Robotics: Science and Systems 2012
Sydney, NSW, Australia, July 09-13, 2012

Visual Route Recognition with a Handful of Bits

Michael Milford
School of Electrical Engineering and Computer Science
Queensland University of Technology, Brisbane, Australia
Email: michael.milford AT qut.edu.au

***Abstract*—In this paper we use a sequence-based visual localization algorithm to reveal surprising answers to the question, how much visual information is actually needed to conduct effective navigation? The algorithm actively searches for the best local image matches within a sliding window of short route segments or 'sub-routes', and matches sub-routes by searching for coherent sequences of local image matches. In contrast to many existing techniques, the technique requires no pre-training or camera parameter calibration. We compare the algorithm's performance to the state-of-the-art FAB-MAP 2.0 algorithm on a 70 km benchmark dataset. Performance matches or exceeds the state of the art feature-based localization technique using images as small as 4 pixels, fields of view reduced by a factor of 250, and pixel bit depths reduced to 2 bits. We present further results demonstrating the system localizing in an office environment with near 100% precision using two 7 bit Lego light sensors, as well as using 16 and 32 pixel images from a motorbike race and a mountain rally car stage. By demonstrating how little image information is required to achieve localization along a route, we hope to stimulate future 'low fidelity' approaches to visual navigation that complement probabilistic feature-based techniques.**

Keywords-localization, route recognition, visual navigation, featureless

I. INTRODUCTION

Current state of the art visual navigation systems are dominated by probabilistic feature-based techniques such as FAB-MAP [1], FrameSLAM [2], and MonoSLAM [3]. These techniques have displayed impressive performance in a range of experiments, the largest of which have occurred over distances of 1000 km [1]. These feature-based approaches have desirable properties such as easy integration with metric pose estimation and semantic mapping techniques, and the ability to localize off a single image. However, such approaches also have shortcomings. Many require training on a suitable dataset to develop a visual 'codebook' before they can be applied in an environment, and using an inappropriate codebook can result in poor system performance. Feature-based techniques also rely on being able to reliably detect features, a requirement that is difficult in changing environmental conditions caused by weather, day-night cycles and seasons. The quest to map increasingly impressive datasets has been accompanied by a trend towards increasingly sophisticated algorithms, burgeoning sensor megapixel counts and large camera field of views. In this quest, one very important question has been largely neglected – *what visual information is actually needed to conduct effective vision-based navigation?*

In this paper we present evidence to suggest that, at least for localizing along a route, a simple sequence-based localization algorithm is able to match or surpass the performance of a state of the art algorithm while using images with resolutions up to *one millionth* the size and less than *two hundredth* the field of view. Specifically, we make the following contributions:

- a route recognition algorithm incorporating whole of appearance image comparison with Dynamic Time Warping [4] sequence recognition which requires no training and is not reliant on feature recognition,
- extensive experimental results showing the effect of *image resolution*, *camera field of view*, *pixel bit depth* and *sequence length*, with comparison to a state of the art method on a publicly available, modern benchmark dataset, and
- further experimental results from rally car, motorbike and office datasets, including localization using two 7 bit Lego light intensity sensors.

II. BACKGROUND

The most relevant use of image sequences to localize was in work by [5], in which loop closure was performed by comparing sequences of images based on the similarity of 128D vectors of SIFT descriptors. Due to its reliance on visual features, the method required the development of additional algorithms to address visual ambiguity caused by repetitive foliage or architecture features. The use of image sequence information has also been used to geo-locate a person based on a sequence of photos they have taken, even when none of the individual images contain recognizable features [6]. In contrast, the technique presented here forgoes the use of features and uses a novel image difference normalization scheme to partially address visual ambiguity.

While there are a large number of vision-based mapping systems [1-3, 7, 8], few current implementations use low resolution images. Earlier research did use relatively low resolution visual images to perform navigation (for reasons of computation as much as anything else), including numerous early systems such as ALVINN [9] and insect-inspired algorithms [10]. More recently, low resolution approaches have been deployed on Sony AIBO robot dogs [11] and Pioneer robots [12], including the biologically inspired RatSLAM

system [13]. In research fields outside of localization such as face recognition [14] and object recognition, matching using low resolution images has been found to be highly effective [15]. In this work we use image snapshots with up to two orders of magnitude less information than in these previous studies. The research presented here also builds on related work by the authors on localization using 'whole of image' appearance-based methods under extreme environmental change [16], in which it was shown that routes could be recognized over day to night, sunny to stormy and summer to winter transitions, albeit with image sizes of approximately 1000 pixels. Therefore we do not specifically address extreme environmental change in this paper, but rather focus on pushing the boundaries even further on the minimum resolution, pixel bit depth, and field of view required to recognize a route under more modest change.

III. Approach

The algorithm consists of two primary modules, the image comparison algorithm and the sequence recognition algorithm.

A. Image Similarity

For panoramic images, mean absolute image differences D between the current image i and all stored images j are calculated using the mean absolute intensity differences, performed over a range of horizontal offsets:

$$D_j = \min_{\Delta x \in \sigma} g(\Delta x, i, j) \tag{1}$$

where σ is the offset range, and $g(\,)$ is given by:

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

where s is the area in pixels of the image. Setting $\sigma = [0, \pi]$ enables recognition when traversing a route in reverse. For perspective cameras, σ can be set to span a range of offset angles to provide some invariance (assuming mostly distal features) to camera yaw. However, for the perspective camera datasets in this paper only a no offset case was used ($\sigma = [0]$).

B. Sequence Matching

Comparisons between the current image and all stored images yield a vector of image differences, as in [5]. The matrix **M** of image differences for the n most recent frames forms the space within which the search for matching sub-routes is performed. The key processing step is to normalize the image difference values within their (spatially) local image neighborhoods, similar to the creation of standard scores in statistics (Figure 1a). The updated image difference vectors (Figure 1b) are given by:

$$\hat{D}_i = \frac{D_i - \overline{D_l}}{\max(\sigma_l, \sigma_{\min})} \tag{3}$$

where $\overline{D_l}$ is the local mean, σ_l is the local standard deviation, over a distance of R_w images acquired before and after the current image i, and σ_{min} is a minimum standard deviation constant used to avoid undefined output, set to *1/256* of the intensity range in this paper. R_w was set to 10 frames for all experiments in this paper. Normalizing the difference values in local image neighborhoods is a process that would be counterproductive when localizing off single frames. However, in the context of recognizing sequences of images, this process ensures there are clear locally best matching images in every sub-route along the entire stored set of routes, to some extent negating the effect of global biases such as lighting changes and image commonalities.

To find likely route matches, we perform a continuous version of the Dynamic Time Warping (DTW) method proposed by Sakoe and Chiba [4]. We impose continuity and slope constraint conditions to constrain the search space. The boundary condition and monotonically increasing constraints are not applicable due to uncertainty in velocity and the need to match both forward and reverse traverses of a route. The search is continuous in that searches are started at every element in the left column of the image difference matrix (shown by the small solid circles in Figure 1).

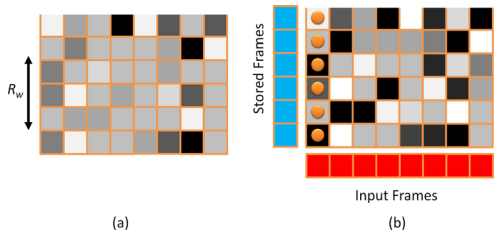

Figure 1. The image difference matrix **M** (a) before and (b) after normalization, with small circles showing the elements where each search originates. Only the bottom section of the difference matrix is shown.

The output of the DTW search produces a vector of sub-route matching scores for each search origin and each slope condition. The best matching sub-route is determined as:

$$i_{\min} = \arg\min_{1 \le i \le m} s(i) \tag{4}$$

where m is the number of stored images and $s(i)$ is the normalized sub-route difference score for sub-route i over all slope constraints:

$$s(i) = \min \mathbf{d_i} \tag{5}$$

The vector $\mathbf{d_i}$ contains the difference scores for sub-route i over all slope possibilities k:

$$d_{ik} = \frac{1}{n} \sum_{j=1}^{n} \hat{D}_{ju(i,j,k)} \tag{6}$$

where n is the sub-route length and u(i, j, k) provides the element row index in the image difference matrix:

$$u(i, j, k) = i + j \tan(v_k) \tag{7}$$

where v_k is a specific slope constraint. The slope constraint is set to span a range of values that encompass possible frame rate variations. For constant frame rate scenarios, such as the

Eynsham dataset or datasets with odometry, it is possible to use a small range or even single value of v_k.

By considering the sum of sub-route difference scores *s(i)* as a sum of normally distributed random variables, each with the same mean and variance, the sum of normalized differences over a sub-route of length *n* frames has mean zero and variance *n*, assuming that frames are captured far enough apart to be considered independent. Dividing by the number of frames produces a normalized route difference score with mean zero, variance *1/n*. Percentile rank scores can then be used to determine an appropriate sub-route matching threshold. For example, for the primary sub-route length $n = 50$ used in this paper, a sub-route threshold of -1 yields a 7.7×10^{-13} chance of the match occurring by chance.

To determine whether the current sub-route matches to any stored sub-routes, the minimum matching score is compared to a matching threshold s_m. If the minimum score is *below* the threshold, the sub-route is deemed to be a match, otherwise the sub-route is assigned as a new sub-route. An example of the minimum matching scores over every frame of a dataset (the Eynsham dataset described in this paper) is shown in Figure 2. In the second half of the dataset the route is repeated, leading to lower minimum matching scores.

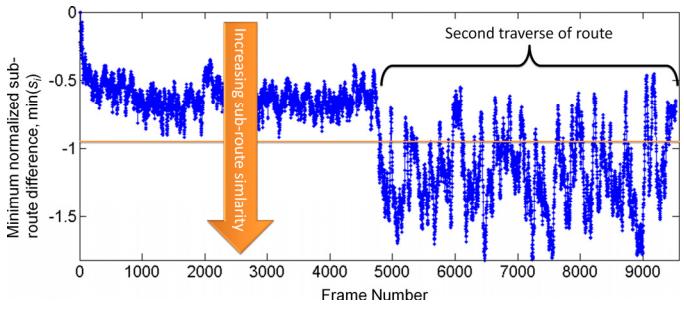

Figure 2. Normalized sub-route difference scores for the Eynsham dataset with the matching threshold s_m that yields 100% precision performance.

IV. Experimental Setup

In this section we describe the four datasets used in this work and the image pre-processing for each study.

A. Datasets

A total of four datasets were processed, each of which consisted of two traverses of the same route. The datasets were: a 70 km road journey in *Eynsham*, the United Kingdom, 2 km of motorbike circuit racing in *Rowrah*, the United Kingdom, 40 km of off-road racing up *Pikes Peak* in the Rocky Mountains, the United States, and 100 meters in an *Office* building (italics indicate dataset names). The Eynsham route was the primary dataset on which extensive quantitative analysis was performed. The other datasets were added to provide additional evidence for the general applicability of the algorithm. Key dataset parameters are provided in Table I, including the storage space required to represent the entire dataset using low resolution images.

Figure 3 shows aerial maps and imagery of the Eynsham, Rowrah and Pikes Peak datasets, with lines showing the route that was traversed twice. The Eynsham dataset consisted of high resolution image captures from a Ladybug2 camera (circular array of five cameras) at 9575 locations spaced along the route. The Rowrah dataset was obtained from an onboard camera mounted on a racing bike. The Pikes Peak dataset was obtained from cameras mounted on two different racing cars racing up the mountain, with the car dashboard and structure cropped from the images. This cropping process could most likely be automated by applying some form of image matching process to small training samples from each of the camera types. The route consisted of heavily forested terrain and switchbacks up the side of a mountain, ending in rocky open terrain partially covered in snow.

TABLE I. Datasets

Dataset Name	Distance	Number of frames	Distance between frames	Image Storage
Eynsham	70 km	9575	6.7 m (median)	306 kB
Rowrah	2km	440	4.5 m (mean)	7 kB
	http://www.youtube.com/watch?v=_UfLrcVvJ5o			
Pikes Peak	40 km	4971	8 m (mean)	159 kB
	http://www.youtube.com/watch?v=4UIOq8vaSCc http://www.youtube.com/watch?v=7VAJaZAV-gQ			
Office	53 m	832	0.13 m (mean)	1.6 kB
	http://df.arcs.org.au/quickshare/790eb180b9e87d53/data3.mat			

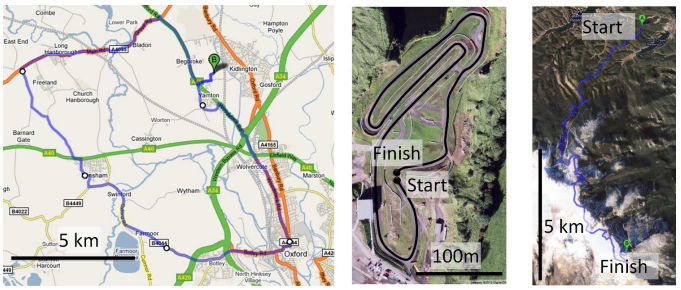

Figure 3. The (a) 35 km Eynsham, (b) 1 km Rowrah and (c) 20 km Pikes Peak routes, each of which were repeated twice. Copyright 2011 Google.

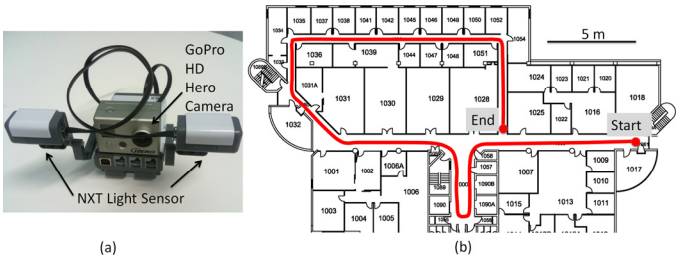

Figure 4. (a) The Lego Mindstorms dataset acquisition rig with 2 sideways facing light sensors and GoPro camera for evaluation of matched routes. (b) The 53 meter long route which was repeated twice to create the dataset.

B. Image Pre-Processing

1) Eynsham Resolution Reduced Panoramic Images

For the Eynsham dataset, image processing consisted of image concatenation and resolution reduction (Figure 5). The raw camera images were crudely cropped to remove overlap between images. No additional processing such as camera undistortion, blending or illumination adjustment was performed. The subsequent panorama was then resolution reduced (re-sampling using pixel area relation in OpenCV 2.1.0) to the resolutions shown in Table II.

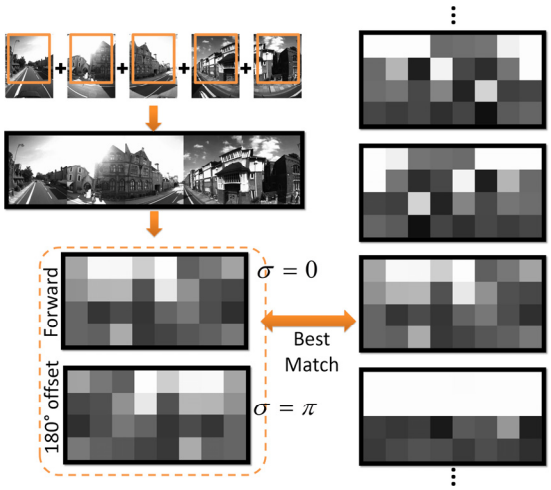

Figure 5. Image pre-processing for the full panoramic images consisted of a crude image stitching stage followed by a reduction in image resolution. The current image was compared with 0° and 180° offsets to all stored images on a pixel by pixel basis to form the image difference matrix described in Section III.B.

2) *Reduced Field of View*

For the reduced field of view experiments, a small area representing 0.4% of the total panoramic image was extracted from the centre of the forward facing image (Figure 6). The resultant 80×60 pixel image was then resolution reduced to the sizes shown in Table II.

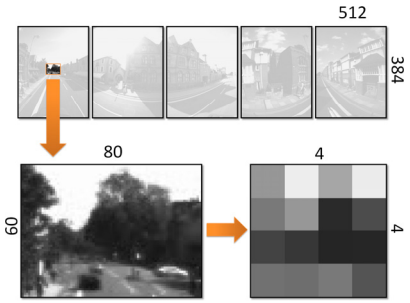

Figure 6. To evaluate the effect of drastically reducing the field of view, an area representing 0.4% of the original panoramic image was extracted and resolution reduced.

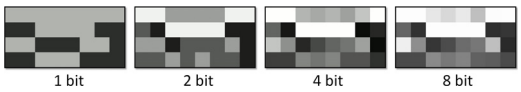

Figure 7. To evaluate the effect of reduced pixel bit depth, the resolution reduced panoramic images were resampled at 1 bit, 2 bit, 4 bit and 8 bit grayscale pixel depths.

3) *Reduced Pixel Depth*

Reduced pixel depths were obtained by reducing the bit depths of each pixel in the resolution reduced images (Figure 7). Grayscale image intensities were evenly distributed over the 256 values possible in an 8 bit intensity range, such that a 1 bit image had intensities values 85 or 171, a 2 bit image had intensity values 51, 102, 154 or 205 and so on.

C. *Precision-Recall Calculation*

To generate precision-recall curves, we used the manually corrected GPS data provided by the authors of the original study [1]. Detected route segment matches were classified as correct if the spatial distance separating the central frames of each route was less than 40 meters, as in the original study. Matches outside this distance were classified as false positives, with missed matches classified as false negatives. Matches were assessed for both traverses of the route, rather than just the second traverse. To generate each precision recall curve, we conducted a sweep over the range of matching threshold s_m values. The range of values was chosen such that for most experiments, a complete range of recall rates from 0% to 100% was obtained.

TABLE II. IMAGE SIZES

Dataset & Image Type	*Reduced Resolution*	*Width×Height*
Eynsham panoramic images (original image 829440 pixels, 1620×512)	4 pixels	2×2
	8 pixels	4×2
	32 pixels	8×4
	128 pixels	16×8
	512 pixels	32×16
Eynsham cropped images (original crop 4800 pixels, 80×60)	2 pixels	2×1
	4 pixels	2×2
	16 pixels	4×4
	64 pixels	8×8
	256 pixels	16×16
Rowrah	16 pixels	4×4
Pikes Peak	32 pixels	8×4
Office NXT	2×7 bit pixels	2×1

V. RESULTS

We present a range of results evaluating the performance of the system with varying image resolution, sequence length, pixel bit depth, and field of view. This extensive testing is performed on the 70 km Eynsham dataset, for which both ground truth and a state of the art comparison is available. We also present additional results demonstrating qualitatively the applicability of the technique to three other varied datasets to demonstrate the wide applicability of the technique.

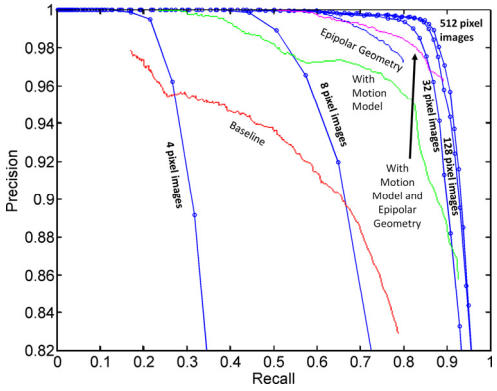

Figure 8. Precision recall curves for a range of reduced resolution panoramic images, with performance compared to four FAB-MAP implementations. Note the axis ranges.

A. *Precision-Recall*

The precision-recall performance using panoramic images from the Eynsham dataset is shown in Figure 8. At high precision levels, 4 pixel images produce superior performance to the baseline FAB-MAP performance. Increasing the resolution to 8 pixels enables the system to overtake the FAB-MAP with motion model results, while with 32 pixels

performance is superior (50% recall at 100% precision) to FAB-MAP with motion model and epipolar geometry, except between 86% and 89% recall rates. The sequence-based technique is also able to attain 100% recall, at 24%, 39%, and 46% precision levels for 4, 8 and 32 pixel images respectively. Figure 9 shows a zoomed in comparison of the techniques. Performance gains are minimal above an image size of 32 pixels.

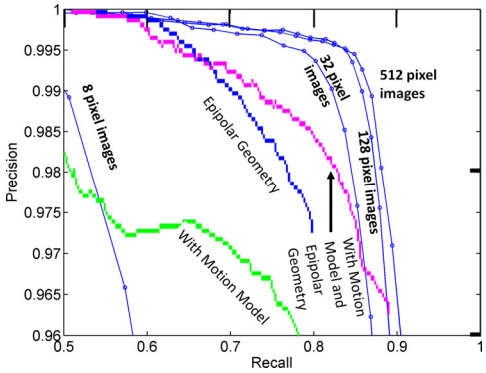

Figure 9. Enlarged plot of the high precision and recall performance curves. Note the rapidly reducing performance gains for image sizes above 32 pixels.

B. *Loop Closure Locations*

Although the precision performance using 32 pixel images is higher for a given recall rate compared to FAB-MAP, this is counteracted partly by inferior spatial loop closure coverage. Consequently, the algorithm requires a higher recall rate to achieve the equivalent loop closure coverage to FAB-MAP. Figure 10 shows the loop closures achieved at a 99% precision level, showing comparable loop closure coverage to the FAB-MAP algorithm. The sections of the route where the algorithm failed to match a route were mostly due to the linear search constraints not finding sequence matches when frame capture spatial frequencies varied significantly. The reverse route is also only thoroughly recognized at higher recall rates (and precision levels below 100%). Further discussion of this issue is provided at the end of the paper.

C. *Sample Route Matches and Speed Ratios*

Figures 11-12 show matched sub-routes for both a forward (11) and reverse (12) sub-route match. The section of the image difference matrix in which the sub-route match was found is shown in panel (a), with white circles indicating the matching locations of five representative images from the matched sub-route. Panels (b-c) show the corresponding images.

Figure 13 shows a histogram of the relative frame sampling speed calculated for all matched sub-routes at the maximum recall, 100% precision point. The peak around 1 shows that frames were spatially sampled at similar rates during the second traverse of the route, while the small peak at -0.8 indicates the sub-routes matched in reverse.

D. *Sequence Length*

Increasing sequence length had a positive effect on performance up to a point (Figure 14). Matching 10 frame sequences was clearly inadequate, but 20 frames provided performance superior at high precision levels to both the baseline and motion model FAB-MAP 2.0 performance. 50 frame sequences provided the best performance at high precision levels, while 100 frame sequences provided the highest recall at lower precision levels. The 50 frame sub-route length (335 meters) is consistent with the warm start localization times of commercial GPS navigation systems of 28.5 seconds [17] (380 meters at 30 miles per hour).

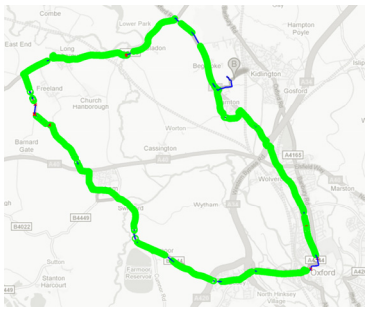

Figure 10. Loop closure map for the Eynsham dataset using 32 pixel images at 99% precision and 82% recall. Loop closures are shown by the thicker green circles, with false positive matches shown by red crosses.

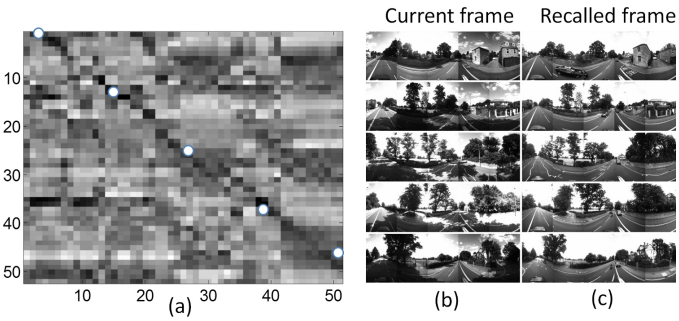

Figure 11. (a) Image difference matrix for a matched sub-route, with white circles showing the corresponding matching frame pairs. The matching gradient was approximately 1, indicating this route segment was traversed at the same speed both times. (b) Frames from the second traverse and (c) matching frames from the first traverse of the route. The full frames are shown for visibility, although the actual processed images were low resolution versions of the top 75% of the frame.

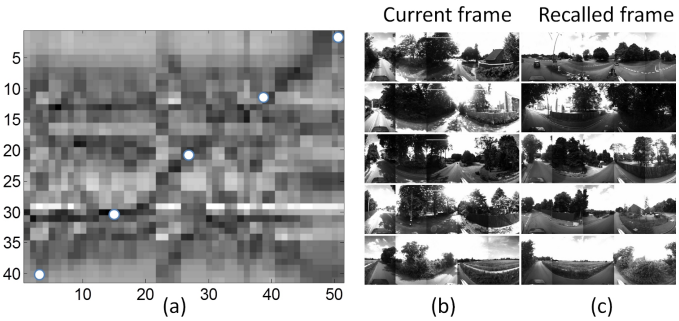

Figure 12. (a) Image difference matrix for the reverse traversal along a section of route. (b) Frames from the second traverse and (c) matching frames from the first traverse of the route. The matching route segment had a gradient of approximately -0.8, indicating that the route was traversed approximately 20% more slowly in reverse.

E. *Pixel Bit Depth*

The pixel bit depth had little effect on system performance beyond 2 bits (4 possible intensities). At a pixel depth of 2 bits,

performance was superior to the best FAB-MAP 2.0 performance at high precision levels, but had lower recall rates at lower precision levels. There was negligible difference in performance between 4 and 8 bit depths. A sequence length of 50 frames was used for all the pixel bit depth results.

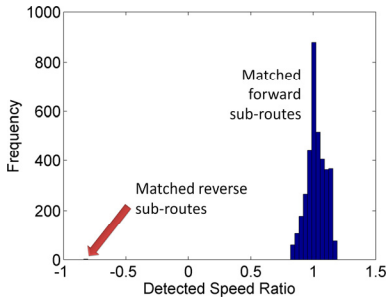

Figure 13. Relative speed ratios calculated for matching Eynsham sub-routes at the 100% precision, maximum recall point.

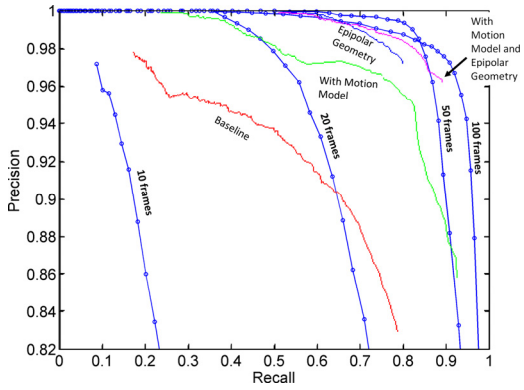

Figure 14. Precision-recall performance using 32 pixel images to match sub-routes of 10, 20, 50 and 100 frames in length.

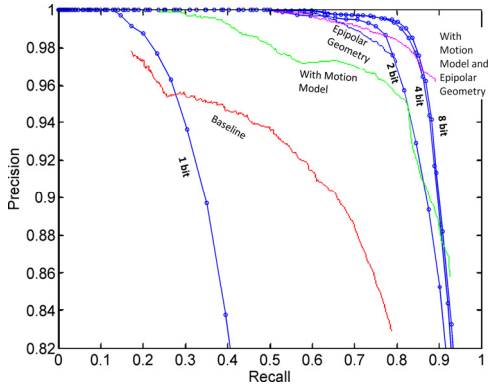

Figure 15. Precision-recall performance using 32 pixel images with 1, 2, 4 and 8 bit grayscale pixel depths. There was no significant improvement in performance above 4 bit pixel depths.

F. *Limited Field of View*

When limiting the field of view to 0.4% of the original panoramic image, the recall rate at 100% precision *improved* to 56% for 16 pixel images, 67% for 64 pixel images, and 69% for 256 pixel images. The effect of increasing resolution was broadly the same as for the full field of view images, with gains rapidly diminishing after reaching about 32 pixels in resolution.

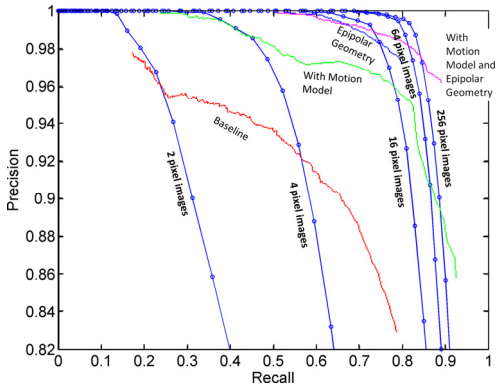

Figure 16. Precision recall curves with a limited field of view equivalent to 0.4% of the original panoramic image (see Figure 6).

G. *Rowrah Track Bike Dataset*

For the Rowrah, Pikes Peak and Office NXT datasets we did not have metric ground truth. Instead, we present the sub-route recognition graphs at a performance level that was qualitatively assessed to be near 100% precision, by comparing the image sequences associated with the matched sub-routes. For the Rowrah dataset, all sub-routes were matched within a few frames of the correct location (each frame separated by an average of 8 meters), as shown in Figure 17a. The vertical axis shows the central frame number of the matching sub-route from the first traverse of the route. Figure 17b shows five frames obtained by evenly sampling a sub-route from the second traverse of the circuit, with Figure 17c showing the corresponding frames from the matching sub-route during the first circuit traverse.

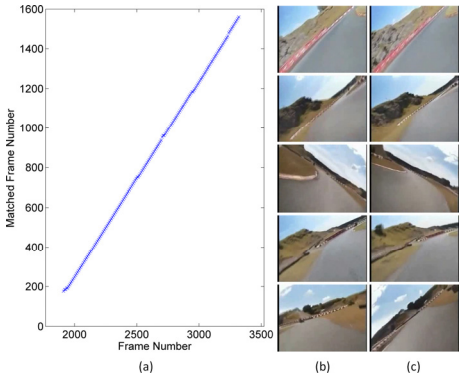

Figure 17. (a) Sub-route localization for the Rowrah dataset. (b) Frames from second route traverse and (c) matching frames from the first traverse.

H. *Pikes Peak Dataset*

The recall level at near 100% precision was around 50% for the Pikes Peak dataset (Figure 18). The lower recall rate was most likely due to significant changes in the racing-line taken by the car, larger variations in vehicle speed and the relatively bland nature of the environment, especially in the later mountainous stages.

I. *Office NXT Dataset*

The route matching performance for the NXT dataset is shown in Figure 19a. Figures 19d and 19e show the NXT light

sensor readings, with illustrative camera frames shown in Figures 19b-c. The first column (d) shows the pairs of light sensor readings obtained at that location during the second traverse, and the second column (e) shows the matched light sensor readings from the first route traverse. Manual inspection of the camera frames verified that every matched route segment was accurate to within approximately a meter.

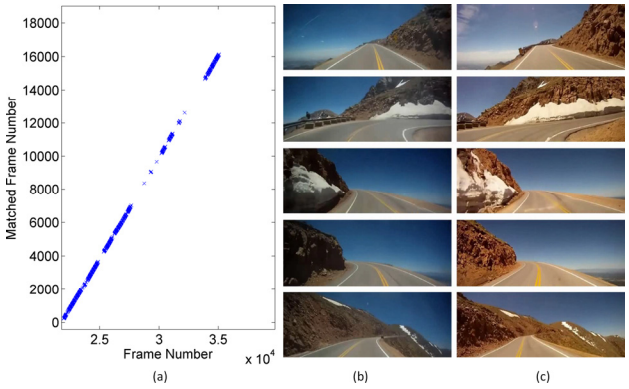

Figure 18. (a) Sub-route recognition for the Pikes Peak dataset. Recall was not achieved in sections of the 20 km route, most likely due to large variations in racing-line. (b) Frames from second traverse and (c) matching frames from the first traverse of the route.

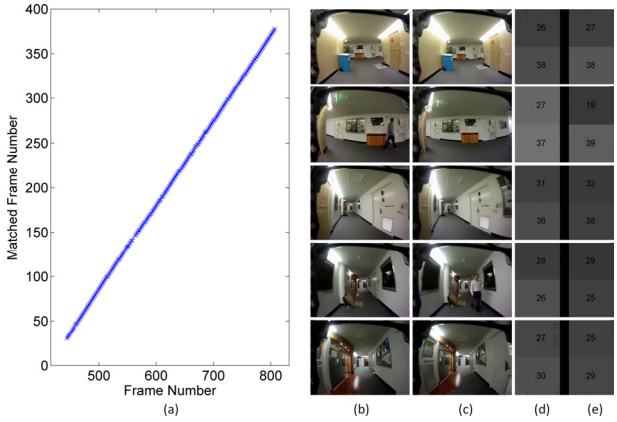

Figure 19. (a) Sub-route recognition for the Office NXT dataset. (b) Frames from second traverse and (c) matching frames from the first traverse of the route. (d) Light sensor readings from second traverse and (e) matching light sensor readings from the first traverse of the route.

J. Computation and Storage

In this section we describe the storage and computational requirements of the system and present a range of computational scenarios for the proposed visual GPS scenario.

1) Storage

At a pixel depth of 4 bits and an image size of 32 pixels, each stored image takes a total of 16 bytes. It is an informative exercise to calculate the storage required to store imagery from the entire global road network. According to the CIA World Factbook [18], the 191 countries surveyed have a total of approximately 70 million kilometers of paved and unpaved roads. Storing images every 5 meters of road would result in 224 gigabytes of data, easily stored on a hard drive dating from 2006 and current solid state media. Therefore it would be possible to store a global database of images either locally on a device, or easily download imagery from local areas. For example, a 10 megabyte download would provide 3000 km of road data, enough for a regional area.

2) Computation

All experiments performed in this paper were performed at real-time or better speed using unoptimized Matlab and C code. Computation is dominated by the search for matching sub-routes. The primary factor affecting computation is the allowance for varying velocities when conducting the matching route segment search. Here we discuss the localization-only scenario where a library of images already exists, starting with the situation where self-motion information (such as car odometry) enables the search space to be constrained to a simple linear search.

For a sub-route length of n frames, the dominant calculation is the nm frame comparisons that must be performed, where m is the number of visual templates stored in the template library. Each frame comparison constitutes s byte-wise comparisons, or $2s$ comparisons to enable forward and reverse route matching. For 32 pixel panoramic images, this constitutes 64 byte-wise comparisons. Table III presents a number of computation scenarios, assuming 5 meter frame spacing and a camera speed of 15 meters per second (54 km/hr). With single instruction, multiple data (SIMD) the large city scenario is achievable on a current desktop machine. To achieve real-time performance during initial localization within an entire country or the world, significant optimizations would need to be implemented. One straightforward method is to cache frame by frame comparisons, comparing new images in the current sub route as they are seen, leading to an n times speed up, at the cost of needing more fast memory. However the fast memory requirement at the large city size is well within all current device capabilities including mobile phones and most other portable devices. Modern graphics card architectures and growing CPU counts even on mobile devices offer the potential for further significant speed ups through leveraging parallel processing. In addition, the implementation of optimized data structure methods could also remove the barrier to achieving country wide localization. Once localized, search spaces could also be massively constrained, as is done with current GPS systems. If a spatially regular spaced frame rate cannot be guaranteed, then the search space must expand to incorporate multiple possible velocities. This increases the compute by a factor dependent on the range of possible velocities. For the Eynsham dataset, allowing for a frame rate variation of 19% increased computation time by a factor of 10.

TABLE III. COMPUTATION SCENARIOS

Route Length	***Qualitative Description***	***Template Storage***	***Number of byte-wise calculations per second***	***Number of calculations with caching***	***Cache fast memory storage requirement***
100 km	Local area	320 kB	192×10^6	3.84×10^6	1 MB
10000 km	Large city	32 MB	19.2×10^9	384×10^6	100 MB
1×10^6 km	Medium country	3.2 GB	1.92×10^{12}	38.4×10^9	10 GB
7×10^7 km	World road network	224 GB	134×10^{12}	2.69×10^{12}	700 GB

VI. Discussion

The presented approach sacrifices single frame matching capability to achieve robust localization along a route, without the need for prior training or feature detection. While the method is not suitable for single snapshot mobile phone localization, a large range of potential applications involve navigation along routes, such as street navigation and indoor mobile robots. The experimental results show that in the context of navigation along routes, vision-based localization can be achieved with remarkably few pixels, tiny fields of view and reduced pixel bit depths.

Localization was achievable through two key, interdependent measures. Performing localization using a sequence of images rather than single image removes the requirement that the image matching scheme be able to reliably calculate a single global image match. Instead, the image matching front end must only on average report matches better than at chance. How much better depends on how long a matching sequence is used, with longer matching sequences reducing the performance requirements for the image matcher but increasing the computation of the sequence matching algorithm. This trade-off is avoided in a subset of real world navigation applications such as domestic car travel, where translational speed information is available from On Board Diagnostic (OBD) systems.

The use of sequences rather than individual images also introduces two types of lag – a delay in initial localization upon startup, and a delay when the route taken consists of several fragmented previously traversed sequences. Variable sequence length matching could partially address the initial localization lag problem, by localizing more rapidly when the environment is easily recognizable. To adapt the system to deal with fragmented sequences, we are pursuing three approaches. The first is to use traditional probabilistic filters, which also potentially removes the need for a sequence length parameter. The second is to expand the local best matching and search from one dimensional routes to two dimensional areas. The third is to maintain localization using odometry in situations where a previously localized system briefly loses localization while traversing several fragmented sequences (such as passing through a complex intersection).

Matching using sequences rather than individual frames allows the image matching algorithm to be modified in ways that would render it useless as a global image matcher. We exploit this ability by normalizing the image difference scores within local sub-routes, forcing the algorithm to calculate a best image match candidate within every section of route stored in the image database. While this measure would produce large numbers of false positive loop closures were single image localization used, by matching over a sequence of images it enhances the ability of the algorithm to localize by removing the effect of systematic biases, in much the same way that applying patch normalization to an image removes some of the effect of illumination change [19]. This combination of sequence matching and local image comparison provides a basis for future development of sequence-based, featureless localization techniques with capabilities complementary to single frame feature-based techniques.

Acknowledgements

This work was supported by an Australian Research Council Fellowship DE120100995 to MM. The author thanks Peter Corke, Gordon Wyeth and Liz Murphy for their helpful comments on the draft manuscript.

References

[1] M. Cummins and P. Newman, "Highly scalable appearance-only SLAM - FAB-MAP 2.0," in Robotics: Science and Systems, Seattle, United States, 2009.

[2] K. Konolige and M. Agrawal, "FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping," IEEE Transactions on Robotics, vol. 24, pp. 1066-1077, 2008.

[3] 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.

[4] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," Acoustics, Speech and Signal Processing, IEEE Transactions on, vol. 26, pp. 43-49, 1978.

[5] P. Newman, D. Cole, and K. Ho, "Outdoor SLAM using Visual Appearance and Laser Ranging," in International Conference on Robotics and Automation, Florida, United States, 2006.

[6] E. Kalogerakis, O. Vesselova, J. Hays, A. A. Efros, and A. Hertzmann, "Image sequence geolocation with human travel priors," in International Conference on Computer Vision, Kyoto, Japan, 2009, pp. 253-260.

[7] P. Newman, G. Sibley, M. Smith, M. Cummins, A. Harrison, C. Mei, I. Posner, R. Shade, D. Schroeter, L. Murphy, W. Churchill, D. Cole, and I. Reid, "Navigating, Recognizing and Describing Urban Spaces With Vision and Lasers," The International Journal of Robotics Research, 2009.

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

[9] D. A. Pomerleau, "Neural network perception for mobile robot guidance," DTIC Document1992.

[10] M. O. Franz, P. G. Scholkopf, H. A. Mallot, and H. H. Bulthoff, "Learning View Graphs for Robot Navigation," Autonomous Robots, vol. 5, pp. 111-125, 1998.

[11] D. Q. Huynh, A. Saini, and W. Liu, "Evaluation of three local descriptors on low resolution images for robot navigation," in Image and Vision Computing New Zealand, Wellington, New Zealand, 2009, pp. 113-118.

[12] V. N. Murali and S. T. Birchfield, "Autonomous navigation and mapping using monocular low-resolution grayscale vision," in Conference on Computer Vision and Pattern Recognition, Alaska, United States, 2008, pp. 1-8.

[13] M. Milford and G. Wyeth, "Persistent Navigation and Mapping using a Biologically Inspired SLAM System," International Journal of Robotics Research, vol. 29, pp. 1131-1153, 2010.

[14] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in International Conference on Control, Automation, Robotics and Vision, Singapore, 2005, pp. 947-954 vol. 1.

[15] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in Computer Vision and Pattern Recognition, Anchorage, United States, 2008, pp. 1-8.

[16] M. Milford and G. Wyeth, "SeqSLAM: Visual Route-Based Navigation for Sunny Summer Days and Stormy Winter Nights," in IEEE International Conference on Robotics and Automation, St Paul, United States, 2012.

[17] J. Mahaffey, "TTFF Comparisons," ed, 2003.

[18] CIA, The World Factbook, 2012.

[19] A. M. Zhang and L. Kleeman, "Robust Appearance Based Visual Route Following for Navigation in Large-scale Outdoor Environments," The International Journal of Robotics Research, vol. 28, pp. 331-356, 2009.