the vehicle's translational speed relative to the ground plane. No attempt was made to extract scale, and consequently the calculated translational speed depended on both the vehicle's speed and altitude. The comparison between patches was performed by calculating the average intensity difference, *f()*, between pixel patches (normalized to 50% mean intensity) in the current and past image over a range of relative offsets:

$$f\left(\Delta x,\Delta y,I^{j},I^{k}\right)=\frac{1}{r^{2}}\sum_{x=0}^{r}\sum_{y=0}^{r}\left(p^{j}_{x+\Delta x,\,y+\Delta y}-p^{k}_{xy}\right) \qquad (1)$$

where I^j and I^k are the past and current images, r is the patch size in pixels, p is the pixel intensity, and Δx and Δy are the patch offsets. The patch shift used for odometry purposes was the shift (Δx_m, Δy_m) that minimized *f()* for the two patches:

$$\left(\Delta x_{m},\Delta y_{m}\right)=\underset{\Delta x,\Delta y\in[-\rho,\rho]}{\arg\min}\; f\left(\Delta x,\Delta y,I^{j},I^{k}\right) \qquad (2)$$

where ρ is the range of patch offsets. The horizontal pixel shift Δx for patch A was multiplied by a gain constant, ς, to obtain a yaw velocity estimate, ω:

$$\omega=\varsigma\Delta x_{m}^{A} \qquad (3)$$

The vertical pixel shift Δy for patch B was multiplied by a gain constant, ν, to obtain a translational speed estimate, s:

$$s=-\nu\Delta y_{m}^{B} \qquad (4)$$

An example of the flyer trajectory calculated using this lightweight visual odometry system is shown in Fig. 1b, and can be compared to the ground truth trajectory in Fig. 5. The visual odometry system is scale-less, but we have selected a suitable gain for the purposes of comparison to ground truth.

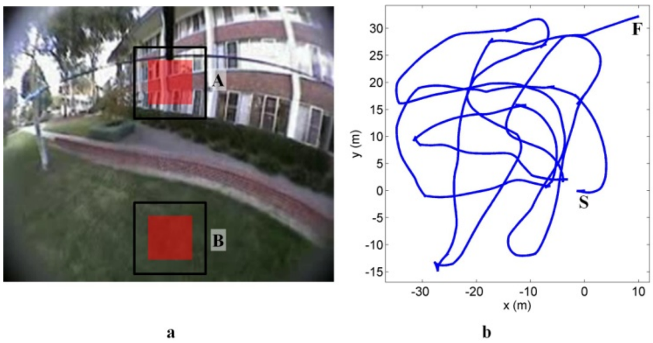

Fig. 1. (a) Inter-frame motion of patch areas A and B provided approximate yaw rate and translational speed, respectively. (b) The robot's trajectory in the (x, y) plane as calculated using only the visual odometry signal (S = Start, F = Finish).

C. Visual Templates

Color images were captured from the robot's onboard cameras at a resolution of 240 × 200 pixels, at a somewhat variable frame rate averaging 12 frames per second (Fig. 2). The video was run through a deshaking filter to reduce some of the more severe image jerk (VirtualDub Deshaker filter, available at [15], default values used). These frames were resolution reduced, converted to grayscale, and Gaussian blurred (radius 5) to form 32 × 32 images, which formed the basis of the visual templates used by the RatSLAM system.

Template differences, *D*, between the current candidate template *i* and each learnt template *j* were calculated using a normalized sum of pixel intensity differences performed over a moving sub frame in the resolution reduced images:

$$D_{j}=\min_{\Delta x,\Delta y\in[-\sigma,\sigma]} g\left(\Delta x,\Delta y,i,j\right) \qquad (5)$$

where σ is the template offset range, and *g()* is given by:

$$g\left(\Delta x,\Delta y,i,j\right)=\frac{1}{s^2}\sum_{x=0}^{s}\sum_{y=0}^{s}\left(p^{i}_{x+\Delta x,\,y+\Delta y}-p^{j}_{x,y}\right) \tag{6}$$

where s is the size of the template sub frame. These template differences were normalized by the current recognition threshold, T_j, of each template to calculate the template with the smallest normalized difference. The current template index, k, was calculated by:

$$k=\begin{cases}\underset{j\in[0,n]}{\arg\min}\, D_j/T_j & D_j/T_j<1\\ i & \min\left(\mathbf{D}/\mathbf{T}\right)\geq 1\end{cases} \tag{7}$$

where n was the number of learnt templates, and i was the index of the current template candidate. If no templates were close enough to the current scene, the current candidate template i was added to the learnt templates. This same image difference metric was used to compare the current and immediately previous frame, to disable template learning and visual odometry for noisy corrupted frames.

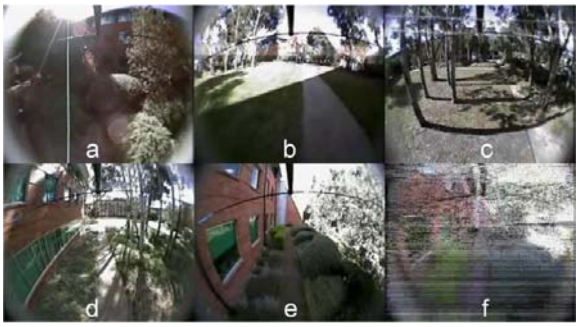

Fig. 2. Sample onboard camera images. Illumination variation and lens flare was quite significant, and there was also environment aliasing (see panels *d* and *e*). Transmission drop-outs around buildings and under trees frequently caused dropped or degraded frames such as shown in panel *f*, which is the same location as shown in panel *e*.

D. Visual Expectation

In all previous RatSLAM visual processing implementations, fixed, pre-determined recognition thresholds were used for all visual templates. As a robot explored an environment, it would compare the current visual scene with a library of visual templates it had already learned. If the scene was similar to an existing visual template, it would "recall" that visual template. However, if the current scene was distinct, it would add it to the template library. In this way, the robot would build up a library of distinct visual templates representing the visual appearance of the environment. This approach, while initially requiring manual tuning of the threshold value, was generally applicable in similar environments and required little further tuning. Here we describe a new visual expectation algorithm for template-based recognition systems, inspired by the recognition process in the mammalian brain.

In the brain, recognition of objects or place can be primed by contextual information [16]. If an animal is in a place it knows to be familiar, circuits in the brain dynamically change their properties to increase its rate of recall of familiar places or scenes. Based on this principle, we describe a new visual expectation algorithm, which consists of using dynamic, scene-specific recognition thresholds, rather than one global threshold. The algorithm implements a *specific* form of expectation; recall of a scene can only increase recall of scenes previously known to have occurred in frames immediately following that scene.

A recalled visual template primes recognition of visual templates previously seen soon after the recalled visual template, by increasing the image comparison threshold, T_i, at which those visual templates are recalled:

$$T_i=T_i+\sum_{j=i-\mu}^{i-1}\psi V_j-\alpha\left(T_i-T_D\right) \tag{8}$$

where μ is the expectation range, ψ is the expectation increment per video frame, V is a binary array encoding the current template matches, and α is a per-frame threshold decay. T is bounded between a default threshold value T_D and a maximum threshold value T_M, set to $2T_D$. Figure 3 shows an example of a recognized visual template initiating a chain of increased recognition thresholds for subsequent frames. When template 1 is recognized (Fig. 3b), it increases the recognition threshold of templates 2 and 3, which leads to a sequence of recognized templates (Fig. 3c-d), which with a static threshold would not have been matched.

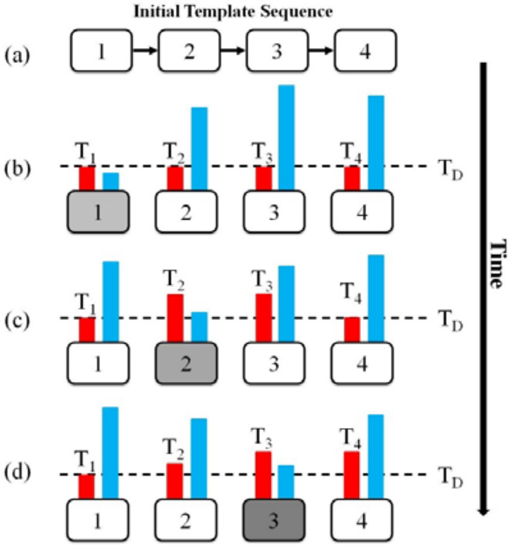

Fig. 3. Enhanced template recognition thresholds due to visual expectation. The left bars indicate the recognition threshold, while the right bars indicate the matching difference for that template.

IV. Experimental Setup

In this section, we describe the robot platform, testing environment, and provide a list of parameter values for the vision algorithms.

A. Robot Platform

A quadrotor flying platform, an Ascending Technologies *Hummingbird*, was used for data collection. The *Hummingbird* is a small, lightweight hovering vehicle that is well suited for flying in cluttered environments. The vehicle was equipped with two small color cameras with fisheye wide-angle lenses (1.7mm focal length, approximately 170 degrees field of view), one facing forwards and one facing backwards. Camera images were relayed to the base station computer via two wireless 5.8 GHz radio links, synchronized with time stamps and stored to disk. The quadrotor's onboard control system used MEMS gyroscope sensors and accelerometers for self-leveling hover control.

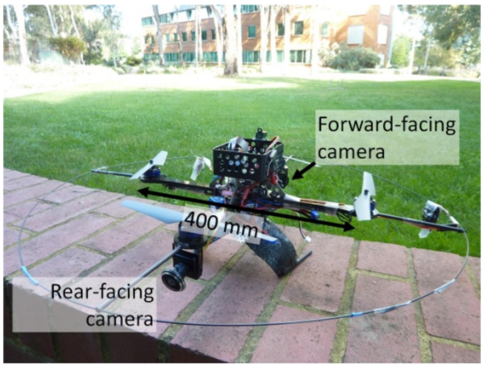

Fig. 4. The quadrotor used in experiments.

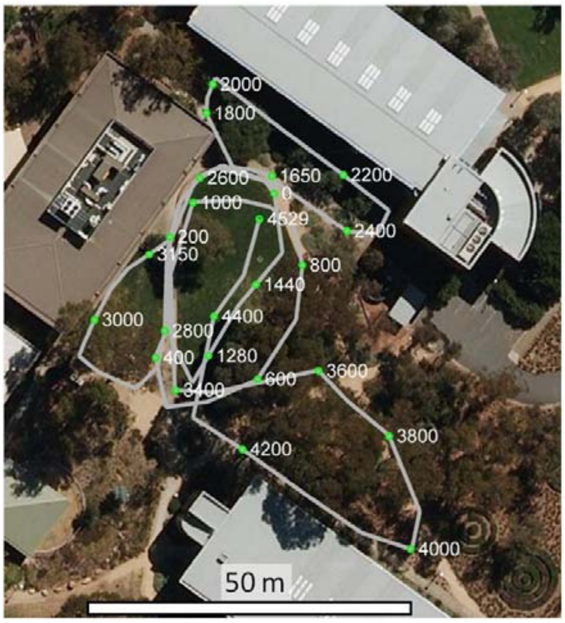

Fig. 5. The robot trajectory through the environment, projected onto a 2D terrain photo (© Google Maps). The order of traversal is indicated by the frame number labels.

B. Testing Environment and Flights

Experiments were run over a period of two days at the Australian National University in Canberra, Australia (Fig. 5). The testing environment was an area measuring approximately 60 × 90 m containing a mixture of open grass areas, trees, pathways and scrub. The quality of video reception was negatively impacted by the surrounding buildings, trees and occlusions. The pilot remote-controlled the vehicle along the various paths while following it on foot, enabling the safe traversal of challenging and narrow passages. Due to the nature of flying in unconstrained airspace the precise routes taken by the flyer were never exactly identical, but were deliberately chosen to be at least similar to previous passes of the same paths.

The primary dataset (dataset 1) described in this paper consisted of a piloted flight on the second day of approximately 500 meters along the trajectory shown in Fig. 5, containing 5 loops. The second data set (dataset 2) described in this paper was a flight from the first day of experimentation, which had some overlap with the primary dataset. The quality of the video-feed for this run was extremely poor, with approximately 30% of frames being unusable from interference, as well as generally degraded image quality.

The nature of the environment made automated GPS or external camera-based ground truth tracking very difficult – the flyer flew under trees and buildings and passed through many visually obscured areas. Consequently, to extract an approximate ground truth a program was constructed which allowed manual tagging of approximate flyer locations for each frame based off both forward and reverse facing onboard video frames. 133 frame locations were manually tagged, and intermediate frame locations were interpolated.

C. Parameters

Table I provides a list and description of all the key visual algorithm parameters and values. RatSLAM parameter values were as given in [13].

TABLE I
PARAMETER LIST

Parameter	Value	Description
r	32 pixels	Odometry patch size
ς	0.67 °/pixel	Yaw gain constant
v	0.05 m/pixel	Translational speed constant
ρ	10 pixels	Patch offset range
μ	5	Expectation range
s	24 pixels	Template sub frame size
σ	4 pixels	Template offset range
ψ	0.1	Expectation increment
T_D	0.03 – 0.15	Default threshold
α	0.02	Per-frame threshold decay

V. RESULTS

In this section, we present the place recognition performance of the visual template system with and without the visual expectation algorithm, as well as the map produced by the RatSLAM system. We present results for the primary dataset, gathered on day 2, as well as for a second dataset gathered on day 1 with very poor video quality due to interference.

A. Dataset 1

We generated an error-recall graph by running 49 trials with and without visual expectation enabled (98 trials in total), for a range of default template recognition thresholds (T_D). An error-recall graph rather than precision-recall graph was used as a more representative metric – because the flyer never repeated the same path exactly, the precision calculation was very sensitive to the matching distance threshold. The emphasis in the results section is on *relative* comparison rather than absolute accuracy.

Figure 6 shows the error-recall graph for the primary experiment on day 2. The graph was constructed by first classifying every frame in every trial as true positive (TP), true negative (TN), false positive (FP) or false negative (FN). To assist in this classification, the dataset was manually divided into novel and repeated sections (indicated by the shaded areas in Fig. 9). The *recall error* was calculated as the distance between the location where a template was first learned and the locations associated with any frames in which that same template was recalled. Any frames in which the recall error was larger than 5 meters were classified as False Positives. All error-recall plots have been truncated to exclude some points obtained from trials with extreme threshold values, corresponding to trials where only a few templates were learned for the entire environment. In sections manually tagged as repeated sections, frames in which new templates were learnt were classified as false negatives.

For any given recall level, the average error with expectation is significantly lower than without expectation. The difference becomes especially noticeable at high levels of recall – with expectation, the error increases slowly up to very high recall levels (97%), while recall rates above 80% without expectation result in a rapid and unstable increase in average recall error. The filtering provided by the pose cells in RatSLAM enables it to operate in the high recall-low error region, with a typical operating point shown by a thick arrow in Fig. 6. The dashed arrows indicate the operating points on the no expectation line for a matching recall rate and a matching error amount. For these points, expectation increases the recall rate from 57% to 86%, and decreases the average error by 46% from 1.74 meters to 0.94 meters.

Figure 7 shows a plot of the average area size encoded by each template (the radius of the minimum bounding circle encompassing all template recall locations, see Fig. 8). Expectation makes the visual templates significantly more spatially specific at all recall levels. For example, at a recall level of 91%, with expectation the radius of the average area encoded by each visual template with expectation is 4.2 meters, but without expectation the radius increases to 12.3 meters. Figure 8 is a plot showing the locations in which a specific template was recalled for two trials with the same recall rate, one with expectation and one without. With visual expectation, templates coded for small specific areas, and there was little ambiguity in coding location (Fig. 8a). Without visual expectation, templates coded for larger areas and were more likely to code for multiple distinct locations (Fig. 8b).

Figure 9 shows the matched template over the entire experiment for the three recall-error operating points highlighted in Fig. 6. The shaded areas indicate repeated sections of path, where recall should have occurred. The dots at the top of the graphs indicate the classification of each frame. With expectation, long sequences of templates are correctly recalled in repeated sections of the dataset, as shown in Fig. 9a. To achieve the same average recall error without expectation, the recall rate drops significantly, resulting in less reliable recognition of familiar templates in repeated sections of the dataset (Fig. 9b). To achieve the same recall rate without expectation, the templates become much less specific, leading to many more false positive matches (Fig. 9c). Figure 10 shows the recalled images over one 135 frame sequence for the three sections highlighted in Figs. 9a-c. Once a template is recalled with expectation, a long sequence of subsequent templates are correctly recalled (Fig. 10b).

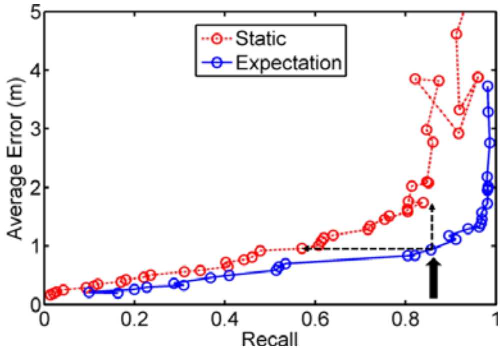

Fig. 6. Error-recall graph comparing template recognition with and without visual expectation. The lines between graph points indicate the direction of increasing default template recognition threshold, T_D.

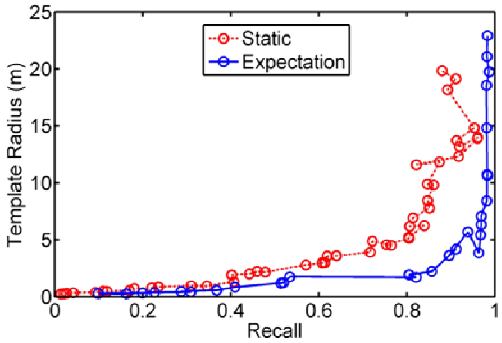

Fig. 7. Template size-recall graph showing the average area encoded by a template with and without visual expectation.

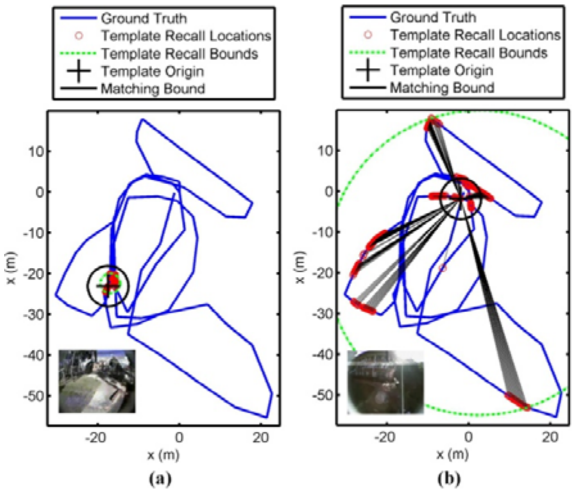

Fig. 8. Single template recall locations for trials with the same recall rate but (a) with and (b) without visual expectation. The matching bound circle is the 5 meter "correct recall" circle. Straight thin lines indicate false positive matches. The cross indicates the location where the template was first learnt.

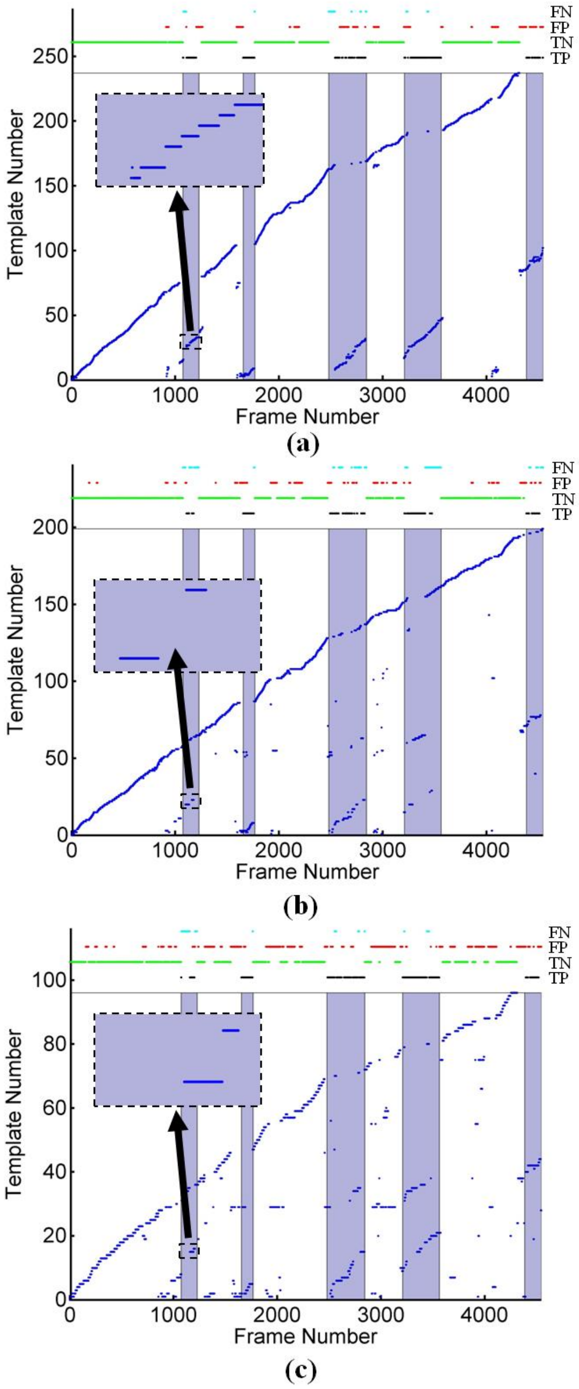

Fig. 9. Visual template graphs (a) with and (b-c) without expectation for a matching (b) error level and (c) recall rate. Lightly shaded areas show repeated sections of the path where recall would be expected to occur. The insets show in detail the regions of the graph corresponding to the frame sequences shown in Fig. 10.

Figure 11 shows the experience map produced with visual expectation for dataset 1. The map contains 759 experience nodes and 794 links between experience nodes. While the map is not a precise metric representation of the environment, it does capture the layout of the environment and the topological connectivity is correct. The video accompanying this paper shows the experience map forming as the visual system learned and recalled visual templates. All visual algorithms and RatSLAM ran at real-time speed.

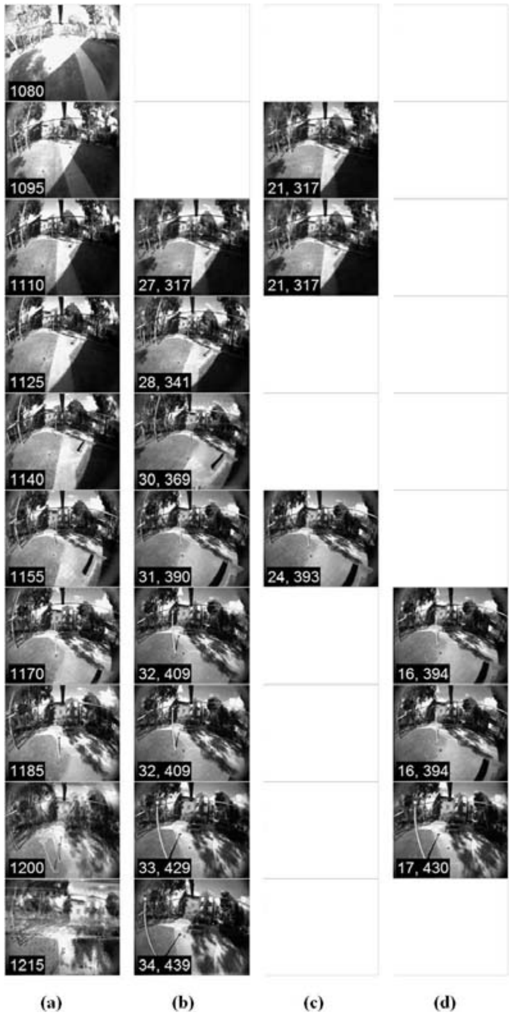

Fig. 10. A comparison of the recalled frames for one repeated section of path at 15 frame intervals, for the three highlighted sequences in Fig. 9a-c. Blank frames indicate false negatives. (a) The video frame number. (b) With expectation, T_D = 0.0625. Numbers indicate (recalled template number, corresponding frame number). (c) Without expectation, T_D = 0.075. (d) Without expectation, T_D = 0.1025.

B. Dataset 2

Dataset 2 consisted of a flight through the same environment on the first day of experiments, at a different time of day to dataset 1. While the image quality was far too poor to perform visual odometry on, there was the possibility of recalling templates learned from dataset 1. Figure 12 shows the error-recall performance of the visual system after prior learning of the first dataset. The recall error is much higher for a given recall level when compared with dataset 1, and the system stops sensibly recalling visual templates above a recall level of about 25%. Expectation still significantly reduced the average recall error for a given recall level. Despite the datasets being from different days and times of day, and the video quality from dataset 2 generally being very poor, about 19% of templates could be recalled with a low average recall error of 0.66 meters. Figure 12 shows some examples of templates learned from dataset 1 being recalled in dataset 2.

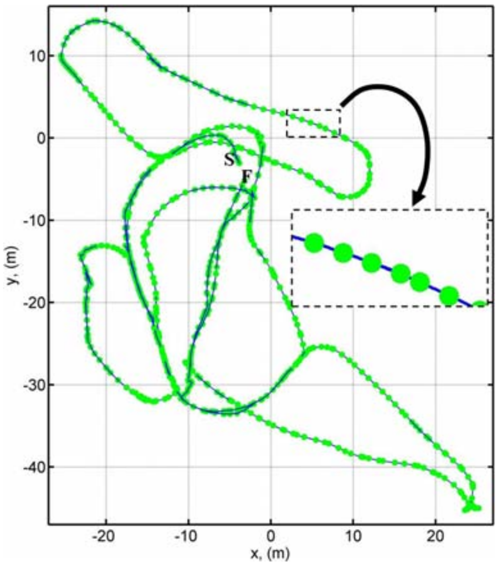

Fig. 11. The RatSLAM experience map produced with visual expectation, T_D = 0.0625. Circles show experience nodes and lines show links between nodes (inset shows a zoomed area of the map).

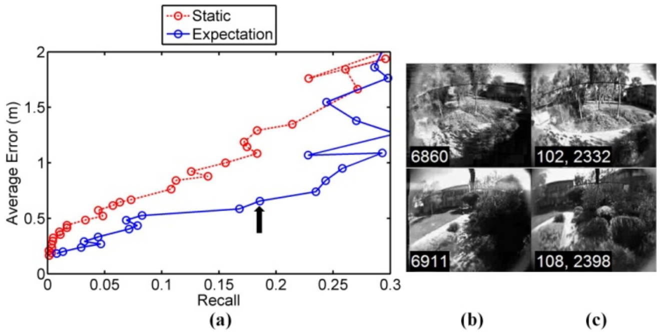

Fig. 12. (a) Error-recall graph with and without visual expectation. (b) Frames and (c) associated templates learned from the first dataset being correctly recalled during the second dataset, for the operating point indicated by the arrow in (a).

VI. Discussion

The visual expectation method presented in this paper enables significantly lower recall errors to be achieved for a desired recall level. We have demonstrated the visual expectation algorithm on a challenging dataset gathered from a flying platform in a cluttered environment. We would expect the algorithm to be equally applicable in any vision-based application where explicit environment geometry extraction is not needed, and where movement is constrained to somewhat repeated paths, rather than open-field motion. Place recognition required only 1000 pixel images, and while the visual odometry system used 240 × 200 pixel images, any lightweight self-motion calculation could be used. We envisage applications in cluttered environments such as mixed indoor-outdoor environments, where onboard movement behaviors will guide the robot along similar repeated paths.

Future work will pursue a number of tracks. By keeping a short time history, it may be possible to perform retrospective backward expectation to better match repeated paths. For example, in Fig. 10b, by altering the matching thresholds for “missed” templates in previous frames, it may be possible to correctly match the entire frame sequence. In addition, the expectation algorithm will be expanded to handle unconstrained movement, rather than only forwards movement along a trajectory. This will enable unconstrained flying robot movement (rather than movement mainly along a primary axis), and application on holonomic ground-based robot platforms. Finally, we will investigate the extent to which visual expectation can be used to match images from perceptually dynamic datasets such as obtained during day-night robot operation.

Acknowledgment

We thank Evan Slatyer for his help in performing the quadrotor experiments.

References

[1] E. Altug, J. P. Ostrowski, and R. Mahony, "Control of a quadrotor helicopter using visual feedback," presented at International Conference on Robotics and Automation, Washington, United States, 2002.

[2] S. Park, D. H. Won, M. S. Kang, T. J. Kim, H. G. Lee, and S. J. Kwon, "RIC (Robust internal-loop compensator) based flight control of a quad-rotor type UAV," presented at International Conference on Intelligent Robots and Systems, Edmonton, Canada, 2005.

[3] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a Navigation System for Autonomous Indoor Flying," presented at International Conference on Robotics and Automation, Kobe, Japan, 2009.

[4] A. Bachrach, R. He, and N. Roy, "Autonomous Flight in Unstructured and Unknown Indoor Environments," presented at Robotics: Science and Systems, Zurich, Switzerland, 2008.

[5] S. Ahrens, D. Levine, G. Andrews, and J. P. How, "Vision-Based Guidance and Control of a Hovering Vehicle in Unknown, GPS-denied Environments," presented at International Conference on Robotics and Automation, Kobe, Japan, 2009.

[6] A. Angeli, D. Filliat, S. Doncieux, and J. A. Meyer, "2D Simultaneous Localization And Mapping for Micro Air Vehicles," presented at European Micro Aerial Vehicles, Braunschweig, Germany, 2009.

[7] M. Blösch, S. Weiss, D. Scaramuzza, R. Siegwart, and E. T. H. Zurich, "Vision Based MAV Navigation in Unknown and Unstructured Environments," presented at International Conference on Robotics and Automation, Anchorage, United States, 2010.

[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] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," presented at International Symposium on Mixed and Augmented Reality, Nara, Japan, 2007.

[10] J. Courbon, Y. Mezouar, N. Guenard, and P. Martinet, "Visual navigation of a quadrotor aerial vehicle," presented at International Conference on Intelligent Robots and Systems, St Louis, United States, 2009.

[11] J. Courbon, Y. Mezouar, N. Guenard, and P. Martinet, "Vision-based navigation of unmanned aerial vehicles," *Control Engineering Practice*, vol. 18, pp. 789-799, 2010.

[12] M. Milford and G. Wyeth, "Persistent Navigation and Mapping using a Biologically Inspired SLAM System," *The International Journal of Robotics Research*, 2009.

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

[14] D. Prasser, M. Milford, and G. Wyeth, "Outdoor simultaneous localisation and mapping using RatSLAM," in *International Conf. Field Service Robot.* Port Douglas, Australia, 2005.

[15] G. Thalin, "Deshaker - video stabilizer," 2.5 ed, 2010.

[16] C. R. Nolan, G. Wyeth, M. Milford, and J. Wiles, "The race to learn: spike timing and STDP can coordinate learning and recall in CA3," *Hippocampus*, 2010.