The International Journal of
Robotics Research
32(7) 766–789

DOI: 10.1177/0278364913490323
ijr.sagepub.com

Vision-based place recognition: how low can you go?

Michael Milford

Abstract

In this paper we use the algorithm SeqSLAM to address the question, how little and what quality of visual information is needed to localize along a familiar route? We conduct a comprehensive investigation of place recognition performance on seven datasets while varying image resolution (primarily 1 to 512 pixel images), pixel bit depth, field of view, motion blur, image compression and matching sequence length. Results confirm that place recognition using single images or short image sequences is poor, but improves to match or exceed current benchmarks as the matching sequence length increases. We then present place recognition results from two experiments where low-quality imagery is directly caused by sensor limitations; in one, place recognition is achieved along an unlit mountain road by using noisy, long-exposure blurred images, and in the other, two single pixel light sensors are used to localize in an indoor environment. We also show failure modes caused by pose variance and sequence aliasing, and discuss ways in which they may be overcome. By showing how place recognition along a route is feasible even with severely degraded image sequences, we hope to provoke a re-examination of how we develop and test future localization and mapping systems.

Keywords

place recognition, vision, SeqSLAM, low resolution, vision-based place recognition

1. Introduction

From the 1970s to the early 1990s, computer vision research was dominated by limitations in computational resources and sensor capabilities. Consequently, some of the most famous research in vision-based navigation was conducted using low-resolution grayscale images, enabling surprisingly good results in fields such as autonomous car driving (Pomerleau, 1992). Fast forward two decades and computer vision researchers can now access an enviable array of sensing, computational and storage hardware, including mass produced multi-megapixel digital camera sensors and computers that are four orders of magnitude more powerful in terms of compute speed and storage.

Not surprisingly, many current state-of-the-art computer vision algorithms make good use of these sensor and computational advances. Vision-based localization and mapping systems generally follow the same process: images are processed, features are detected, statistical confidence measures are determined and place or image recognition likelihoods are calculated, sometimes in order to generate a map. Images are typically obtained from sensors ranging from video graphics array (VGA) resolution web cameras to top of the line industrial cameras such as the Ladybug 3. In general, the drive in vision-based mapping and localization research has been towards producing increasingly accurate maps of ever larger environments. Systems, such as FAB-MAP (Cummins and Newman, 2009), FrameSLAM (Konolige and Agrawal, 2008), Cat-SLAM (Maddern et al., 2012), RatSLAM (Milford and Wyeth, 2008) and MonoSLAM (Davison et al., 2007), are capable of data association and in some cases mapping on journeys through real-world environments of up to 1000 km in length (Cummins and Newman, 2009).

This quest towards mapping the largest environment possible with the highest degree of accuracy has been ably supported by the still rapid rate of improvement in camera and computer technology. However, in this quest, we believe that one important area of vision-based localization remains poorly understood, one which has been addressed in other fields such as face recognition (Phillips et al., 2005) and text recognition (Mirmehdi et al., 2003), but that has perhaps been largely overlooked in this age of multi-megapixel cameras.

Queensland University of Technology, Brisbane, Queensland, Australia

Corresponding author:
Michael Milford, Queensland University of Technology, Brisbane, Queensland, Australia 4001.
Email: michael.milford@qut.edu.au

In this paper, we seek to answer that question, which is as follows:

> How little and what quality of visual information is needed to conduct effective vision-based place recognition?

We attempt to answer this question within the domain of path-like environments using the SeqSLAM algorithm first presented in (Milford and Wyeth, 2012). The paper title "how low can you go" refers not only to the quantity of visual information but also to its sophistication – at its core SeqSLAM uses a simple sum of absolute differences (SAD) matcher to compare low-resolution, intensity normalized images. To perform the analysis in this paper, we made a number of assumptions about the scope:

- We do not incorporate self-motion information, which means SeqSLAM performs only the place recognition component of localization but not the "path integration" or "dead-reckoning" component. We discuss ways to incorporate self-motion information throughout the paper, and speculate that self-motion information would radically improve system performance.
- Testing is constrained to path-like environments, which encompasses a wide range of robotic and computer vision applications, such as indoor service robots and outdoor autonomous cars. We provide a brief study showing the effect of pose variance on system performance.
- Our "mapping" process is trivial in that it learns places at a steady rate regardless of camera speed or scene familiarity and has unbounded linear growth in compute over time. However real-time speed or better performance is achieved for even the largest datasets analyzed, and we discuss methods for bounding compute.

Within this scope, we present evidence to suggest that place recognition performance using a *sequence* of very low-resolution, poor-quality images is comparable to or superior to the state-of-the-art algorithms using *single* (or very short sequences of) high-quality imagery.

Specifically, we present the following findings and contributions, all for sequences of images with individual resolutions of 512 pixels or less:

- Place recognition performance comparable or superior to FAB-MAP 1.0 (FAB-MAP 2.0) at 100% precision with *any one of* :
  - sequences of 50 images with 4-pixel (32-pixel) resolutions;
  - sequences of 20 (50) images with 32-pixel resolutions;
  - sequences of 50 images with 1-bit (2-bit) pixel depths; and
  - sequences of 50 images with 1/250th the field of view (FOV) of the original Eynsham panoramic images at 2-pixel (16-pixel) resolutions.
- Low-resolution place recognition on motorbike circuit and mountain rally car stages.
- General place recognition using image sequences:
  - with extreme motion blur equivalent to camera exposures of up to 10 seconds in duration on a platform moving at 55 km/h;
  - across day–night cycles with illumination changes an order of magnitude larger than ever achieved before using consumer hardware;
  - with high image compression ratios; and
  - using two 7-bit single-pixel Lego NXT light sensors over multiple floors of an office environment.
- Case studies of failure modes caused by pose variance and sequence aliasing.
- An analysis of the statistical characteristics of individual image matching with low-resolution images that provides insight into why the sequence-based algorithm works.

The research presented here builds on the initial studies performed in (Milford, 2012). New contributions include the image region utility study, the motion blur studies, the image compression studies, a 1-pixel image study, the pose variance and sequence aliasing case studies, analysis of the statistical distribution of individual image matching scores over the entire Eynsham dataset and a significantly larger NXT experiment that involves both forward and backward recognition of a route.

The primary insight from the work, that localization using sequences of images can be performed with low-quality individual images, has a secondary implication that a navigation system using such an approach could localize successfully if provided with a map storing a minimal amount of image data. To provide an illustration of how little stored data is required to localize, we have combined the image data used in all seven datasets presented in this paper (at the resolutions at which performance was comparable to the state of the art) into one single 2 megapixel image, shown in Figure 1(a). The combined image is similar in size to a single original panoramic camera image from the Eynsham dataset (Figure 1(b)), but contains sufficient imagery to localize along 131 km of road travel and 1160 m of indoor travel.

The paper proceeds as follows. In Section 2, we review the background of general vision-based localization algorithms, touching on the use of sequences for matching and low-resolution computer vision algorithms. Section 3 describes the SeqSLAM algorithm, which is our chosen place recognition method. In Section 4, we describe the main focus of the paper, the extensive set of studies that were performed. Results corresponding to each study are presented in Section 5, as well as some analysis of compute and storage scaling scenarios, such as implementing a global camera-based global positioning system (GPS) system. Finally, we conclude the paper with a discussion and some conclusions.

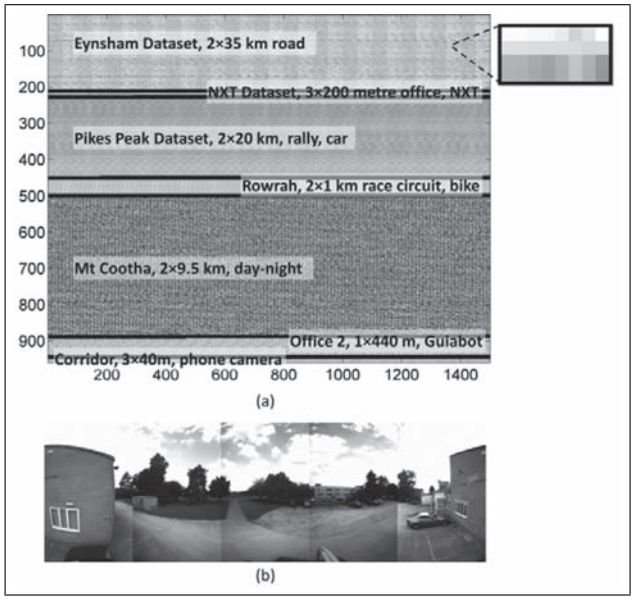

Fig. 1. (a) Place recognition across all of the datasets presented in this paper can be achieved by storing imagery data equivalent to that in a single 2 megapixel, 8-bit grayscale image. This image contains data from 131 km of car and motorbike driving and 1160 m of travel in an indoor office environment. (b) A single original panoramic image from the Ladybug 2 camera used to acquire the Eynsham dataset contains a similar amount of data.

2. Background

In this section, we first review the typical approach to vision-based localization, followed by methods that use sequence information, and finally cover algorithms that have used low-resolution sensory input.

The observation that most current vision-based localization and mapping techniques use features is perhaps not surprising, given their many advantages. Many of the state of the art feature detectors such as scale-invariant feature transforms (SIFT) (Lowe, 1999) and speeded up robust features (SURF) (Bay et al., 2006) exhibit desirable properties such as scale and rotation invariance, and a limited degree of illumination invariance. Feature-based techniques that consider the geometry of features are also desirable because of their relatively easy integration with metric pose estimation algorithms. For robot applications that involve scene understanding or semantic mapping, feature-based techniques are also suitable given their common heritage with object and scene recognition techniques. However, despite their advantages, these approaches also have shortcomings, and it is through these shortcomings that we provide some additional framing to the central question that we address in this paper.

All feature-based approaches require a feature detector that is suitable for the type of environment – features that are easily found in the middle of a modern city may not be suitable for forested environments and vice versa. Furthermore, some techniques such as FAB-MAP (Cummins and Newman, 2009) require a training stage, during which data that is known to be similar to the operational environment is processed in order to generate a codebook that stores the salience information about the combinations of features found in that environment. Using the wrong feature detector type or using an inappropriate codebook can result in poor system performance. If the environment changes, feature-based techniques may also fail since visual features can change radically over day–night, weather and seasonal cycles (Milford and Wyeth, 2012). Furthermore, although there has been some work towards "feature" detection using blurry images (Klein and Murray, 2008), feature-based techniques generally require relatively high-resolution, crisp and noise-free imagery, limiting their applicability in low illumination situations or on very small or toy optical sensors, scenarios addressed in this work.

Recent research using low-resolution images has shown the potential for vision-based recognition across much larger environmental changes than has previously been achievable (Milford and Wyeth, 2012). Furthermore, using pixel by pixel comparison techniques such as SAD has the advantage of not requiring assumptions about the type of visual features in an environment, and furthermore does not require training and selecting a suitable codebook for that environment. The main disadvantage of pixel by pixel recognition techniques is a lack of pose invariance compared to feature-based techniques. To some extent limited pose invariance can be introduced through comparing image snapshots over a range of offsets in order to enable mapping and localization on less path-constrained vehicles such as quadrotors (Milford et al., 2011).

The SeqSLAM algorithm attempts to match route segments rather than individual images, and is similar in concept to work by Newman et al. (2006), in which loop closure was performed by comparing sequences of images based on the similarity of 128D vectors of SIFT descriptors. Owing 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 (Kalogerakis et al., 2009). In contrast, the technique presented here forgoes the use of features and uses a novel image difference normalization scheme to partially address visual ambiguity. We also emphasize that the focus of this paper is not the optimality of the localization algorithm but rather the results – we are putting an *upper bound* on the minimal amount and quality of visual information required for place recognition, but it is quite possible an alternative system may require even less visual information.

Of the large number of vision-based mapping systems (Cummins and Newman, 2009; Davison et al., 2007;

Konolige and Agrawal, 2008; Newman et al., 2009; Sim et al., 2005), few that are currently active research areas use low-resolution images. Early computer vision researchers were forced by sensor and computation limitations to use low-resolution images for tasks such as map building (Franz et al., 1998), but they were able to achieve remarkably impressive performance in applications such as autonomous car driving with now famous systems such as the Autonomous Land Vehicle In a Neural Network (ALVINN) (Pomerleau, 1992). Although most algorithms now take full advantage of improved sensing technology, some relatively recent low-resolution techniques have been used on Sony AIBO robot dogs (Huynh et al., 2009) and Pioneer robots (Murali and Birchfield, 2008), including the biologically inspired RatSLAM system (Milford and Wyeth, 2008, 2010; Milford, 2008). However, we note here that while these systems take advantage of the usefulness of low-resolution imagery for localization, they do not constitute systematic studies of how different image properties affect localization. In non-localization research areas, impressive facial recognition performance has been achieved using low-resolution images (Phillips et al., 2005), both by humans and by algorithms. General image recognition using low-resolution images has also been found to be highly effective (Torralba et al., 2008).

3. Approach

In this section we describe the two key components of the SeqSLAM algorithm: image comparison and sequence recognition. In broad terms, the algorithm calculates difference scores between the recent sensory snapshots and every past sensory snapshot, and then searches for spatially coherent sequences of low difference scores. These low difference score sequences represent previously traversed route segments or sub-routes that match closely to the current route segment.

Conceptually, SeqSLAM performs both mapping and place recognition, albeit with some trivial components in the current implementation. The mapping process learns images at a fixed rate regardless of camera speed or scene familiarity, and is consequently unbounded in compute and storage respects. The resultant "map" is simply a database of images with topological links (as found by the sequence matching process) but no geometric properties like a conventional simultaneous localization and mapping (SLAM) system. The method performs only place recognition, rather than full localization through the use of motion information. In Section 6, we address ways in which SeqSLAM could integrate geometric self-motion information and bound map growth to create a more conventional SLAM system with a geometric map. We also note that there are no separate stages of mapping and localization and that all experiments in this paper were run with no initial map. However, all experiments were designed to have repeated segments in order to evaluate place recognition performance.

3.1. Image similarity

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

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

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

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

where s is the area in pixels of the image and p is the pixel intensity value. Setting $\sigma = [0, \pi]$ enables recognition when traversing a route in reverse with a panoramic sensor. 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 no offset case was used ($\sigma = [0]$). We also used $\sigma = [0, \pi]$ for the 2-pixel Lego NXT light sensor dataset to recognize routes in reverse.

It should be noted that in the current implementation, each new image is stored regardless of how closely it matches to any existing images, meaning compute grows linearly with time rather than the size of the environment. Because the image comparison compute is so trivial, real-time performance is easily achievable even over quite large datasets such as the 70 km Eynsham dataset. In Section 5, we discuss compute issues and options for future optimization.

3.2. Sequence matching

Comparisons between the current image and all stored images yield a vector of image differences, as in Newman et al. (2006). The matrix $\boldsymbol{M}$ of image difference vectors for the n most recent frames forms the space within which the search for matching route segments 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 2(a)). The updated image difference vectors (Figure 2(b)) 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 Rw images acquired before the current image i and σ_{min} is a minimum standard deviation constant used to avoid undefined output, set to the minimum discrete non-zero sensor reading possible (one for the images with 256 grayscale intensity possibilities and one for 100 NXT light sensor reading possibilities). R_w was set to 20 frames for all experiments in this paper. Normalizing the difference values in local image neighborhoods is a process that would be counterproductive when

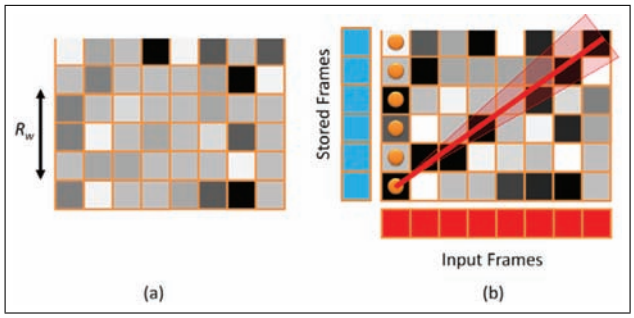

Fig. 2. The image difference matrix **M** (a) before and (b) after normalization, with small circles showing the elements where each search originates. Only the bottom section of the difference matrix is shown. A sample search range and matched trajectory is shown by the thick red line and the shaded region.

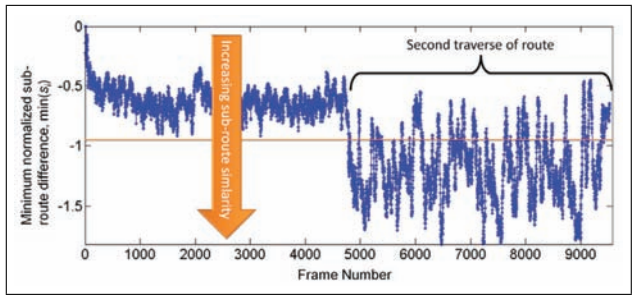

Fig. 3. Normalized sub-route difference scores for the Eynsham dataset with the matching threshold s_m that yields 100% precision performance.

attempting to find a single, global best match candidate, such as with FAB-MAP. 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. Section 5.9.10 provides quantitative analysis of the improvement in individual image matching performance that local neighborhood normalization provides.

To find likely route matches, we perform a continuous version of the dynamic time warping (DTW) method proposed by Sakoe and Chiba (1978), in essence finding straight lines that pass over low scoring elements in the matrix $\boldsymbol{M}$. 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 2).

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

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

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

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

The vector $\boldsymbol{d_i}$ 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 scenarios with a constant spatial separation between frames, such as the Eynsham dataset, it is possible to use a small range or even single value of v_k. In Section 5, we describe the benefit of using any available odometry signals to drastically reduce the compute requirements by constraining the slope search range.

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 3. In the second half of the dataset the route is repeated, leading to lower minimum matching scores.

If frames are taken far enough apart to be considered independent, then we can calculate the likelihood of a false-positive route match based on a given matching threshold. 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 a mean of zero and a variance of n. Dividing by the number of frames produces a normalized route difference score with a mean of zero and a variance of 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 threshold of -1 yields a 7.7×10^{-13} chance of the match occurring by chance. The extremely low probability does not of course take into account the large number of ways (in terms of varying camera velocities and paths) in which a 50-frame sequence of images can be formed. We also note that the frame independence assumption, which underpins other systems such as FAB-MAP, is not valid in most practical applications.

Table 1. Datasets.

Dataset name	Total distance	Number of frames	Distance between frames	Image storage
Eynsham	70 km	9575	6.7 m (median)	306 kB
Studies	Resolution reduction, FOV, pixel bit depth, image compression, sequence length			
Rowrah	2 km	440	4.5 m (mean)	7 kB
Studies	Resolution reduction			
Pikes Peak	40 km	4971	8 m (mean)	159 kB
Studies	Resolution reduction			
Mt Cootha	19 km	1473	12.9 m (mean)	1131 kB
Studies	Motion blur, day–night			
Office	600 m	4645	0.13 m (mean)	9.3 kB
Studies	NXT light sensor 2 pixel data			
Office2	440 m	1424	0.31 m (mean)	68.4 kB
Studies	Sequence aliasing example			
Corridor	120 m	353	0.3 m (mean)	16.9 kB
Studies	Lateral pose variance			

4. Experimental setup

In this section, we describe the seven datasets and associated studies, the sensor pre-processing for each study and how the performance metrics such as precision–recall curves were calculated.

4.1. Datasets

A total of seven datasets were processed, each of which consisted of at least two traverses of the same route. The datasets were as follows (*italics* indicate dataset names):

- two journeys along a 35 km road route in Eynsham, United Kingdom (*Eynsham*);
- two circuits of a 1 km of motorbike racing track in Rowrah, United Kingdom (*Rowrah*);
- two journeys along a 20 km off-road rally car stage up Pikes Peak in the Rocky Mountains, United States (*Pikes Peak*);
- two journeys along an unlit 9.5 km road circuit of Mt Cootha, Brisbane, Australia, one during the day and one during the middle of the night (*Mt Cootha*);
- two forward traverses and one reverse traverse of a 200 m path over three floors of an office building (*Office*);
- repeated traverses by a Guiabot of an office building floor over approximately 440 m (*Office 2*); and
- three traverses of an indoor corridor at various lateral offsets across the corridor (*Corridor*).

The Eynsham route was the primary dataset on which several of the studies were performed, providing both extensive quantitative performance data and comparison to several versions of the state of the art FAB-MAP and FAB-MAP 2.0 algorithms. The motorbike and rally car datasets were added to check whether the low-resolution place recognition capability extended to other datasets. The Mt Cootha datasets provide an extreme change in environment appearance an order of magnitude greater than that presented in Milford and Wyeth (2012). The Office dataset provides a means by which to test whether other types of "visual" data, namely the 7-bit single pixel intensity readings from two Lego NXT light sensors, provide enough information to perform place recognition. The Office 2 and Corridor datasets illustrate two of the failure modes of the system, sequence aliasing and pose variance. Key dataset parameters are provided in Table 1, including the storage space required to represent the entire dataset using low-resolution images, and Section 4.1.1 discusses generalization across datasets. For the latter datasets, metric ground truth was not available for a number of reasons including environmental conditions, topography, indoor operation or source (YouTube), so we instead present results for operation with no significant false positives, as assessed by a human manually parsing the videos and matches reported by the algorithm.

Figure 4 shows aerial maps and imagery of the Eynsham, Rowrah, Pikes Peak and Mt Cootha datasets, with lines showing the route that was traversed. The Eynsham dataset consisted of high-resolution image captures from a Ladybug 2 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 Pikes Peak, with the car dashboard and structure cropped from the images. The route consisted of heavily forested terrain and switchbacks up the side of a mountain, ending in rocky, open terrain partially covered in snow. 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 Mt Cootha dataset was obtained from both a Nikon D5100 taking long-exposure day and night-time images and a Logitech C910 camera taking short-exposure day-time images. The NXT dataset was obtained using a Lego NXT attached to two light sensors and a GoPro Hero camera (used for visualization), mounted to a chair which was pushed around the office building and carried up and

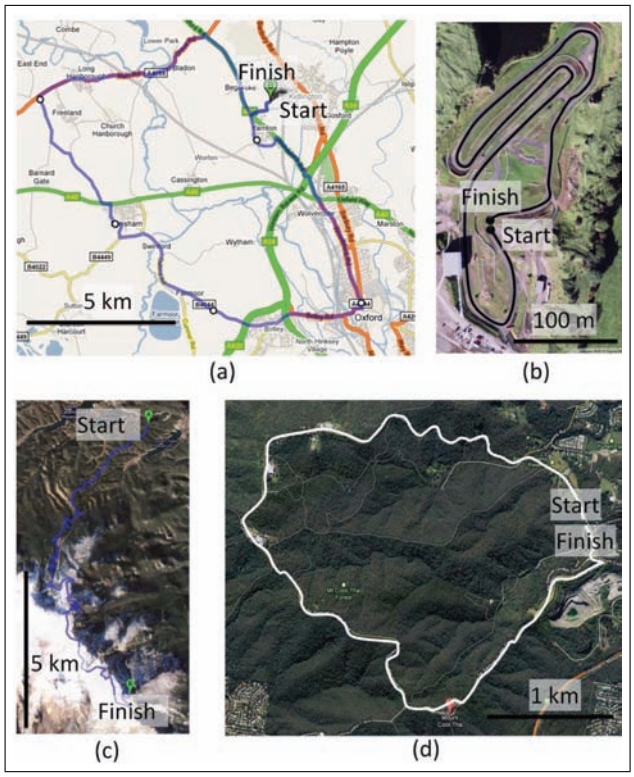

Fig. 4. The (a) 35 km Eynsham, (b) 1 km Rowrah (c) 9.5 km Mt Cootha and (d) 20 km Pikes Peak environments, each of which was repeated twice.

Source: Reproduced with permission from DigitalGlobe, GeoEye, Getmapping plc, The GeoInformation Group, USDA Farm Service Agency, Infoterra Ltd and Bluesky. Map data © 2012 Google.

down stairs (Figure 5). The Office 2 and Corridor datasets were obtained from cameras mounted on a Guiabot robot and Samsung Galaxy S3 phone. Extension 1 contains all of the datasets that were generated for use in this paper along with links to the remaining publicly available datasets.

4.1.1. Generalization and changes across datasets The major manual changes in the experimental setup and operating parameters from dataset to dataset were the resolution reduction, cropping and frame sub-sampling of the videos, and the choice of a hard-coded frame sequence length for performing matching. In comparison to similar work across robotics and computer vision, the video pre-processing is reasonable, especially considering the vast diversity in video sources which ranged from high-resolution, panoramic industrial class cameras to low-resolution, low-quality consumer cameras mounted haphazardly on a motorbike. However, the manual choice of the sequence length parameter is a significant obstacle to autonomous deployment of this system, and we propose solutions to overcome this limitation in Section 6.

4.2. Sensor pre-processing

4.2.1. Eynsham resolution reduced panoramic images For the Eynsham dataset, image processing consisted of image concatenation and resolution reduction (Figure 6). 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 2.

Table 2. Image sizes.

Dataset and image type	Reduced resolution	Width×height
Eynsham panoramic images (original panoramic images 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
	64 pixels	8×8
	256 pixels	16×16
Rowrah	16 pixels	4×4
Pikes Peak	32 pixels	8×4
Mt Cootha	768 pixels	32×24
Office NXT	2×7 bit pixels	2×1
Office 2 (sequence aliasing)	48 pixels	8×6
Corridor (lateral pose variance)	48 pixels	8×6

4.2.2. Reduced field of view For the initial reduced FOV experiment, a small area representing 0.4% of the total panoramic image was extracted from the center of the forward facing image (Figure 7). The resultant 80×60 pixel image was then resolution reduced to the sizes shown in Table 2. Based on the high performance of that region, we then divided the image up into 15 equal regions and ran the SeqSLAM algorithm on resolution reduced 8×2 pixel images of each region.

4.2.3. Reduced pixel depth Reduced pixel depths were obtained by reducing the bit depths of each pixel in the resolution reduced images (Figure 8). 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.

4.2.4. Mt Cootha motion blur and day–night experiments As in previous day–night experiments, we performed patch normalization (subtracting the mean value of the surrounding pixel intensities then dividing by the standard deviation of the surrounding pixel intensities) on the resolution reduced images, and manually cropped the different camera images to approximately corresponding FOVs. Example patch normalized images are shown in Section 5. Datasets consisted of both artificially generated motion blur images

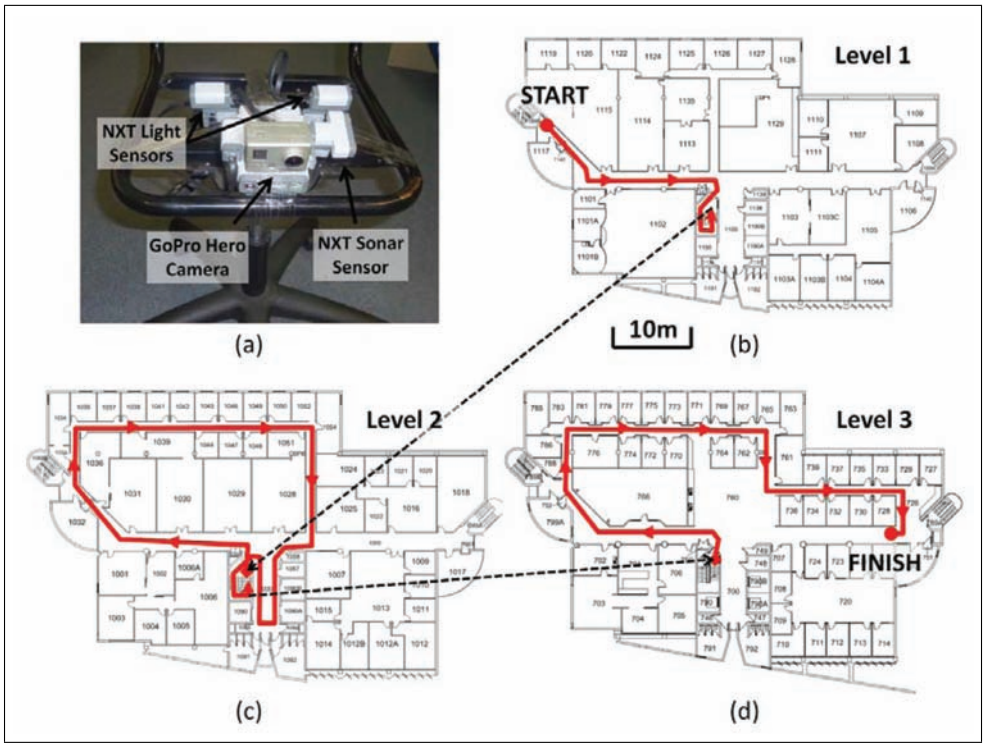

Fig. 5. (a) The Lego Mindstorms dataset acquisition rig with two sideways facing light sensors and GoPro camera for evaluation of matched routes. (b)–(d) The approximately 200 m long route which was traversed twice in a forward direction (as marked) and once in reverse.

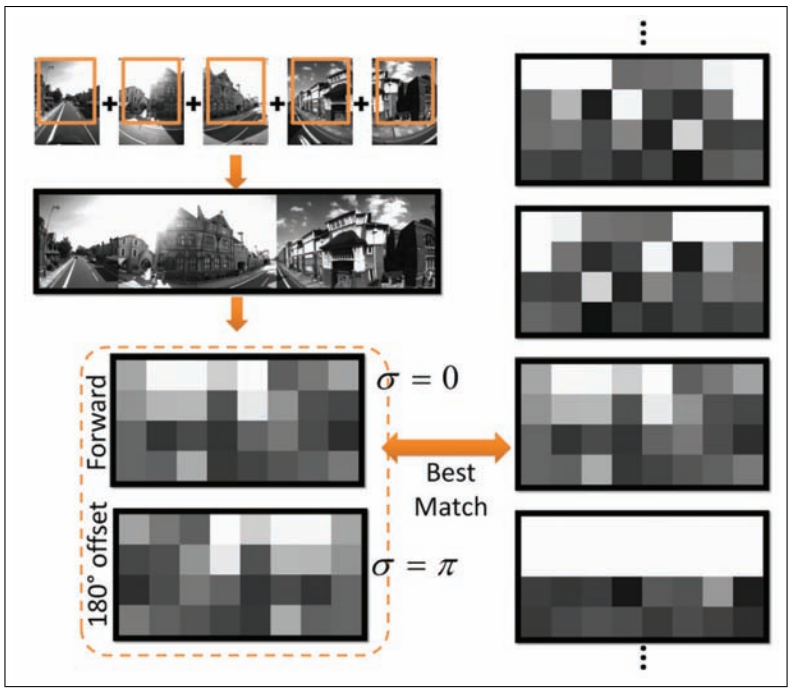

Fig. 6. Image pre-processing for the full panoramic images consisted of a crude image stitching stage followed by a reduction in image resolution. The current image was compared with 0° and 180° offsets to all stored images on a pixel by pixel basis to form the image difference matrix described in Section 3.2.

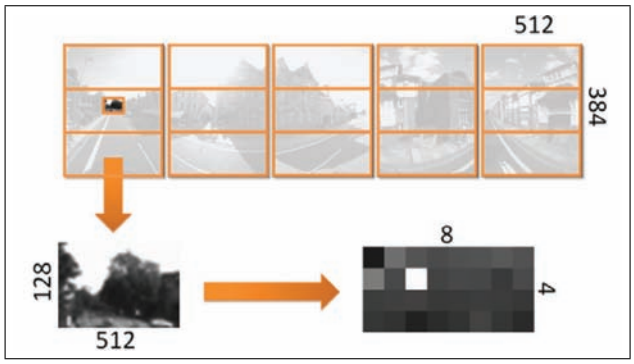

Fig. 7. To evaluate the effect of drastically reducing the FOV, an area representing 0.4% of the original panoramic image was extracted and the resolution was reduced (smallest single rectangle). We then divided the entire panoramic image into 15 equal regions and ran the SeqSLAM algorithm on 8×2 pixel images of each indvidual region.

and real long-exposure imagery (630 ms). To create arbitrary motion blur we combined the appropriate number of frames from a short-exposure dataset, such as combining together 150 frames from a 15 fps video to simulate 10 s exposures. Real long-exposure imagery was obtained using a Nikon D5100 camera both during the day and at night. To achieve a 630 ms exposure duration during the day, we used an ISO rating of 100 and a nine F-stop neutral density (ND) filter, which reduced the incoming light intensity by a factor of 512. At night, the filter was removed and the ISO rating was increased to 25,600. The 630 ms exposures were captured at a frame rate of 1 fps (rather than 1.6 fps) due to a slight storage and shutter delay between image captures.

4.2.5. Office NXT dataset Because the Office dataset consisted of both forward and reverse traverses along a route (and hence the left facing sensor would need to match data being produced in reverse by the right facing sensor), we multiplied the light intensities from one of the NXT light sensors by a factor of 0.9 to compensate for a bias in that sensor.

4.2.6. Office Guiabot dataset To illustrate the effect of aliased *sequences* of images in an environment, we ran SeqSLAM on a video dataset obtained from a forward facing camera on a Guiabot traversing the corridors of a university building (Figure 9(a)) (Murphy et al., 2012). The original video was down-sampled to two different resolutions; firstly to a resolution of 8×6 pixels, to show the effect of sequence aliasing, and also to 320×240 pixels, to test if significantly increasing the resolution would reduce the degree of aliasing.

4.2.7. Pose variant Corridor dataset To illustrate the effect of lateral pose variation we gathered three traverses of a corridor with a forward facing fish-eye camera (Figure 9(b)), one taken while hugging the left corridor wall, another along the center of the corridor and a third while hugging the right wall. We used a cheap ($20 USD) 190° FOV fish-eye lens attached to a Samsung Galaxy S3 camera.

4.3. Precision-recall and metric error calculations

Because the Eynsham dataset has accurate hand corrected GPS ground truth data, we were able to calculate precision–recall curves for all the studies involving that dataset. To generate precision–recall curves, we used the manually corrected GPS data provided by the authors of the original study (Cummins and Newman, 2009). Detected route segment matches were classified as correct if the spatial distance separating the central frames of each route was less than 40 m, 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.

For the motion blur experiments using the Mt Cootha datasets and the Office NXT experiments, we manually labeled matching frames to create the ground truth, using the GoPro images as an aid for the NXT datasets. Owing to the somewhat lower accuracy of this ground truth information, we generated average error statistics for runs in which no "significant" localization errors were produced (rather than precision–recall curves).

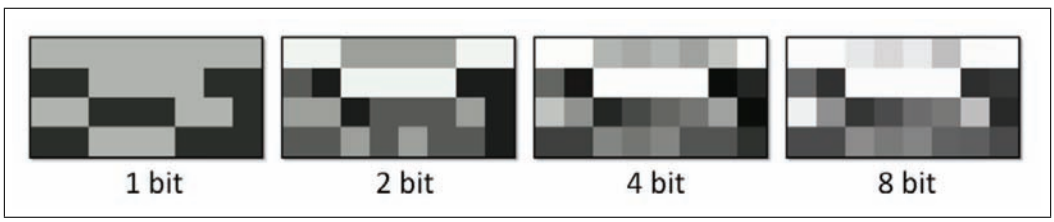

Fig. 8. To evaluate the effect of reduced pixel bit depth, the resolution reduced panoramic images were resampled at 1-bit, 2-bit, 4-bit and 8-bit grayscale pixel depths.

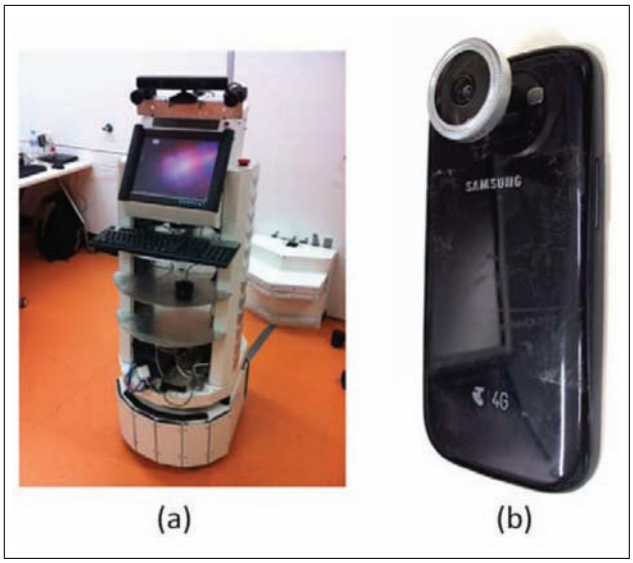

Fig. 9. (a) Guiabot platform with forward facing cameras and (b) 190° FOV fish-eye lens attachment for a Samsung S3 Galaxy phone camera.

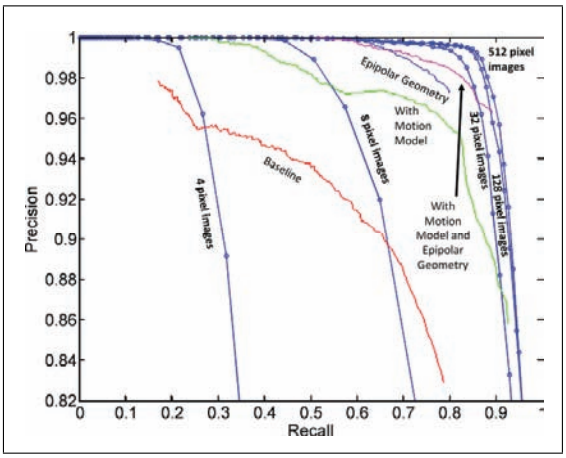

Fig. 10. Precision–recall curves for a range of reduced resolution panoramic images, with performance compared to four FAB-MAP implementations. Note the axis ranges.

5. Results

This section presents the results from the following studies, all evaluating the effect of a particular image property on place recognition performance:

- image resolution (panoramic images);
- sequence length;
- pixel bit depth;
- FOV;
- image compression;
- long-exposure-induced motion blur;
- 2-pixel Lego NXT light sensor study;
- image resolution (perspective images);
- sequence aliasing;
- lateral pose variance;

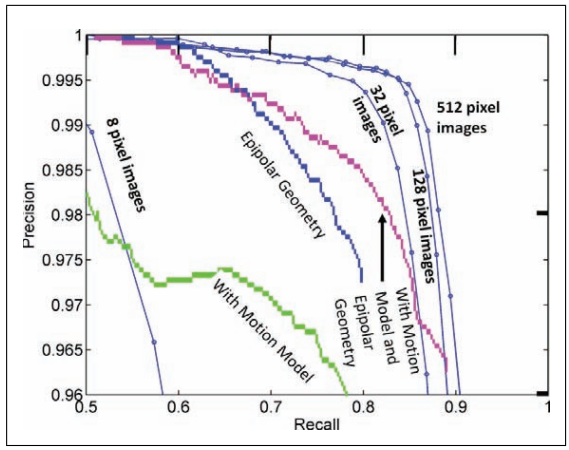

Fig. 11. Enlarged plot of the high precision and recall performance curves. Note the rapidly reducing performance gains for image sizes above 32 pixels.

- statistical analysis of individual image matching performance; and
- computation and storage.

5.1. Image resolution (panoramic images)

The precision–recall performance using panoramic images from the Eynsham dataset is shown in Figure 10. At high precision levels, using a sequence of 50 images with 4-pixel resolutions produces superior performance to the baseline FAB-MAP performance operating on single high-resolution frames. Increasing the resolution to 8 pixels enables the system to overtake the FAB-MAP with motion model results, while with 50-frame sequences of 32-pixel images 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 11 shows a zoomed in comparison of the techniques. Performance gains are minimal above an image size of 32 pixels.

Figure 12 shows the precision–recall curve for a 50-frame sequences of 1-pixel images. No significant recall rate is achievable without very high error rates, with only 1.6% of the route recognizable in an error-free manner.

5.1.1. Loop closure locations Although the precision performance using sequences of 32 pixel images is higher for a given recall rate compared to FAB-MAP, this is counteracted partly by inferior spatial loop closure coverage (significant gaps where no localization is achieved). Consequently, the algorithm requires a higher recall rate to achieve the equivalent loop closure coverage to FAB-MAP. Figure 13 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

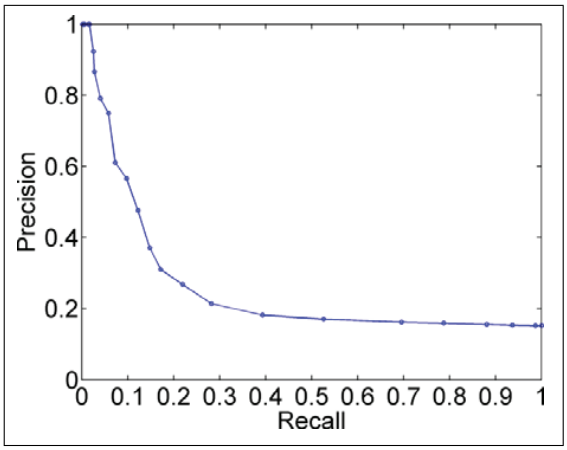

Fig. 12. Precision–recall curves using a 1-pixel image (note the different axis range), with a maximum of 1.6% recall achieved at 100% precision. No significant level of recall is achievable without precision dropping to a very low level.

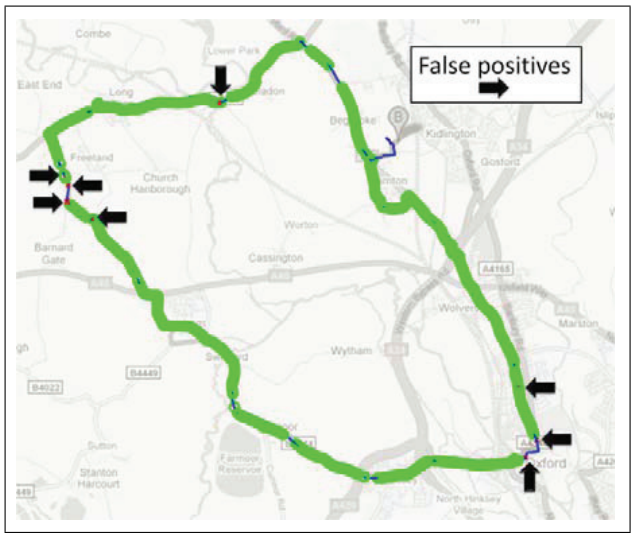

Fig. 13. Loop closure map for the Eynsham dataset using 32 pixel image sequences at 99% precision and 82% recall. Loop closures are shown by the large green circles, with false positives shown by red crosses and highlighted by black arrows.

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.

An example false positive sequence match is shown in Figure 14, which occurred when recall was increased and precision dropped below 100%. While the start and end frames look significantly different, the similarity between the middle frames in each sequence lead to the false positive match. Training on a typical environment dataset could possibly learn the saliency (or lack thereof) of these "open" areas with "flat" sky–ground transitions and increase the weighting applied to salient patches such as the black above horizon pixels caused by trees.

5.1.2. Sample route matches and speed ratios Figures 15 and 16 show matched sub-routes for both a forward (Figure 15) and a reverse (Figure 16) sub-route match. The section of the image difference matrix in which the sub-route match was found is shown in Figure 15(a), with white circles indicating the matching locations of five representative images from the matched sub-route. Figures 15(b) and (c) show the corresponding images.

Figure 17 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.

5.2. Sequence length

Increasing sequence length had a positive effect on performance up to a point (Figure 18). Matching 10 frame sequences was clearly inadequate, but 20 frames provided performance superior at high precision levels to both the baseline and the motion model FAB-MAP 2.0 performance. The 50-frame sequences provided the best performance at high precision levels, while 100-frame sequences provided the highest recall at lower precision levels. Increasing sequence length indefinitely does not result in continual performance improvement because the longer sequences violate the constant velocity assumptions upon which the current algorithms are built. The 50-frame sub-route length (335 m) is consistent with the warm start localization times of commercial GPS navigation systems of 28.5 s (Mahaffey, 2003) (380 m at 30 mph). However, compared with a single-frame localization system such as FAB-MAP, SeqSLAM is clearly at a disadvantage where localization latency is critical. In addition, with a hard-coded and long matching sequence length parameter, SeqSLAM is unable to localize on journeys containing short overlapping sections with previously traversed routes. For situations where those short overlapping sections contain distinctive visual information, a dynamic sequence length matching approach would overcome this second limitation.

5.3. Pixel bit depth

The pixel bit depth had little effect on system performance beyond 2 bits (four possible intensities) (Figure 19). At a pixel depth of 2 bits, performance using 50-frame sequences of 32 pixel images 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.

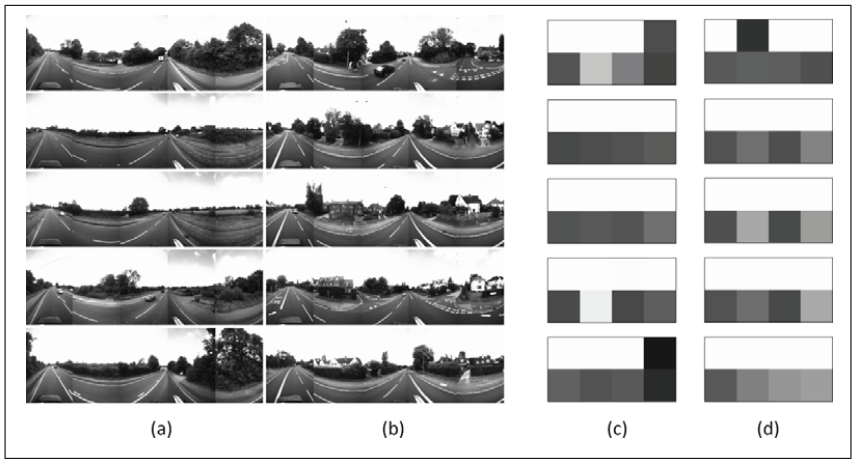

Fig. 14. (a) Example sequence which incorrectly matches back to (b) another sequence when precision drops below 100%. (c), (d) The corresponding 4×2 pixel images, with the high degree of similarity in the middle section of the sequence causing the incorrect match despite the different start and end frames.

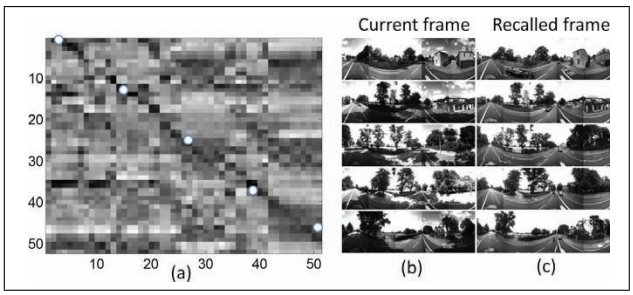

Fig. 15. (a) Image difference matrix for a matched sub-route, with white circles showing the corresponding matching frame pairs. The matching gradient was approximately 1, indicating this route segment was traversed at the same speed both times. (b) Frames from the second traverse and (c) matching frames 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.

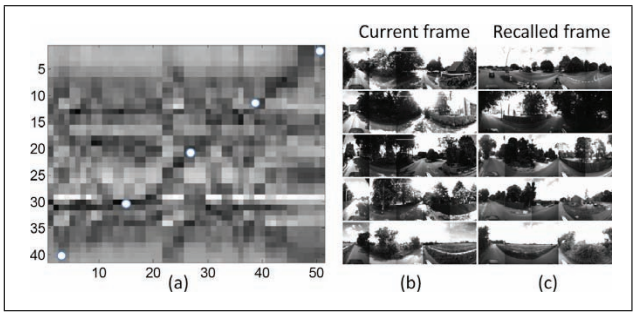

Fig. 16. (a) Image difference matrix for the reverse traversal along a section of route. (b) Frames from the second traverse and (c) matching frames from the first traverse of the route. The matching route segment had a gradient of approximately –0.8, indicating that frames were acquired at greater spatial intervals in that section of the route.

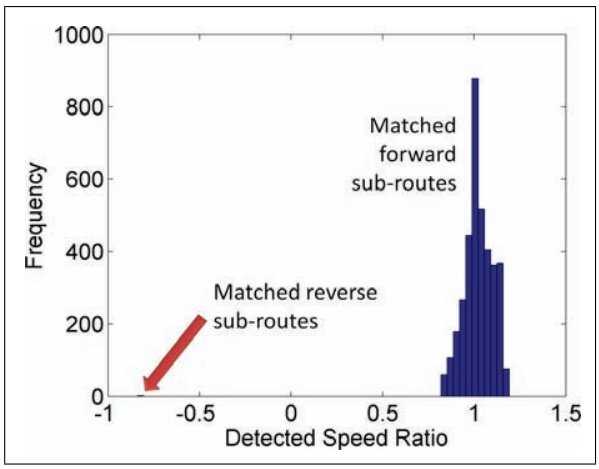

Fig. 17. Relative speed ratios calculated for matching Eynsham sub-routes at the 100% precision, maximum recall point.

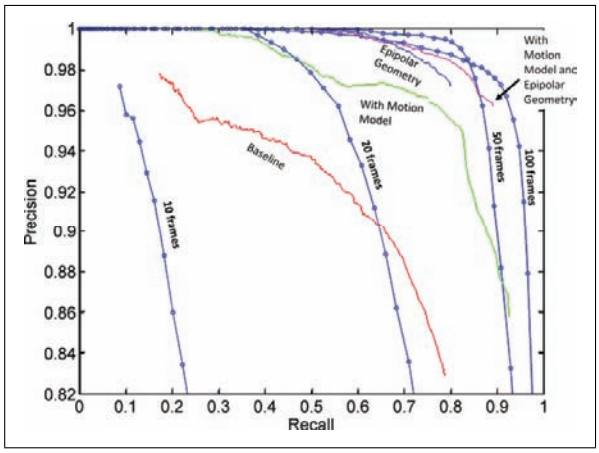

Fig. 18. Precision–recall performance using 50-frame sequences of 32-pixel images to match sub-routes of 10, 20, 50 and 100 frames in length.

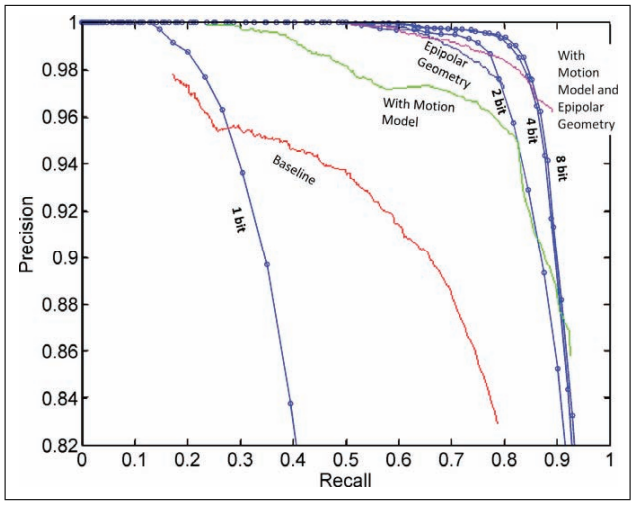

Fig. 19. Precision–recall performance using 50-frame sequences of 32-pixel images with 1-, 2-, 4- and 8-bit grayscale pixel depths. There was no significant improvement in performance above 4-bit pixel depths.

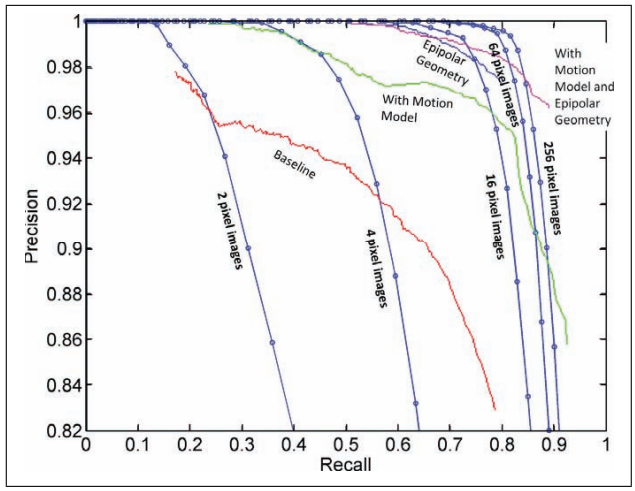

Fig. 20. Precision–recall curves with a limited FOV equivalent to 0.4% of the original panoramic image (see Figure 7).

5.4. Field of view

When limiting the FOV to 0.4% of the original panoramic image, the recall rate at 100% precision *improved* to 56% for 50-frame sequences of 16-pixel images, 67% for 64-pixel images and 69% for 256-pixel images (Figure 20). The effect of increasing resolution was broadly the same as for the full FOV images, with gains rapidly diminishing after reaching about 32 pixels in resolution.

The performance of SeqSLAM when run on the individual sub-regions was highly varied, as shown in Figure 21, which shows precision–recall curves and the maximum recall rates achieved at 100% precision. The middle row of the image yields very good matching performance with a slight drop off for sideways facing areas of the image. The bottom row comprising mostly road and the ground has somewhat poorer performance. The top row, which often contains only sky, has highly variable and generally poor localization performance, with the best performance achieved for the forward and rear facing regions of the image.

5.5. Image compression

Medium-resolution video (128×40) could be compressed quite severely before localization performance was adversely affected, as shown in Figure 22. A recall rate of around 44% at 100% precision using 50 frame sequences was achieved until the video compression ratio exceed a factor of 23, at which stage recall at 100% precision dropped significantly, although recall at lower precision levels was not as badly affected (see Table 3). In Section 5.11, we discuss the possibility of using highly compressed higher-resolution images rather than the (relatively incompressible) lower-resolution 8×4 pixel images.

Figure 23 shows the same frame at each of the different compression levels. It is interesting to note that the higher compression levels preserve the broad image contrast pattern but "blockify" the image, achieving an effect broadly similar to reducing the image resolution.

5.6. 2-pixel Lego NXT light sensor study

The route matching performance for the Lego NXT dataset is shown in Figure 24. This was the only study in which a longer sequence length of 200 frames was used (mainly due to the higher acquisition rate). The horizontal axis of the figure shows the (cumulative) frame numbers from the second and third traverses of the office environment, while the vertical axis shows the frame number that was deemed a match. The red line shows the ground truth.

The positive gradient during the second traverse indicates the route was traversed in the same direction as the first traverse. Almost the entire second traverse is successfully matched back to the first traverse, with no significant errors. The third traverse of the environment, which was performed in the reverse direction, matches back to both the first and second forward traverses. The overall recall rate across both the second and third runs was 79%, with recognition of traverses in the same direction generally more successful than in the reverse direction.

Figure 25 shows NXT light sensor profile readings that were matched for two sequence matches, one matching a forward sequence in the second run to a forward sequence in the first run (Figure 25(b)), and a second matching a reverse sequence in the third run to a forward traverse in the second run (Figure 25(d)). For illustrative purposes we also show corresponding frames from the GoPro camera (Figure 25(a) and (c)), although this imagery was not used to perform localization. The left column of each of the 5 frames in Figure 25(a) is evenly sampled over the sub-route

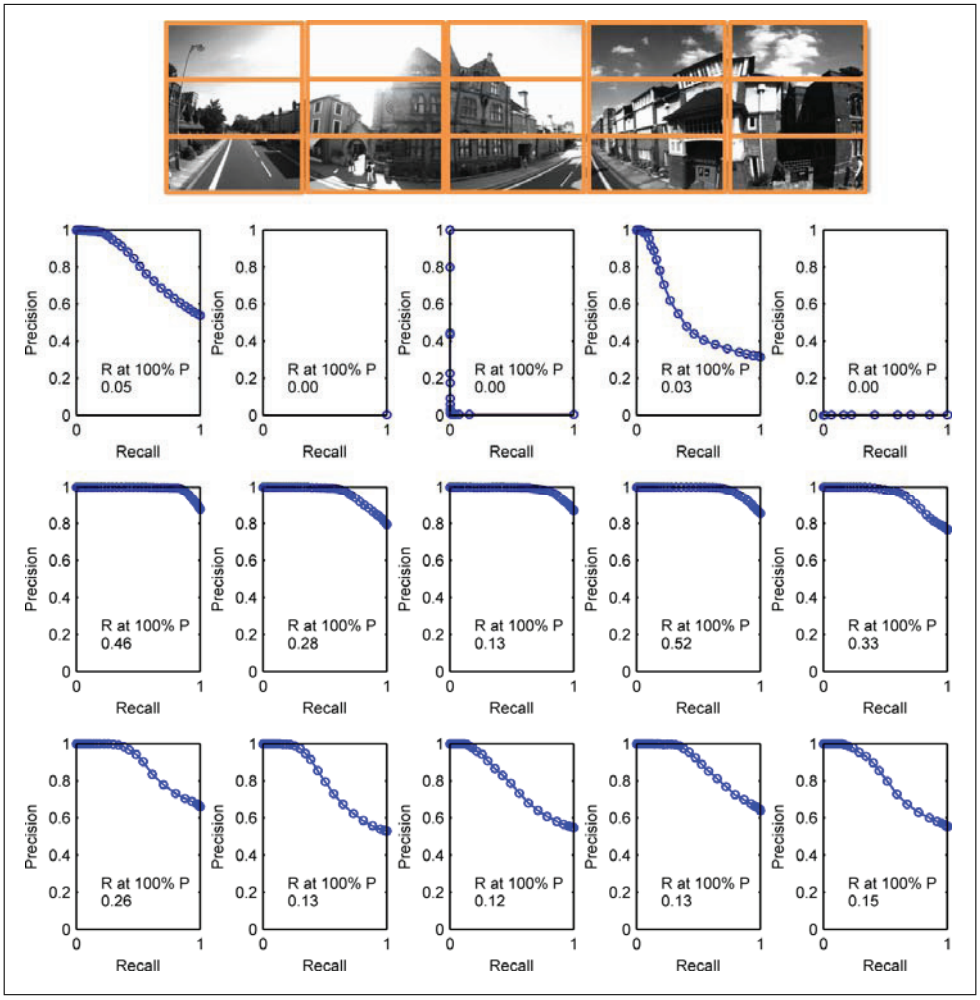

Fig. 21. Precision–recall curves and maximum recall at 100% precision values (numerical inset) when running SeqSLAM individually on each of the 15 image regions. The entire middle row of the image yields very good matching performance of up to 52% recall at 100% precision, with the bottom regions providing somewhat lower performance. The top row of regions is highly variable and generally poor, with the forward and reverse regions giving the best performance and the upper side facing regions failing to provide any matches at all.

from the second traverse that was matched back to the sub-route in the first traverse. The right column shows the corresponding frames from the matched sub-route in the first traverse.

In Figure 25(a) and (b), two relatively flat light sensor profiles were matched, and the corresponding camera frames show the correctness of the place match. Figure 25(c) and (d) are slightly harder to interpret because sequences from opposite directions of travel were matched, although the light sensor signal is clearly aligned as shown in Figure 25(d). The GoPro frames show, from top to bottom: looking in opposite direction from a stairwell landing, looking up and down while halfway up a set of stairs, looking in opposite directions on a lower stairwell landing, looking towards and away from an external building window in the elevator foyer and then a location closer to the window with the directions flipped (Figure 25(c)). Extension 2 shows examples of GoPro frames corresponding to matched NXT light sensor sequences.

Table 3. Compression results.

Compressed video size uncompressed (46 MB)	Compression ratio	Average image size (4.8 kB)	Recall at 100% precision
26 MB	1.8	2.7 kB	44%
5.7 MB	8.1	590 bytes	44%
3.0 MB	15	310 bytes	44%
2.0 MB	23	210 bytes	46%
1.7 MB	27	178 bytes	18%

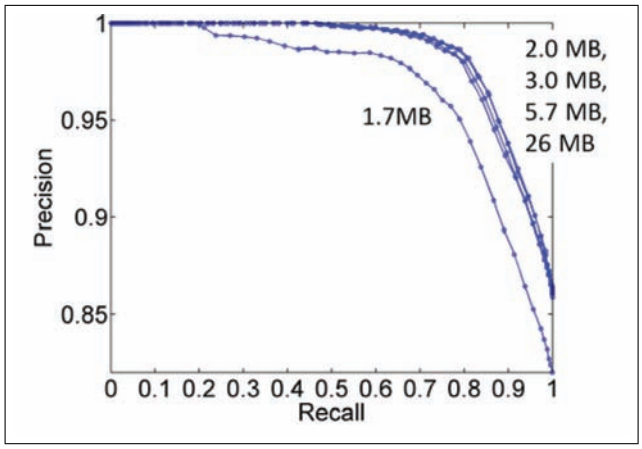

Fig. 22. Precision–recall curves for the Eynsham dataset using varying levels of video compression. Performance is virtually unchanged until the video is compressed by a factor of 27 times from 46 MB to 1.7 MB, at which stage recall at 100% precision drops to 18%.

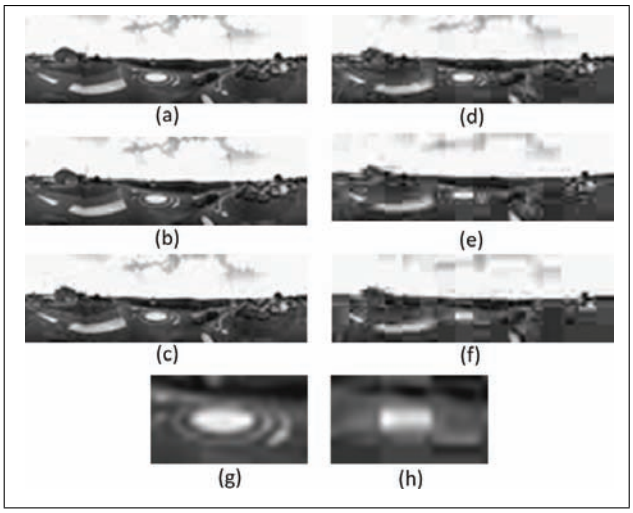

Fig. 23. A sample 120×40 frame which was successfully matched at all compression levels, ranging from no compression (a) to the highest compression (f). (g), (h) Enlargements showing the effect of compression on the center of the image, which still preserves the general contrast despite an almost complete loss of detail.

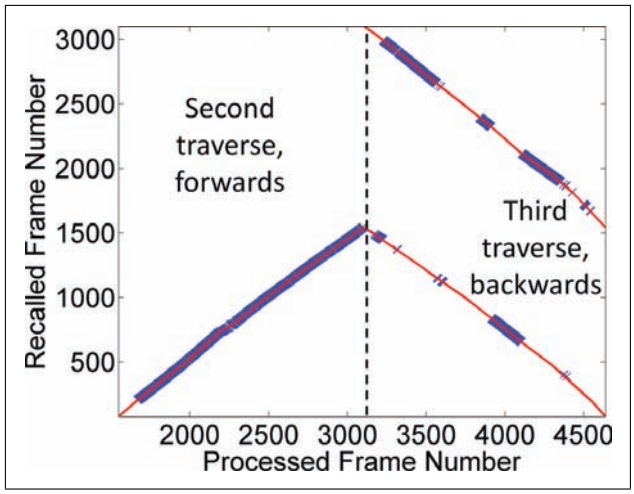

Fig. 24. Frame matches from the second (forward) and third (reverse) traverse of the three-floor office environment, overlaid on ground truth. Almost the entire forward second traverse is correctly matched back to the first route, while the third backwards traverse is matched back to both the first and second forward traverses. The overall recall rate was 79%.

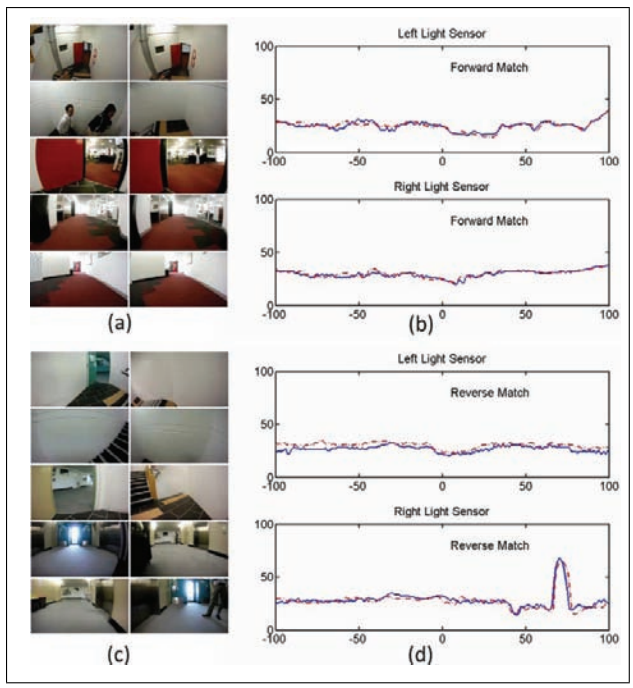

Fig. 25. Sub-route recognition for the Office NXT dataset. (a) Frames from matching sub-routes in the first and second traverses and (b) the matched left and right light sensor intensity profiles. (c) Frames from matching (opposing direction) sub-routes in the first and third traverses and the (d) matched left and right light sensor profiles (already adjusted for the reversal and left–right swapping of the signal).

5.7. *Long-exposure induced motion blur*

Localization was surprisingly robust to motion blur caused by simulated exposure durations of up to 10,000 ms. Figure 26 shows frame matches overlaid on ground truth for all six exposure durations. Only above 5000 ms do false positives start to creep in. Not surprisingly, because we were matching to fixed 630 ms exposure images, performance was best for the simulated 500 ms and 1000 ms exposure durations (Table 4). Note that the maximum recall achievable was 93.2% due to algorithm needing a full sequence length before localization could commence. Figure 27 shows sample original images and resolution reduced and patch normalized versions for one of the matched sequences in the 5000 ms study.

The gradually increasing mean and maximum localization errors for 1000 ms and longer durations

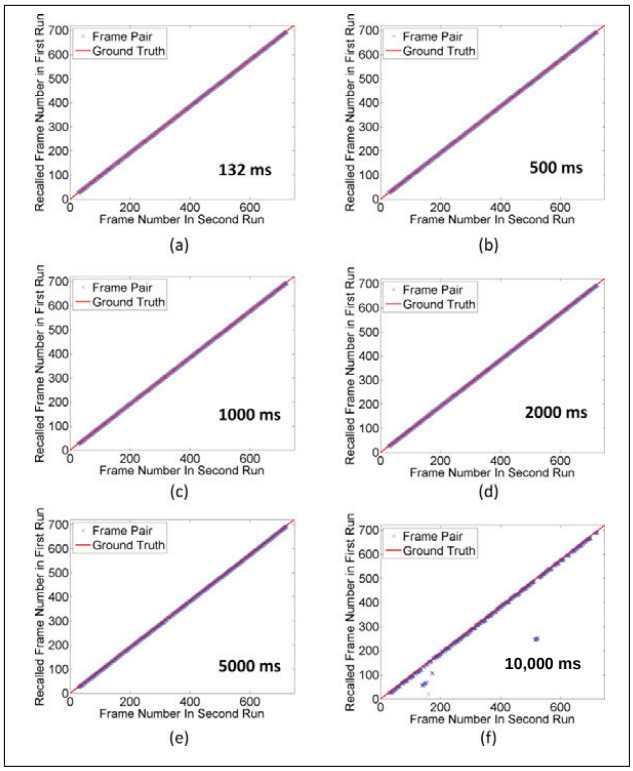

Fig. 26. Matches between the second (varied blur) run and the first fixed exposure run for motion blur corresponding to (a) 132 ms, (b) 500 ms, (c) 1000 ms, (d) 2000 ms, (e) 5000 ms and (f) 10,000 ms exposure durations.

Table 4. Motion blur recall rates and localization errors.

Exposure length	Recall	Mean localization error		Max localization error	
		Frames	Meters	Frames	Meters
132 ms	93.2%	0.44	5.8	1.38	18
500 ms	93.2%	0.376	5.0	1.35	18
1000 ms	93.2%	0.410	5.4	1.71	23
2000 ms	93.2%	0.797	10.5	2.22	29
5000 ms	93.2%	2.46	32.4	4.27	56
10,000 ms	87.3%	11.5	152	252	3320

led us to examine a zoomed in section of the frame matching graphs (Figure 28). The graph clearly shows there is a lag in the frame matching, which increases as the degree of motion blur increases. Upon consideration, this effect is to be expected, as longer and longer exposure times will create an image which represents a temporal average of images further and further backwards in time.

Although the focus of the paper is on investigating the limits of vision-based localization, we briefly present localization results for a day–night experiment to provide one potential application of the long-exposure results. Because the matching process is robust to long-exposure durations, compression artifacts and motion blur, bright images can be obtained even on unlit mountain roads by combining

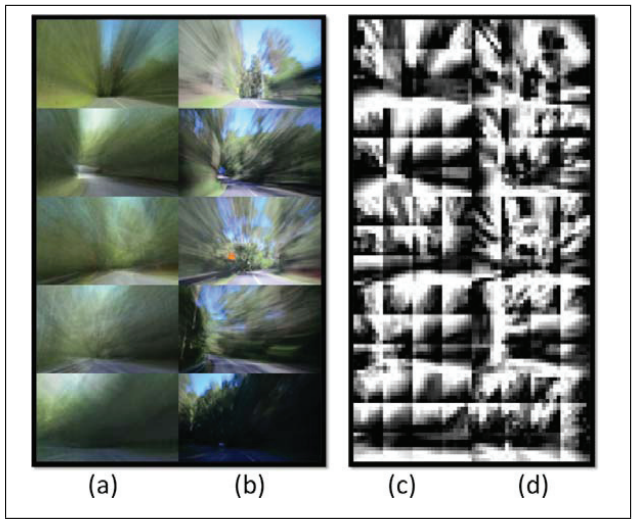

Fig. 27. Sample (original) frames from a matched sequence of the (b) 630 ms fixed exposure images matched back to the (a) images with motion blur corresponding to 5000 ms exposure durations. (c), (d) The patch normalized images actually used by the matching algorithm.

high sensor gains and long-exposure durations. Figure 29 shows the frames from a short-exposure day-time dataset being correctly matched back to fixed 630 ms exposure images obtained in the middle of the night on an unlit road. The corresponding images for one of the matched sequences are shown in Figure 30. The majority of locations along the route are successfully recognized despite the high degree of motion blur and perceptual change. Extension 3 shows examples of crisp day-time image sequences that were successfully matched back to blurry night-time image sequences.

5.8. Image resolution (perspective images)

In this section, we present results from running SeqSLAM on two perspective camera datasets using low-resolution (16- and 32-pixel) images. Given the datasets' source (YouTube), we did not have metric ground truth. Instead, we present the localization graphs at a performance level that was qualitatively assessed to provide 100% precision, by comparing the image sequences associated with the matched sub-routes, with a match being deemed as correct if it (a) visually matched the correct location as deemed by a human assessor and (b) was confirmed as being in the correct section of the dataset (to avoid assessing a match as correct in aliasing situations).

5.8.1. Rowrah motorbike dataset 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 m), as shown in Figure 31(a). The vertical axis shows the central frame number of the matching sub-route from the first traverse of the route. Figure 31(b) shows five frames obtained

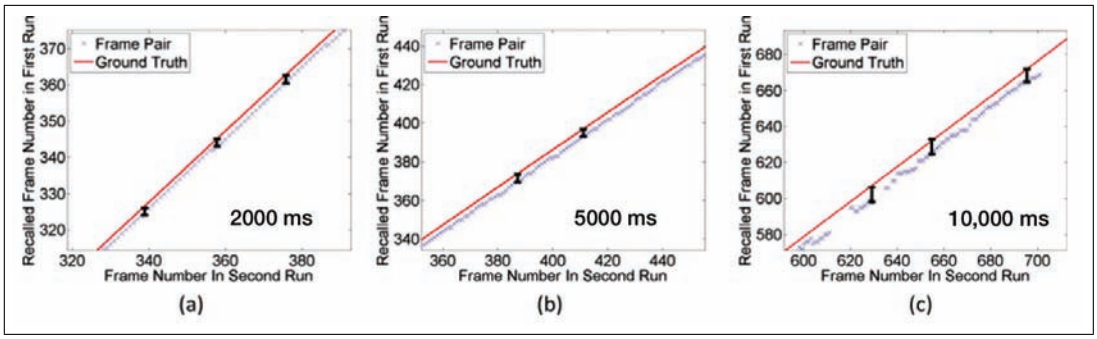

Fig. 28. Zoomed in section of the image match graphs shown in Figure 26 for the more severe motion blur experiments: (a) 2000 ms, (b) 5000 ms and (c) 10,000 ms. The black vertical bars show the offset between the ground truth matches (solid red line) and the reported matches, with the offset increasing as the degree of motion blur increases.

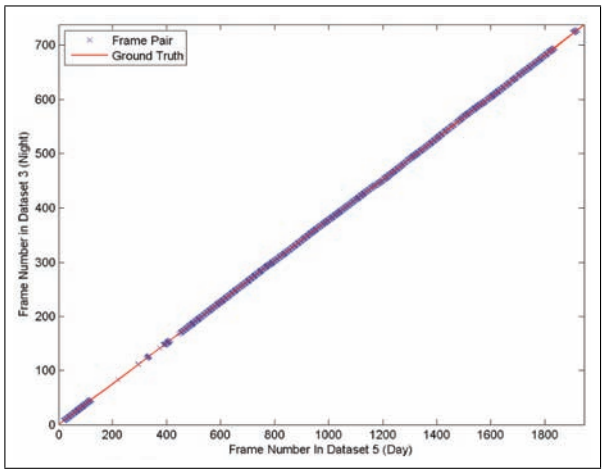

Fig. 29. Matches between a crisp short-exposure duration day-time dataset and 630 ms night-time exposures, overlaid on ground truth.

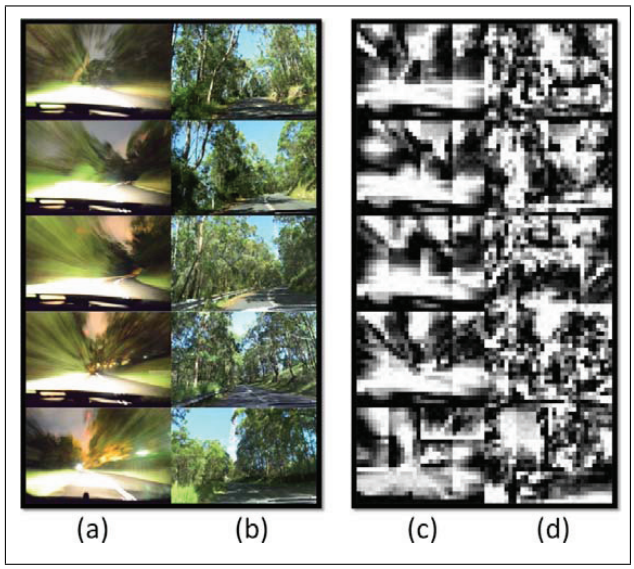

Fig. 30. Sample (original) frames from a matched sequence of the (b) short-exposure day-time images matched back to (a) 630 ms exposure images obtained during a night-time lap. (c), (d) The patch normalized images actually used by the matching algorithm.

by evenly sampling a sub-route from the second traverse of the circuit, with Figure 31(c) showing the corresponding frames from the matching sub-route during the first circuit traverse.

5.8.2. Pikes Peak dataset The recall level at near 100% precision was around 50% for the Pikes Peak dataset (Figure 32). 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.

5.9. Illustrative failure cases

In this section we show illustrative failure cases demonstrating the limitations of the current approach with respect to significant pose variance and susceptibility to sequence aliasing – routes in the environment that appear similar to other routes.

5.9.1. Sequence aliasing Figure 33 shows the frame difference matrix for the Guiabot dataset. Each column contains the difference scores for a frame to all previous frames in the dataset, with darker pixels representing smaller difference scores (and, hence, better matching frames). A section of corridor on this particular floor has a highly repetitive nature, meaning that there are not just aliased individual frames but aliased sequences of frames. The starting frames from three aliased sequences are shown in Figure 33(c) to (e), with the low-resolution versions shown in Figure 33(f) to (h). For the aliasing problem, SeqSLAM does not benefit from a 40-fold increase in resolution in each dimension (from 8×6 to 320×240); if anything performance is worse (Figure 34). This lack of improvement is in contrast to feature-based techniques, where increasing the resolution offers the potential for detecting small but disambiguating

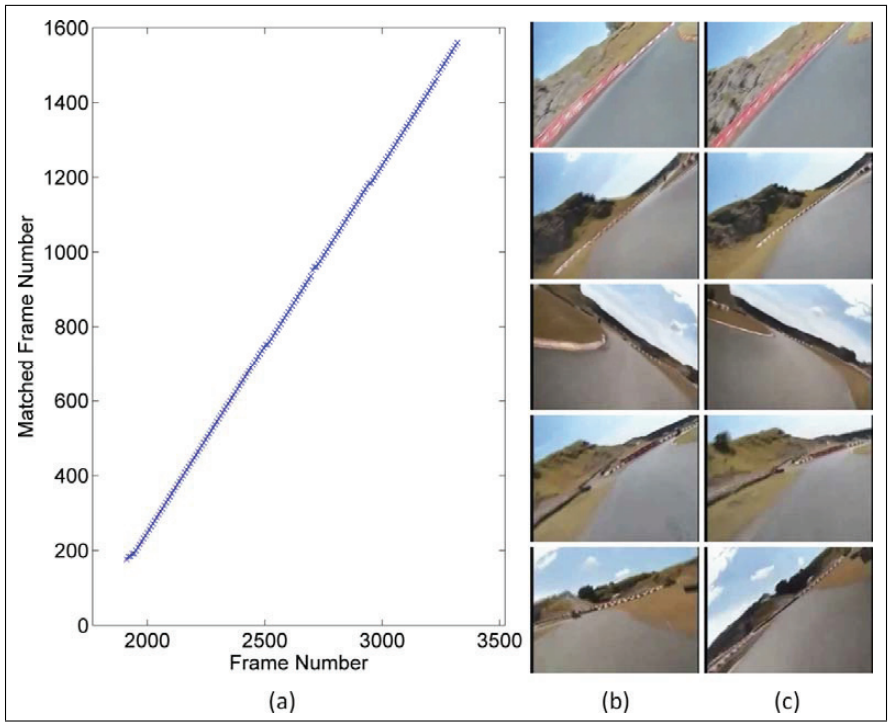

Fig. 31. (a) Sub-route localization for the Rowrah dataset. (b) Frames from second route traverse and (c) matching frames from the first traverse.

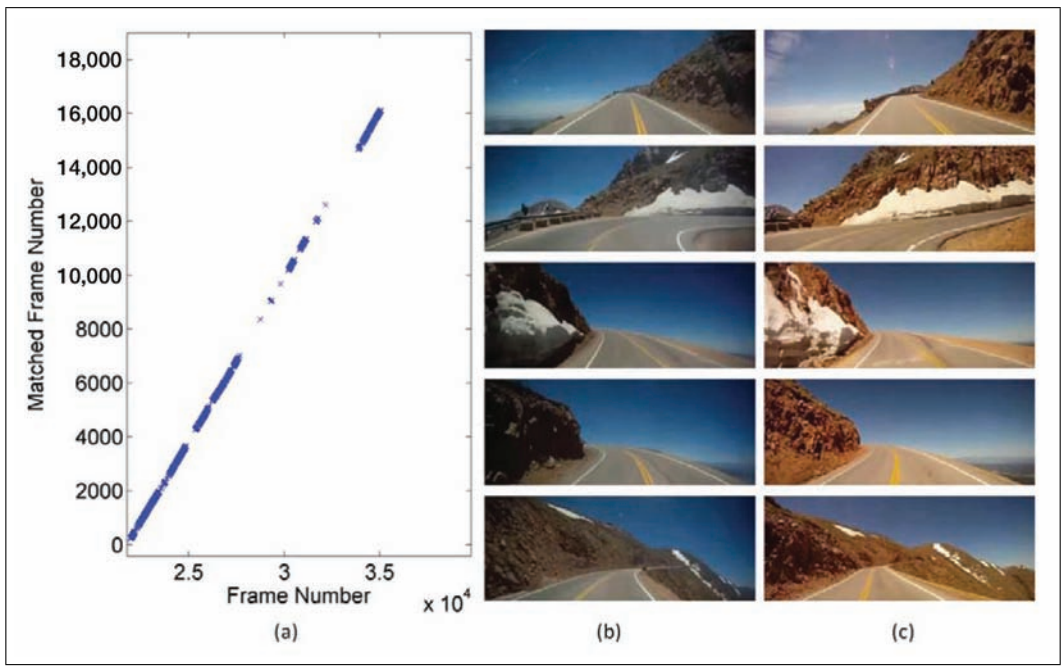

Fig. 32. (a) Sub-route recognition for the Pikes Peak dataset. Recall was not achieved in sections of the 20 km route, most likely due to large variations in the racing-line. (b) Frames from the second traverse and (c) matching frames from the first traverse of the route.

features. While using a longer sequence to perform localization would solve the aliasing problem in this specific situation, it is clear that there can be arbitrarily long aliased sequences that cause the method to fail.

5.9.2. Lateral pose variance Figure 35 shows the frame difference matrix generated by SeqSLAM when processing all three corridor datasets in order (left, center and then right). The strong matching diagonals between left and

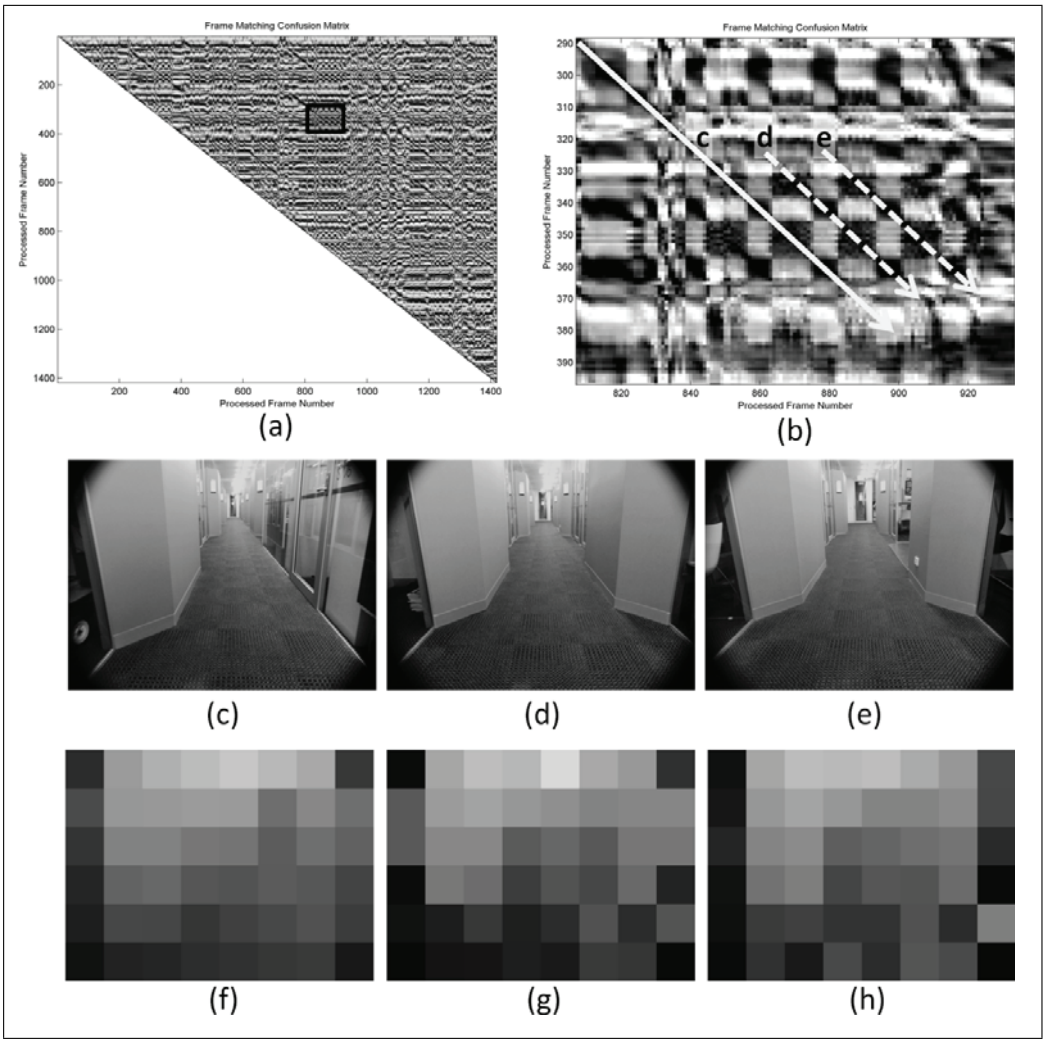

Fig. 33. (a) Frame difference matrix for the Guiabot dataset, with significant sequence aliasing as revealed by the (b) zoom inset. Darker pixels indicate lower frame differences (and hence better frame matches). The *x*-axis corresponds to the frame number in the dataset, with each column showing the difference scores between that frame and all previous frames in the dataset. (c)–(e) The frames at the start of each aliased sequence. The sequence starting at (d) incorrectly matches back to the sequence starting at (c), while the sequence at (e) incorrectly matches back to the sequences starting at (c) and (d).

center, and between center and right, indicate that the system has some degree of lateral pose invariance. However, matches between the left and right traverses of the corridor result in poor to non-existent matches. Extension 4 shows examples of frame sequences from all three corridor datasets.

5.10. Statistical analysis of individual image matching performance

Two questions that arose from the studies were, firstly, is it possible to simply perform matches off individual images, rather than sequences of images, and secondly, why is the algorithm so successful? Although the first question was partially addressed by the sequence length study, we decided to further investigate both questions further. We used the metric ground truth provided with the Eynsham dataset to analyze the performance of the individual image matching algorithm presented in Section 3.1. For every 8×4 pixel frame of the second 35 km traverse of the Eynsham environment, we ranked the image matching scores produced by the image similarity calculation. We then identified where within that ranked list the correct image match (as determined by the ground truth data) was located.

The first question was easily addressed – less than 1.6% of the top image match candidates were actually correct – which meant that using a sequence length of one, the maximum recall level achievable at 100% precision would only be 1.6%, a 35-fold reduction in performance compared to a sequence length of 50. It is interesting to compare this poor single frame performance with the vastly superior performance by all implementations of FAB-MAP, albeit operating on much higher-resolution images. We concluded that relying on *individual* very low-resolution images does not

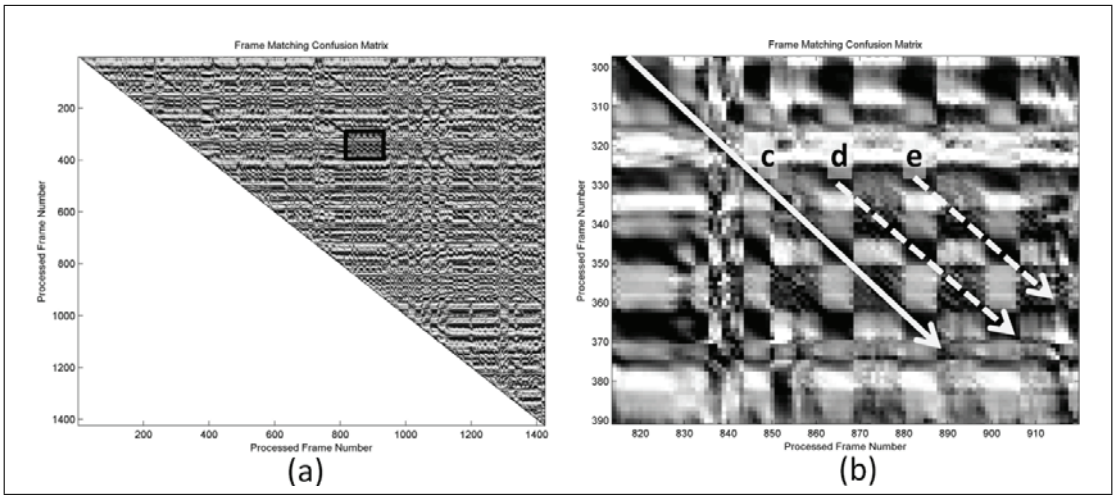

Fig. 34. Increasing the image resolution significantly to 320×240 does not decrease the degree of sequence aliasing, as shown by (b) the enlarged view.

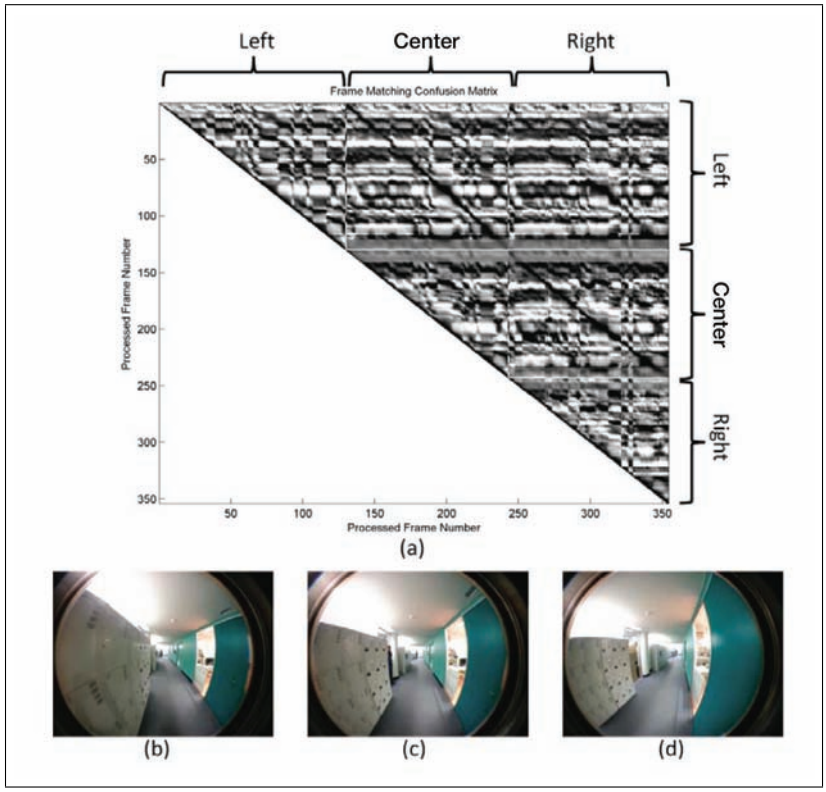

Fig. 35. (a) SeqSLAM matching matrix from three traverses of a corridor taken by (b) hugging the left wall (c) following the center of the corridor and (d) hugging the right wall. Strong matching diagonals are obtained between for the left-center run pair and the center-right run pair, but matching is weak to non-existent for the left-right pair.

yield good localization performance. However, this still left the question of why matching sequences of images works so well.

To answer this second question, we produced a correct image match ranking graph (Figure 36) and a cumulative match ranking graph, shown in Figure 37. Figure 37 reveals that, while the correct image match is almost never ranked as the number one match candidate, it is always ranked very highly. 92.9% of the actual (as determined by ground truth) matching images are ranked in the top 10% of image match candidates, 98.5% are ranked in the top 20% of image matches and a staggering 99.96% of actual image matches

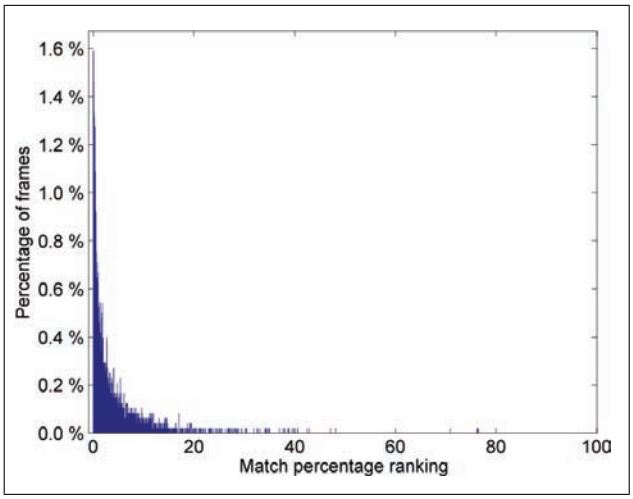

Fig. 36. At each time step, the individual image matching process ranks how closely all previous images match to the current image. This figure shows a histogram of the ranking of the actual correct image match (as determined by ground truth) within the entire set of images. It becomes clear that less than 1.6% of image frames are correctly matched, if picking the best match candidate.

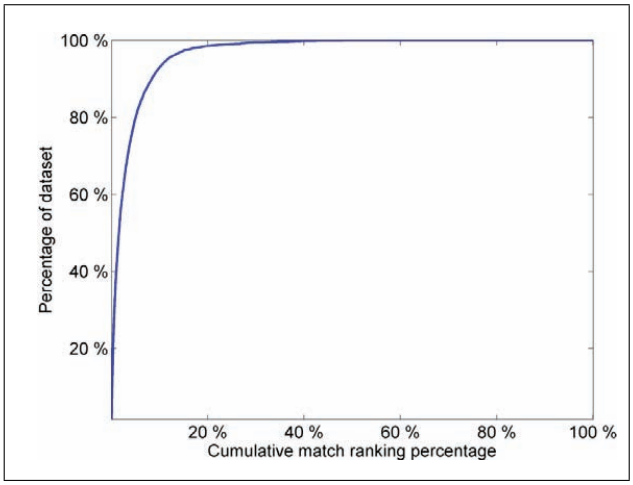

Fig. 37. Although the best matching image candidate output by the individual image matching process is usually incorrect, it is almost always ranked highly as shown by the cumulative ranking percentage plot shown here. The figure shows that more than 98.5% of the correct image matches are ranked within the top 20% of image match candidates.

are ranked in the top 50% of image matches. Clearly any algorithm that attempts to match a sequence of images will benefit from this property, by combining lots of good, although not usually the best, image matches.

5.11. 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.

5.11.1. 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 World Factbook* (CIA, 2012), the 191 countries surveyed have a total of approximately 70,000,000 km of paved and unpaved roads. Storing images every 5 m of road would result in 224 GB 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 MB download would provide 3000 km of road data, enough for a regional area.

5.11.2. Compression Further gains in storage are difficult– it is challenging to achieve any more significant storage reduction for 8×4 grayscale images using state of the art image and video compression algorithms such as .jpg image compression or the H.264 video compression standard. As shown in the video compression study, it is however possible to significantly compress *higher*-resolution images and video without adversely affecting localization performance. A 120×40 image can be compressed down to 210 bytes per frame without compromising performance. Whether uncompressible low-resolution images or highly compressed larger video resolutions are optimal will most likely depend on the application.

5.11.3. Computation Here we discuss the two key computation scenarios – use as a localization only system and also as a real-time SLAM system (with a spatial, rather than frame-based map). 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. We first discuss the localization only scenario where a library of images already exists, starting with the situation where self-motion information (such as automobile 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 5 presents a number of computation scenarios, assuming 5 m frame spacing and a camera speed of 15 m/s (54 km/h). 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

Table 5. 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
10,000 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

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 central processing unit (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 expansion increases the computational load 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.

From an application standpoint, we also note that current GPS-based automobile navigation systems primarily rely on localization information and a prior map to guide a user, and most do not have any "dead-reckoning" ability to path integrate without a direct GPS signal. Deploying SeqSLAM on a car would have the same characteristics – a prior map of low-resolution imagery from street view datasets, and localization output. In addition, the effective mean localization accuracy of a consumer automobile navigation system (5–10 m) is no better than SeqSLAM's mean day–night localization accuracy ($<$10 m for image exposures $<$2000 ms).

To develop the underlying algorithms into a traditional SLAM system with a bounded, spatial map and dead-reckoning ability, self-motion information such as obtained through visual odometry or wheel encoders could be combined with some form of graphical map. For example, the system structure would integrate well with the existing RatSLAM system, by replacing the filtering provided by the pose cell neural network with the sequence matching provided by SeqSLAM, which in effect performs the same role. However, the primary required modification would be to modify the system to only learn new sensory snapshots when not already recognizing a familiar place, so that the growth in the number of stored images is theoretically capped by the size of the route.

6. Discussion

The primary contribution of this paper is to show that in path-like environments, vision-based place recognition can be achieved using sequences of remarkably small and/or poor-quality images. The localization algorithm used, SeqSLAM, sacrifices single frame localization capability in order to achieve more robust recognition using sequences of images, and the results demonstrate that such an approach can be successfully deployed in a range of environments and situations. Some components of the approach such as the search method and image comparison technique are quite simple, and as such we are establishing an "upper bound" for the minimum visual information content required to localize along a route. It may well be possible that, with the introduction of a more sophisticated sequence searching algorithm, localization can be achieved with even smaller images. Improved algorithms will result in the studies in this paper being more widely applicable to scenarios where vehicle motion is less constrained or takes place in higher dimensions.

Performing localization using a sequence of images rather than a single image removes the requirement that the image matching scheme needs to be able to reliably calculate a single global image match. Instead, the image matching front end must only on average report matches better than could be done by chance. How much better depends on how long of 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 automobile travel, where translational speed information is available from on-board diagnostic (OBD) systems. We have established that, at least in the realm of low-resolution images, this "better than average" matching requirement is more than adequately met, as shown in the study investigating the statistical distribution of the individual image matching results.

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 images are distinctive. 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 techniques from one-dimensional routes to two-dimensional areas. The third is to maintain localization using odometry in situations where a system briefly loses localization while traversing several fragmented sequences (such as passing through a complex intersection). We note that obtaining sufficiently accurate visual odometry from the degraded images used in these experiments is very challenging, although there is some precedent (Milford and Wyeth, 2008). An applied system would most likely have to use higher-resolution imagery to perform visual odometry, or in situations where this is not available, another source of odometry information such as an OBD key.

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 currently 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. It may be possible in future to fulfill this "lots of local best matches" requirement by implementing a visual processing system that can autonomously choose from a range of scene comparison algorithms, depending on the nature of the images being examined.

In conclusion, we hope the experiments performed in this paper provide a somewhat provocative reference for future place recognition research. As mentioned earlier in the paper, the current state of the art approaches that primarily use image features are here to stay, and for good reasons including a lack of latency and easy integration object recognition and three-dimensional reconstruction systems. We are not advocating a reversion to the low-resolution systems of decades ago. Rather, we suggest that if competitive place recognition performance is possible using sequences of tiny images, tiny fields of view or highly blurred images, then maybe there are still significant advances to be made that combine the best of both worlds.

Acknowledgements

We thank James Eather for his assistance with the NXT and motion blur experiments, and Peter Corke, Gordon Wyeth, Ben Upcroft, Liz Murphy, Tim Morris, Feras Dayoub and Stephanie Lowry for their comments on various versions of the manuscript and/or providing the Guiabot dataset.

Funding

This work was supported by an Australian Research Council Fellowship DE120100995 to MM.

References

Bay H, Tuytelaars T and Van Gool L (2006) SURF: speeded up robust features. In: *Computer vision – ECCV 2006*, pp. 404–417.

CIA (2012) *The World Factbook*. https://www.cia.gov/library/publications/the-world-factbook/

Cummins M and Newman P (2009) Highly scalable appearance-only SLAM - FAB-MAP 2.0. In: *Robotics: Science and Systems, Seattle*, WA.

Davison AJ, Reid ID, Molton ND et al. (2007) MonoSLAM: real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29: 1052–1067.

Franz MO, Scholkopf PG, Mallot HA et al. (1998) Learning view graphs for robot navigation. *Autonomous Robots* 5: 111–125.

Huynh DQ, Saini A and Liu W (2009) Evaluation of three local descriptors on low resolution images for robot navigation. In: *IEEE Image and vision computing* Wellington, New Zealand, pp. 113–118.

Kalogerakis E, Vesselova O, Hays J et al. (2009) Image sequence geolocation with human travel priors. In: *IEEE International conference on computer vision*, 253–260, Kyoto, Japan.

Klein G and Murray D (2008) Improving the agility of keyframe-based SLAM. In: *European conference on computer vision*, pp. 802–815.

Konolige K and Agrawal M (2008) FrameSLAM: from bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics* 24: 1066–1077.

Lowe DG (1999) Object recognition from local scale-invariant features. In: *IEEE International conference on computer vision*, Kerkyra, Greece.

Maddern W, Milford M and Wyeth G (2012) CAT-SLAM: probabilistic localisation and mapping using a continuous appearance-based trajectory. *The International Journal of Robotics Research* 31: 429–451.

Mahaffey J (2003) TTFF Comparisons.Milford M (2012) Visual route recognition with a handful of bits. In: *Robotics: science and systems VIII*, Sydney, Australia. Cambridge, MA: MIT Press.

Milford M and Wyeth G (2008) Mapping a suburb with a single camera using a biologically inspired SLAM system. *IEEE Transactions on Robotics* 24: 1038–1053.

Milford M and Wyeth G (2010) Persistent navigation and mapping using a biologically inspired SLAM system. *The International Journal of Robotics Research* 29: 1131–1153.

Milford M and Wyeth G (2012) 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.

Milford M, Schill F, Corke P et al. (2011) Aerial SLAM with a single camera using visual expectation. In: *IEEE International conference on robotics and automation*, Shanghai, China.

Milford MJ (2008) *Robot Navigation from Nature: Simultaneous Localisation, Mapping, and Path Planning Based on Hippocampal Models*. Berlin-Heidelberg: Springer-Verlag.

Mirmehdi M, Clark P and Lam J (2003) A non-contact method of capturing low-resolution text for OCR. *Pattern Analysis and Applications* 6: 12–21.

Murali VN and Birchfield ST (2008) Autonomous navigation and mapping using monocular low-resolution grayscale vision. In: *IEEE Conference on computer vision and pattern recognition*, Anchorage, AK, pp. 1–8.

Murphy L, Morris T, Fabrizi U et al. (2012) Experimental comparison of odometry approaches. In: *International symposium of experimental robotics*, Quebec City, Canada.

Newman P, Cole D and Ho K (2006) Outdoor SLAM using visual appearance and laser ranging. In: *IEEE International conference on robotics and automation*, Florida, United States.

Newman P, Sibley G, Smith M et al. (2009) Navigating, recognizing and describing urban spaces with vision and lasers. *The International Journal of Robotics Research* 28: 1406–1433.

Phillips PJ, Flynn PJ, Scruggs T et al. (2005) Overview of the face recognition grand challenge. In: *IEEE International conference on control, automation, robotics and vision*, Singapore, vol. 941, pp. 947–954.

Pomerleau DA (1992) Neural network perception for mobile robot guidance. DTIC document.

Sakoe H and Chiba S (1978) Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing* 26: 43–49.

Sim R, Elinas P, Griffin M et al. (2005) Vision-based SLAM using the Rao-Blackwellised particle filter. In: *International joint conference on artificial intelligence*, Edinburgh, Scotland.

Torralba A, Fergus R and Weiss Y (2008) Small codes and large image databases for recognition. In: *IEEE Computer vision and pattern recognition*, pp. 1–8, Anchorage, AK.

Appendix: Index to Multimedia Extensions

The multimedia extensions to this article are at http://www.ijrr.org.

Table of Multimedia Extensions

Extension	Type	Description
1	Data	Provides datasets and dataset access information.
2	Video	Shows camera frame sequences aligned by the NXT-based sequence matching experiment.
3	Video	Shows video of sequences of long-exposure night-time images matched to short-exposure day-time images.
4	Video	Shows dataset examples of functional and non-functional degrees of lateral camera pose variance.