Experience-based navigation for long-term localisation

Winston Churchill and Paul Newman

The International Journal of Robotics Research 2013 32: 1645 originally published online 16 September 2013
DOI: 10.1177/0278364913499193

Article

The International Journal of
Robotics Research
32(14) 1645–1661

DOI: 10.1177/0278364913499193
ijr.sagepub.com

Experience-based navigation for long-term localisation

Winston Churchill and Paul Newman

Abstract

This paper is about long-term navigation in environments whose appearance changes over time, suddenly or gradually. We describe, implement and validate an approach which allows us to incrementally learn a model whose complexity varies naturally in accordance with variation of scene appearance. It allows us to leverage the state of the art in pose estimation to build over many runs, a world model of sufficient richness to allow simple localisation despite a large variation in conditions. As our robot repeatedly traverses its workspace, it accumulates distinct visual experiences that in concert, implicitly represent the scene variation: each experience captures a visual mode. When operating in a previously visited area, we continually try to localise in these previous experiences while simultaneously running an independent vision-based pose estimation system. Failure to localise in a sufficient number of prior experiences indicates an insufficient model of the workspace and instigates the laying down of the live image sequence as a new distinct experience. In this way, over time we can capture the typical time-varying appearance of an environment and the number of experiences required tends to a constant. Although we focus on vision as a primary sensor throughout, the ideas we present here are equally applicable to other sensor modalities. We demonstrate our approach working on a road vehicle operating over a 3-month period at different times of day, in different weather and lighting conditions. We present extensive results analysing different aspects of the system and approach, in total processing over 136,000 frames captured from 37 km of driving.

Keywords

Localisation, mobile and distributed robotics, SLAM, mapping, field robots, field and service robotics

1. Introduction

To achieve long-term autonomy, robotic systems require the ability to navigate accurately and reliably for extended periods of time. When operating over such time scales they will experience changes in the environment, and must be able to deal with this seamlessly: we see this as a big challenge. Current state-of-the-art solutions can perform tasks such as autonomous retro-traverse of a robot against a previously taught route (Furgale and Barfoot, 2010), however these are only feasible if the prior map captures the current state of the world. In reality we find many commonly occurring events that humans deal with naturally can disrupt these autonomous systems, as the appearance of the world changes sufficiently for current algorithms to fail. Such change can come from many sources: dynamic object/scenes, lighting conditions, time of day, weather and seasonal shifts. An autonomous robot must be equipped with the ability to deal with these regular events. Self-driving cars that can only be used on overcast autumn afternoons, or autonomous agricultural vehicles that can only operate before noon on sunny days are underwhelming. If we expect true autonomy from our robots this problem must be addressed.

Considering the case of performing navigation on a car equipped with a camera, the obvious approach would be to implemented a simultaneous localisation and mapping (SLAM) system that can map and localise all at once. But what should we do if we revisit a place and its appearance has changed drastically—perhaps it has snowed? What do we do if a place's appearance slowly creeps as summer turns to autumn? How should we manage the images seen in Figure 1, which exhibit a mixture of slight to extreme variation in appearance? Should we undertake some unifying data fusion activity to yield a monolithic map in which we can localise? We argue that we should not; in the limit such a map would hold all observed features, coerced into a single

Oxford University Mobile Robotics Group, Oxford, UK

Corresponding author:
Winston S. Churchill, Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, UK.
Email: winston.churchill@eng.ox.ac.uk

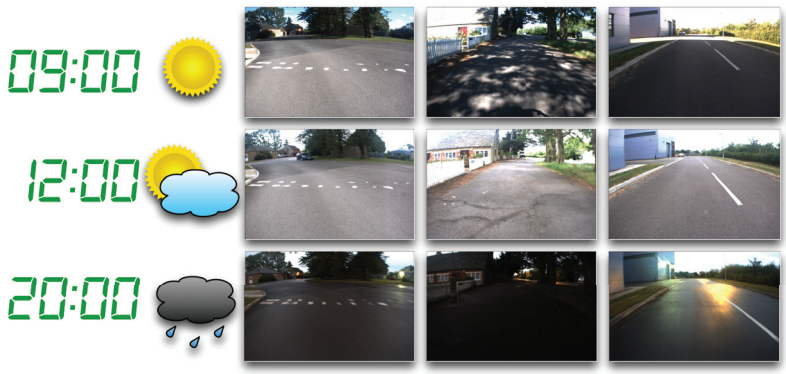


Fig. 1. The same place can look very different, depending on when it is observed. This variation may come from structural change, lighting conditions or through shifting seasons. Attempting to produce a conventional map which is in a single, consistent frame of reference may be difficult due to lack of correspondences.

reference frame through likely questionable data associations. The things we observe on a given tree in winter are simply not the same things we observe again in the summer; the road's texture is different at high noon compared with a dry dawn. Therefore, we shall not force things to be coherent. If, for example, part of a workspace on Tuesday looks wildly different on Wednesday then we shall treat these as two independent experiences which equally capture the essence of the workspace. We shall only ever tie them together topologically.

This paper will lay out exactly how we accumulate, manage and exploit visual experiences to maintain seamless navigation. But to begin with, a high level view of our approach is appropriate. When visiting a new place for the first time, we save the visual odometry (VO) output (similar to most systems), and, for reasons that will become clear, refer to it as an "experience" rather than a map. When the robot returns to the same area, it performs localisation against the experience using the live image stream (while also performing VO on the live stream). Should localisation fail, a new experience is created based on the live VO output. However, if at any point in the future the robot manages to re-localise in the original experience, the saving of the current experience is stopped and the system returns to localising. Figure 2 gives an overview of our approach. Using this method, places with many faces have more experiences associated with them than regions that are more staid, we naturally capture the variation in appearance of the world. We refer to the collection of all experiences as the plastic map.

A core competency on which we depend is a VO system which continuously produces a (possibly ephemeral) 3D model of the world using a stereo pair. This system is always on, always consuming the live stereo pair stream and estimating the relative transformations between camera poses and producing 3D feature locations relative to camera poses. Concretely an experience is a stored set of relative poses and feature locations. Note that we use a *relative* framework, as in Sibley et al. (2010), which allows us

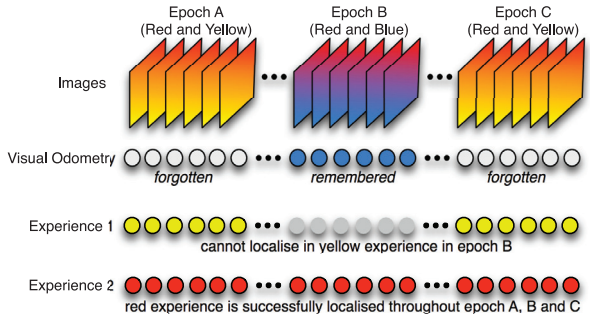


Fig. 2. An overview of our approach (best viewed online in colour). A VO system continuously consumes the live image stream. In parallel a series of localisers attempt to localise each live frame in their own experience. In epochs A and C both localisers successfully localise the frames in their experiences, so the VO output is forgotten. In epoch B, localisation is only successful in one saved experience (experience 2), which is deemed too few, so the VO output is saved in a new experience.

to entirely avoid operating in a single global frame. All we require is an ability to render a metrically correct idea of camera motion and 3D feature locations in the vicinity of the robot's current pose: we do not care about the location of things that are far away and which we cannot see. Upon revisiting an area, localisation is attempted in all previous experiences that are relevant to the area.

By keeping experiences independent we are able to run a "localiser" for each. This can trivially be done in parallel and allows the system to utilise relevant experiences. In reality, at runtime we see that the number of active and successfully localised experiences is small. After all, each new experience is only created out of necessity because it is visually different from all others. Therefore subsequent visits to an area should be able to localise in only a small number of experiences as they are by construction visually different. Finally we would like to stress that although we describe the framework using vision, it is actually agnostic to the sensing modality and could be used with other

sensors such as laser range finders so long as equivalent systems to those described above are supplied.

We have tested our system on 53 runs of two laps of a 0.7 km circuit, covering 37 km in total and consisting of over 136,000 stereo frames. The data were collected over a 3-month period at many different times of day and in different weather conditions. The remainder of the paper is organised as follows. Section 2 looks at related work. Sections 3 and 4 presents the main idea in this paper and Section 5 briefly outlines the implementation details. Sections 6 presents results with discussions and conclusions given in Section 7.

2. Background

Localisation and mapping algorithms have matured in recent years, with continually greater distances being successfully navigated. Ego-motion estimation systems have been demonstrated performing up to tens of kilometres (Konolige et al., 2007; Newman et al., 2009; Milford and Wyeth, 2009), while topological approaches have been shown successfully operating to 1,000 km (Cummins and Newman, 2009). However, the problem of achieving similar results over large *time* scales has received relatively little attention, and is becoming a major obstacle to the realisation of long-term autonomy. We now review some of the approaches to tackling long-term navigation which has been studied using both vision and lidar sensors.

Central to our work is the idea that the same physical space can appear differently to sensors and hence should be modelled by multiple impressions. Three papers in particular have also adopted this point of view.

The first is Milford and Wyeth's biologically inspired RatSLAM system (Milford and Wyeth, 2009) which maintains a non-metric topological graph referred to as an "experience map". We must stress here that the overloading of the term "experience" here is an unfortunate clash of nouns between this work and theirs. In their work a node in the graph is referred to as an experience and contains a view of the world at that point, a pose estimate and transition information to other experiences (from wheel odometry). This map is relaxed over time to minimise the difference between global node positions and the relative transitions. A new experience is created when the scene is not sufficiently similar in appearance to any of the previous ones. This allows experience creation when the robot visits a new place in the world, and when revisiting areas that have changed. To prevent their map becoming overly saturated with experiences they prune it by placing a regular grid over the map and only allowing one node to exist per cell. The single surviving node is picked at random, they find this produces similar results to other deletion metrics such as connectivity or recency. The system is shown performing successfully for a two week period in an office environment.

The second is by Konolige and Bowman (Konolige and Bowman, 2009) who developed their view-based maps system (Konolige et al., 2010) to deal with a changing world. The original system creates a skeleton graph of keyframes from a VO system, while a place recognition component provides loop closure information. The map is in a single frame of reference and is incrementally optimised via Toro (Grisetti et al., 2007). Over time the environment changes, resulting in the current view failing to match against previous keyframes. This causes new keyframes to be added to the system. To keep the density of keyframes down in their graph, for a local metric neighbourhood a maximum number of allowed keyframes is defined. Their deletion scheme favours maintaining view diversity (based on a keyframe similarity metric) and then recency. Their system is shown operating in a lab environment that includes moving people and furniture, and changing lighting conditions over a few days.

The third is by Burgard et al. (2007) who map the environment using a laser scanner, which they split into sub-maps up to 20 m^2. Each time a sub-map is revisited, the associated sequence of observations is recorded. The set of observation sequences are then clustered using a fuzzy k-means algorithm, and the Bayesian information criterion is used to select the model that has the best compromise between fitting the data and number of clusters. This is repeated for every sub-map. When performing localisation in a sub-map, the choice of cluster becomes another variable in their particle filter solution. Results from a robot operating within an office space are presented, where typical changes are door configurations (open or closed). By including the model of the dynamic environment, the localisation error is improved compared to the standard static map approach.

Biber and Duckett (2005) also describe a laser-based system for dealing with structural change in the environment. They highlight the issue that using simple recency-weighted averaging leads to measurement predictions that have never been seen, as structural change tends to occur in discrete step sizes. They approach the problem by maintaining local maps for a number of points in the environment. Each local map stores a set of measurement sequences. The size of the sequences is fixed, and the resample rate (how quickly new data replaces old data) is varied for each set. This creates a local map that has both short-term and long-term structure. Localisation is then performed against a map synthesised from the best fitting time scale sequences of nearby local maps. In their experiments the shortest half-life is a few seconds, resulting in a map which is effectively the current sensor measurements. The longest half-life is nearly 2 weeks. They show improved localisation accuracy over the period of several weeks in an environment that included a busy lab, corridor and public hallway when compared to the static map captured on the first day.

Dayoub and Duckett (2008) present an image based variant of Biber and Duckett's previous time-scale approach

for topological mapping with an omnidirectional camera. The environment is mapped by a set of known points called reference views. Each view has two explicit memory time scales, short term and long term. Features seen enough times in the short-term memory advance to the long-term memory, while those not repeatedly seen are forgotten. Similarly in the long-term memory, features not seen for a while are forgotten. They show the current view has a higher similarity to these adaptive reference views compared to static reference views over a 9-week period. In this work the reference views are fixed and known, as is the position of the robot relative to the views, however these limitations were removed in later work (Dayoub et al., 2011).

Recently Taneja et al. (2011) have detected structural changes in their dense reconstruction of urban environments, allowing them to only update their outdated parts of the model. By incorporating semantic information they are able to ignore short-term changes caused by things such as cars and people. In this work they assume they are able to ignore weather and lighting conditions, given that they are consistent across visits.

We also note the work of Furgale and Barfoot (2010) who develop a teach-and-repeat system using stereo vision in an outdoor setting. While not necessarily long-term mapping, they do encounter and highlight some relevant issues. During a teach phase, the robot is manually driven along a desired route, creating a series of connected sub-maps using VO. For the repeat stage, a localisation module is used to retrace the original path. The system is shown performing successful repetitions of taught routes, totalling more than 32 km length, in both urban settings and the Canadian High Arctic. They highlight feature repeatability issues caused by changing lighting conditions (using Speeded Up Robust Features (SURF) (Bay et al., 2008)). After 10 hours a repeat pass was unable to localise against the original taught pass due to changes in lighting conditions. Further, they found routes taught on overcast days are problematic if used for localisation on sunny days (and vice versa). They do not attempt to capture changing appearance from the original traverse.

In a similar vein of demonstrating large-scale localisation against a prior map (but without the repeat step driving the vehicle), Lategahn and Stiller (2012) recently demonstrated a road vehicle localising in a large prior map containing sparse feature points created from a previous drive. They show successful localisation over two $\sim$1 km trajectories but do not attempt to model changing conditions.

3. Experience-based navigation

3.1. Overview

We begin with a high-level overview of our framework. The main building block used in this work is a VO system. This gives us a sequence of *relative* poses and the associated constellation of visual features. While operating, regardless of whether localisation is currently successful, VO is always performed on the live image stream. On the initial visit to a new area we save this output of the VO like most systems. We refer to the whole sequence of saved poses and related features as an "experience". When revisiting the area the robot attempts to use the live stream of images to localise in the saved experience. If at any point this is unsuccessful, a new experience is created based on the current appearance of the world. As the robot continues, still saving to this new experience, it is also trying to re-localise in its previous experience(s). If this is successful at any point, saving is stopped and the system returns to localising in its previous experience. An overview of the approach is shown graphically in Figure 2.

Importantly this methodology causes the system to "remember" more representations for regions that change often, and fewer for regions that are more static. This results in the system naturally capturing the varying complexity of different parts of the world. We call the collection of all experiences the "plastic map". Note that we handle new routes and complete localisation failures seamlessly: indeed it is the failure of localisation which drives the saving of a new experience. This is because we make the assumption that our localisation fails because of bad or unsolvable data association: what was there before is simply not there now. Making experience creation a function of successful localisation as opposed to some arbitrary image similarity metric, allows it to be directly influenced by what the system is trying to achieve: successful navigation. Choosing an image similarity metric could result in experiences being saved when the current ones are sufficient for localisation, or vice versa.

Another aspect of our experiences is that we keep them independent in terms of features and frames of reference. We only look to link them topologically. Once an experience is created its poses and features are not updated, merged or deleted based on subsequent observations of that place, the experience is fixed, and is either successfully localised against or not. Taking this decision means we can avoid the difficult data association issues that would arise. Further, we do not force experiences that cover the same physical place in the world to live in the same reference frame, relative or global, still we only link them topologically. Again this means we do not have to coerce two different trajectories to be consistent via questionable data associations. The result of these design choices means experiences are independent and so localisation across multiple experiences can be trivially parallelised. However, the lack of a single frame of reference introduces a new challenge of how we find our position in multiple experiences. This issue is addressed in Section 4 but we first formalise the notion of an experience, a localiser and how these are used for navigation.

The plastic map created in this work has similar properties to the hybrid metric/topological map representations developed by previous researchers of large-scale navigation, notably the work of Furgale and Barfoot (2010) and Sibley et al. (2010), and can be used in a similar way for

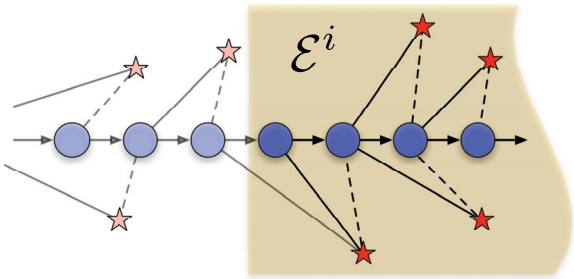


Fig. 3. An experience is simply the saved output from the VO system. Here we show the output of the VO: blue nodes represent stereo frames and red stars are relative landmarks. The shaded area is stored as a new experience $\mathcal{E}^i$ while the faded section is forgotten.

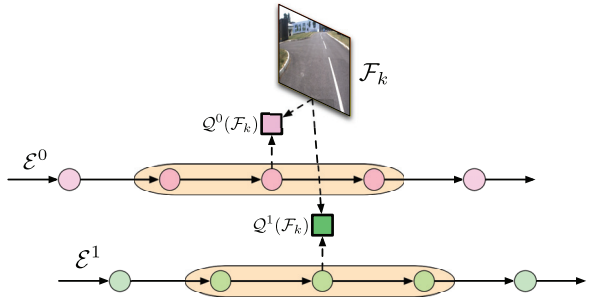


Fig. 4. Each experience, $\mathcal{E}^i$ has an associated localiser $\mathcal{Q}^i$. The localiser takes as input the current frame and attempts to match it to its assigned experience. The landmarks for matching are taken from a local region (shaded ovals) around the previous position in the experience.

driving a robot. The plastic map can be used to repeat previously visited routes, similar to the work of Furgale and Barfoot. However, because we often have multiple passes through an environment we have a wider region of operation than witnessed in a single-pass teach-and-repeat system. This approach lends itself well to autonomous systems that operate in environments that exhibit visual change, e.g. self-driving cars. As we adopt the relative approach, we do lose the ability to plan short cuts from the plastic map.

3.2. Experiences

A single experience is simply the saved output of the VO system. The VO system operates as follows. Given a sequence of stereo frames $\mathcal{F}^{\mathbf{k}} = \{\mathcal{F}_0, \ldots, \mathcal{F}_k\}$, at time k a stereo frame is processed and a camera node n_k is linked to node n_{k-1} by a 6 degree of freedom transform $\mathbf{t} = [x, y, z, \theta_r, \theta_p, \theta_q]^{\mathrm{T}}$. If the frame, $\mathcal{F}_k$, initialises new landmarks, these are stored relative to n_k. We denote the gth such landmark attached to n_k, where g is a global index,[1] as $\mathbf{p}^g_{s=k} = [x, y, z]^{\mathrm{T}}$: a vector stored relative to the instantiating stereo frame, $s = k$. Finally, n_k also contains a list of all landmarks observed in $\mathcal{F}_k$, many of which will be attached to other nodes: those in which they were initialised.

The VO system runs continuously on the live frame stream. When this needs to be saved (see Section 3.4) a new experience $\mathcal{E}^i$ is created and the output from the VO system is stored in this experience. Then $\mathcal{E}^i$ is simply a chain of camera nodes, the inter-node transforms and associated 3D features, examples are given in Figure 3. We refer to nodes in experiences as $\mathcal{E}^i_m$. Later we will explain how these chains are related (topologically) to form in concert a plastic map.

3.3. Localisers

We now introduce the idea of a localiser. Each saved experience is assigned a localiser. Given a live frame $\mathcal{F}_k$, its task is to calculate the transformation from the frame to a node in the experience. It operates in a very similar way to the live VO system except the proposed landmark set comes from the saved experience, not the previous live frames. The landmarks are taken from the local region surrounding the previous position in the experience. In addition, the localiser does not attempt to add or update landmarks in either the current VO output or the saved experience. It is completely passive in terms of its impact on both. Two localisers are shown in Figure 4.

An important competency of the localiser is its ability to tell whether it is "lost". This happens when the incoming frame can no longer be localised in the previous experience. There are many ways this can be calculated. Possibilities include the number of landmarks found and/or classified as inliers, and comparisons with the current VO output. We use the latter, which is described in Section 6. The output of each localiser at each time step is a binary result indicating if it is still successfully localised:

$$\mathcal{Q}^i(\mathcal{F}_k) = \begin{cases} 1 & \text{if localised} \\ 0 & \text{if lost} \end{cases} \tag{1}$$

If successful, the localiser can be queried for the node in the experience that $\mathcal{F}_k$ was nearest to

$$\mathcal{E}^i_m \leftarrow \mathcal{Q}^i() \tag{2}$$

Once a localiser is declared lost, it stays in this state until it receives outside assistance, discussed in Section 4.

3.4. Navigating by experience

We now explain how these experiences are leveraged to achieve long-term navigation in changing environments. While running we always perform VO on the live image stream. The question of whether this is saved as a new experience is a function of the result of the localisers running on the current experience set. We define N to be the minimum number of successful localisers running at any point in time. If N or more localisers are successful, we believe our current representation of the local region to be sufficient and discard the VO output. However, when the number of

successful localisers falls below N, we create a new experience from the VO output. This continues until the number of successful localisers returns to N or above, when saving is stopped.

Experience creation is driven by the success or failure of localisation to prior experiences. This results in us naturally capturing the varying complexity of the world. In areas of high visual variation we store more experiences, while in regions that remain visually similar over time, we save relatively few experiences as our prior ones are sufficient for localisation. By allowing N to be greater than 1, the system is more robust to single localisation failures and it gives us more confidence in our position. This creates multiple experiences that encompass the same physical place and visual appearance, leading to common features across experiences. We choose not to merge, average or discard shared features as we want to explicitly avoid the difficult data association problem across experiences.

4. Connecting experiences

One of the disadvantages of storing each experience in its own independent relative frame is that position look-up across experiences is more difficult. Consider the simplest case of two experiences, $\mathcal{E}^1$ and $\mathcal{E}^2$, starting at the same position in the world and following the same trajectory. For a given node $\mathcal{E}^1_m$, it is not possible to simply integrate the local transforms in $\mathcal{E}^2$ until the equivalent position is reached, as they are globally inaccurate. However it is important to link experiences so they can share position information.

Ideally we strive to store the minimum number of experiences needed to represent an environment, a surplus leads to wasted computation and memory resources. We want just the right number to facilitate long-term navigation.[2] This motivates us to fully exploit all the information in each and every experience, and also how they inter-relate. Failure to do so reduces our ability to re-localise within an experience, leading to an unnecessary genesis of a new experience.

The opportunity to exploit experiences optimally occurs naturally in two forms. The first is when starting a localiser to run on an experience, as every experience is not always relevant to the current position in the world. Consider a vehicle travelling from a relatively unchanging area to one of high visual variance, it needs to know when to activate the extra experiences related to that area. The second is when an experience temporarily becomes insufficient to explain the current visual feed (the localiser fails), for example the robot is taking a short detour. The experience will be relevant a short while later, at which point it needs to be re-started.

As global position look-up is not possible, we look for other ways localisers can be started or re-started. One option would be to use an external loop closer such as FAB-MAP (Cummins and Newman, 2009), where the live image is matched to a node in the saved experience. However, it has a relatively low recall rate at high precision, so we may find ourselves "lost" for some time before it fires. An alternative would be to use the new SeqSLAM (Milford and Wyeth, 2012) system. Another approach would be to annotate experiences with metadata, such Global Positioning System (GPS) points, or higher-order descriptions such as known road junctions or buildings, however these are not always available and the GPS signal can wander over time. There are occasions when we do need these higher-order loop-closing techniques, and these are discussed in Section 4.2, but before we need to resort to these external mechanisms, we look to exploit topological links between experiences.

4.1. The experience graph

As explained previously, an experience is a set of connected nodes. Consider if these were sub-graphs in a single large graph containing all experiences, G. Now assume that we have a method of introducing an edge between two sub-graphs that indicates the two connected nodes observed the same physical place in the world. When a successful localiser arrives at a node with one of these edges, it can query the edge for which experience (and location within that experience) is at the other end. The localiser associated with the other experience can then be informed where to start (if the localiser is not currently active) or to restart (if the localiser is lost). Using this connected graph, localisers can inform each other when to start, and to aid each other when lost.

The focus here is on how the inter-experience edges of G can affect the performance of the localisation system. By introducing as many high-quality edges as possible, we increase the information shared between experiences. These can then be used to aid localiser initialisation and restarting. In the results section we present four methods for discovering the structure of G. The first approach results in a very sparse G to highlight its importance, the second is a method which only uses the live stream to provide links. The final two are introspective processes which run between vehicle outings. Their aim is to take any new experiences created from the previous outing, introduce the sub-graph formed from their nodes to G, and then look for opportunities to create edges between the newly inserted nodes and all other nodes (experiences). We will outline these processes in more detail later.

4.2. External loop closing

Sometimes all of the localisation processes become lost, at which point the system does not know where it is in the plastic map. This may be because the current location has changed significantly (e.g. it snowed), or because the robot is exploring a new route. In either case the VO system will continue to process the live frame stream and will be saving

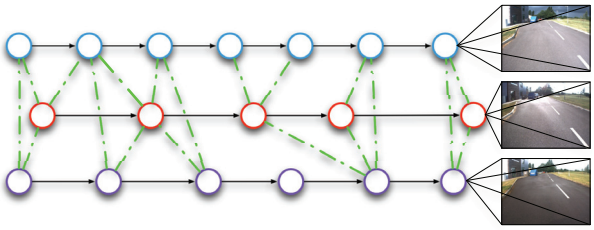


Fig. 5. Each experience can be represented as a graph, where the stereo frames are represented as nodes (shown as circles) and metric transformation information describes how they are connected (indicated by the directed black arrows). This figure shows three experiences, blue, red and purple. These can be formed into a single larger graph G, which contains all experiences. Edges can then be added between experiences (denoted as undirected dashed green lines) indicating that the nodes refer to the same physical place in the world. This is demonstrated for the far right node in each experience. Each associated image is of the same place in the environment, so edges can be created between them. By increasing the quality and quantity of these edges we assess what impact this has on localisation performance.

the output to a new experience. However, it is possible the robot will subsequently return to a place it can successfully localise in and regain its position in the plastic map. For a solution to this problem, which should become less and less frequent over repeated traverses, we use an external loop closer which can reinitialise lost localisers.

5. Implementation

Our VO system uses FAST corners (Rosten et al., 2008) and Binary Robust Independent Elementary Features (BRIEF) (Calonder et al., 2011) descriptors to perform the initial matching step. These associations are then refined using efficient second-order matching (Mei et al., 2008). BRIEF can achieve matching performances approaching those of scale invariant feature transform (SIFT) of SURF, but without the requirement of a GPU for frame rate operation.

All data was collected with the Wildcat vehicle shown in 6. A Point Grey Research Bumblebee2 stereo camera was mounted on the front bumper to capture image data. Colour images with a resolution of 512×384 were logged at 20 Hz. The vehicle has a dual-antenna OXTS RT3043 INS which was used occasionally to provide an external loop closure signal to the system, more details of this are given in Section 6. The Wildcat's internal computer uses two Intel Xeon X5570 2.93 GHz CPUs, offering 16 cores for computation (with hyper-threading).

After the set of relevant localisers has been decided, the localisation step of each localiser (i.e. determining the transform between the live frame and an experience) can be run in parallel as they are independent at this stage. Given that the data association and trajectory estimation

Fig. 6. The group's vehicle, the Wildcat, was used to collect 37km of visual data. The vehicle is equipped with a range of sensors including cameras, lasers and an INS. It also houses a high-end computer.

steps dominate the computation time, by being able to parallelise the localisers and having no GPU dependency, we are able to run at 15 Hz on a multi-core CPU. The localisers described in Section 3.3 use exactly the same processing pipeline as the VO, except the input landmarks are from an experience, not the live map.

The system needs a way to determine if a localiser is successfully localising, i.e. $\mathcal{Q}^i(\mathcal{F}_k) = 1$. For this we chose to compare the localisers translation from $\mathcal{F}_{k-1}$ to $\mathcal{F}_k$ with the same translation computed by the VO system running on the live stream, and require them to be in agreement to some percentage threshold. We investigate how this threshold affects the performance in the following results section.

6. Results

To test our proposed navigation approach we require many visits to the same environment over a large period of time. For testing we use the 0.7 km route around the Begbroke Science Park. The dataset was collected with the Wildcat vehicle over a 3-month period and contains 53 loops. The vehicle was driven at different times of day and under different weather conditions. The route, with results, is shown in Figure 7. The outer loop, denoted by the thicker line, was driven on the first 47 traverses while the last 6 traverses went via the inner loop, indicated by the thinner line. For illustrative purposes we controlled the signal from the external loop closer so it only fired at 14 predefined points on each loop. The points were spaced approximately evenly along each loop. The reported accuracy of the inertial navigation system (INS) in position is 0.1 m, however we found over the course of the 3 months that the reported position was only consistent to a few metres. The solution also suffered when the GPS signal was attenuated by trees or buildings. These affects are discussed further in Section 6.7.

6.1. Where were experiences saved?

Figure 7 indicates where we laid down experiences. The intensity of the plot shows how many experiences were saved at each point. We find that some regions require more experiences than others. The northern and eastern (the

Fig. 7. Overhead of the two routes visited during data collection. The outer loop, indicated by the thicker line, was driven 47 times, while the inner loop, shown by the thinner line, was driven 6 times. The intensity of the trace shows how many experiences have been laid down at each point. See Figure 8, 9, 10 and 11 for examples of places with low and high experience density.

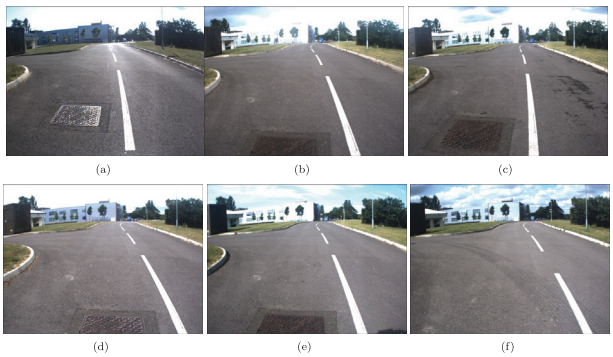


Fig. 8. Images taken from the northern road of the Begbroke Science Park across several months. We find this region has high visual stability and as a result reliable localisation can be achieved from relatively few experiences of the area.

image is orientated north up) regions of the route require relatively few experiences. Typified by open road flanked by buildings and low hedges, these areas are largely unaffected by lighting, weather and time of day changes. Example image sets from the northern and eastern road are given in Figure 8 and 9. We see here that despite been taken from many different days the scenes looks very similar, meaning that localisation is relatively easy from a few experiences.

Conversely we see that some places require many experiences as they exhibit high visual variation. One example of such a place is the northwest corner, where the camera observes a car park. The contents and configuration of this space varies daily meaning localisation against previous experiences is difficult. We show example scenes from this place in Figure 10. A second example is the south west corner. Here the vehicle is travelling in a section covered by overhanging trees. When the sun is out, it casts strong, intricate and ephemeral shadows. While these are useful for

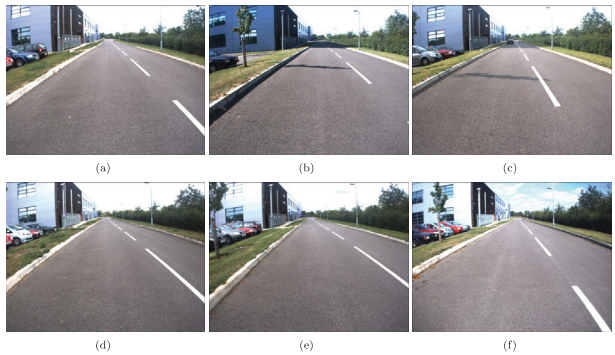


Fig. 9. Images taken from the eastern road of the Begbroke Science Park across several months. Again due to the unchanging nature of the scene it can be modelled with relatively few experiences.

the live VO system, they are often not encountered again meaning previous experiences are not useful. This encourages the use of the VO from the live image stream at all times. A set of example images from this region is shown in Figure 11.

6.2. When were experiences saved?

Figure 12 shows the results of the system for $N = 1$ on the Begbroke Science Park loops dataset. Here N is the minimum number of successful localisers permitted for the system not to save the live VO as a new experience. For example, if $N = 2$ and the system is successfully localising in one prior experience, the current VO output will be saved as a new experience. In Figure 12 the x-axis indicates the visit number and the y-axis is the percentage of the visit distance. We plot both the amount of time spent lost, and the amount of time spent saving experiences. For $N = 1$ these values are the same, but later when $N > 1$ these values will differ. We also plot the time of day each dataset was collected. Initially everything is new, so on the first visit everything is saved. On the second visit, approximately 60% of the time was spent lost, so the associated VO output during this time was saved, and so on. The first full traverse without any localisation failure occurs on the 16th visit: the percentage of the visit spent saving or lost goes to zero. The large jump around traverses 35–38 happens because for the first time we collected data as dusk fell. The roads also had standing pools of water and it was raining lightly, something the system had not encountered before. The second spike at visit 47 is caused by driving the inner loop for the first time. Suddenly no localisations are successful and the whole section is saved until the loop closer fires. Figure 14 shows examples of localisation failures on visit 4, where strong shadowing effects are encountered for the first time. Figure 15 shows examples of localisation failures on visit 38 when driving at dusk on wet roads. However, apart from these cases where we explicitly did something out of the ordinary to test the system (such as driving at dusk or a

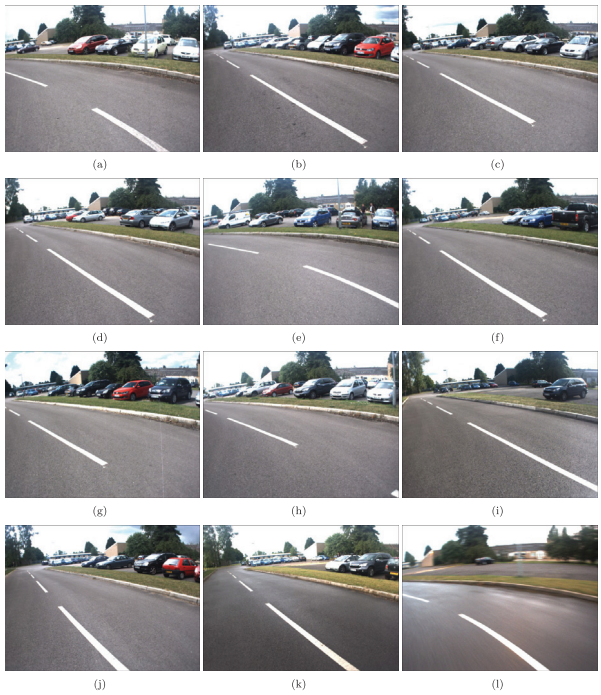


Fig. 10. Sample images from the northwest corner of the Begbroke Science Park. This area generates several experiences due to the car park. The vehicles in the area change daily meaning localisation against previous experiences is difficult.

Fig. 11. Sample images from the southwest corner of the Begbroke Science Park. The section of road is covered by overhanging trees which, when the sun is out, generates ever changing shadow effects. These are useful for the live VO system as they provide strong and reliable features, but due to their ephemeral nature are not useful for localisation on future visits. This results in a large number of experiences being generated for this area.

different route), we find that as we revisit the route we typically need to remember less and less each time. We will show later that the shape of this graph can be improved. Weather statistics for each visit are given in Figure 13.

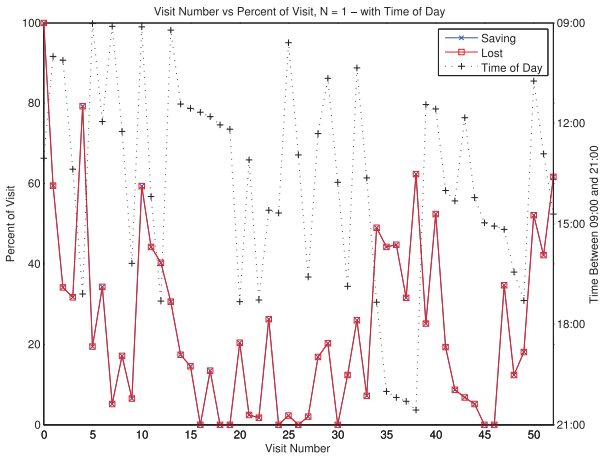


Fig. 12. System performance on the repeated loops of the Begbroke Science Park. Visit number is displayed on the x-axis and visit percentage on the y-axis. Visit percentage indicates the amount of time on that visit that was spent performing a certain action. Here we plot the amount of time the system spent saving (blue) and lost (red). As $N = 1$ here these values are equal. We also plot the time of day each dataset was collected (black crosses).

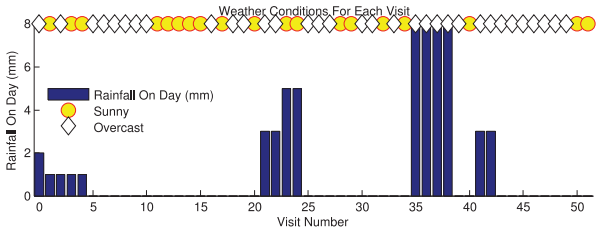


Fig. 13. Weather statistics for each traverse (Weather Online UK, 2012).

Fig. 14. Examples of localisation failure leading to the creation of new saved experiences on visit 4: (a), (c) from previously saved experiences; (b), (d) from the live stream.

6.3. Choices of N

By varying N we can influence how many similar representations of the environment the system needs. While this does lead to multiple representations of the same place, it also offers increased robustness. Should localisation fail in one experience for some reason there are still other experiences that can be used. Note that even though these experiences will share some common measurements we make no attempt to fuse, merge or average them as we want to explicitly avoid the difficult data association and fusion problems that would arise. We show plots of the system performance for $N = 2$ and $N = 3$. Note that now the saving (blue line) and lost (red line) are now different as it is possible to be localised but still saving. For example, consider the case when $N = 2$. If the system is only localised in one prior experience, the VO output will be saved as the number of successful localisers is less than N, however the system will not be lost. We also tabulate the results in Table 1.

We see that for $N = 1$, the system saves the least, but also spends the most time lost. Moving to $N = 2$ brings a significant increase in localisation performance: the time spent lost is reduced by nearly 10%. This comes for less than 5% increase in the amount of experiences saved. This boost is due to the increased robustness and redundancy introduced by having more than one experience available. The system is not so brittle to single localisation failures. We see further increases in performance for $N = 3$. Also note for increasing N we see more visits which were traversed with no localisation failures: the amount of time spent lost goes to zero more often.

Fig. 15. Examples of localisation failure leading to the creation of new saved experiences on visit 38: (a), (c) from previously saved experiences; (b), (d) from the live stream.

6.4. Are multiple experiences needed?

To demonstrate the value of saving multiple experiences and building the plastic map, we evaluated the performance of the system if we only made one previous experience available for localisation. This represents the model most traditional localisation methods use, where a single previous view of the world is privileged and assumed correct:

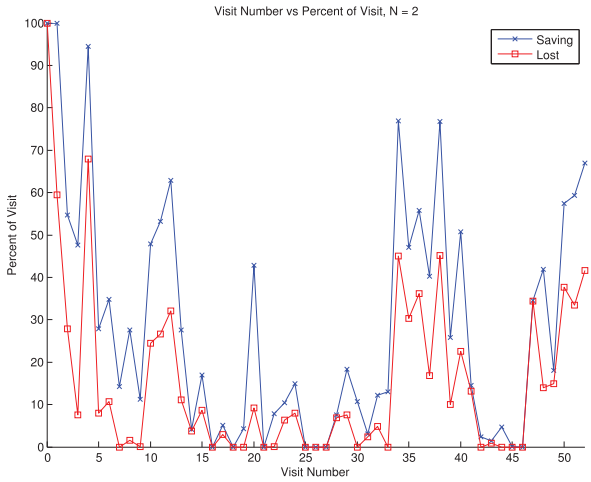


Fig. 16. Performance results for $N = 2$, with the percentage of each visit that was spent saving and lost shown. The system saved 29.25% of all of the possible experiences and was lost 15.73% of the time.

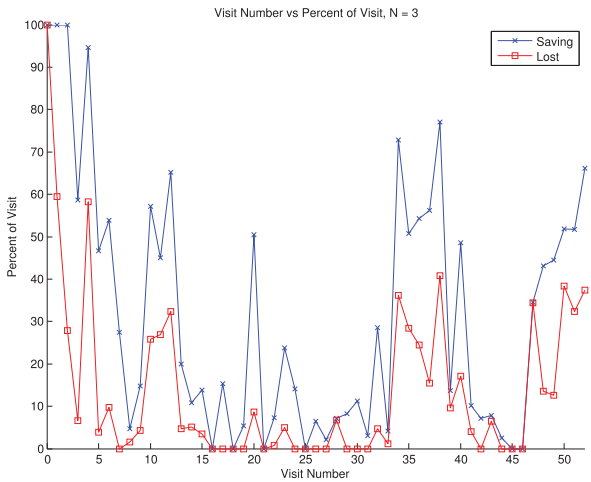


Fig. 17. Performance results for $N = 3$, with the percentage of each visit that was spent saving and lost shown. The system saved 31.95% of all of the possible experiences and was lost 14.12% of the time.

Table 1. Experience-based navigation results on the Begbroke loops for different choices of N. The percentages are of the total possible number of experiences, e.g. 100% saving would mean the system always saved the VO output; 50% lost means the system was lost half the time. Lower is better in both cases, and localisation is the one we really care about (not being lost). Here we see for modest increase in saving we get good boosts to localisation.

	Percentage saved	Percentage lost
$N = \mathbf{1}$	24.80%	24.80%
$N = \mathbf{2}$	29.25%	15.73%
$N = \mathbf{3}$	31.95%	14.12%

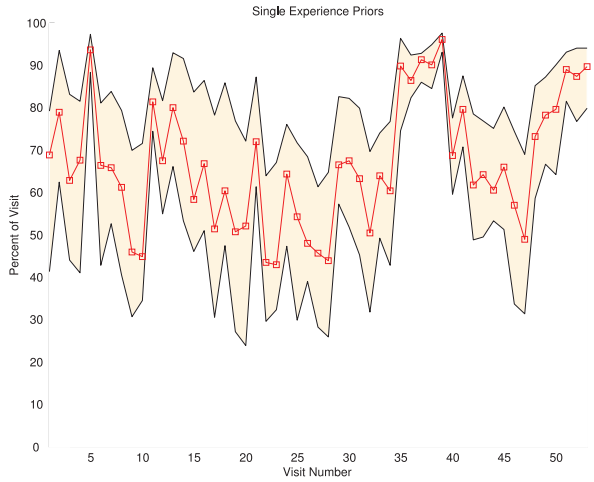
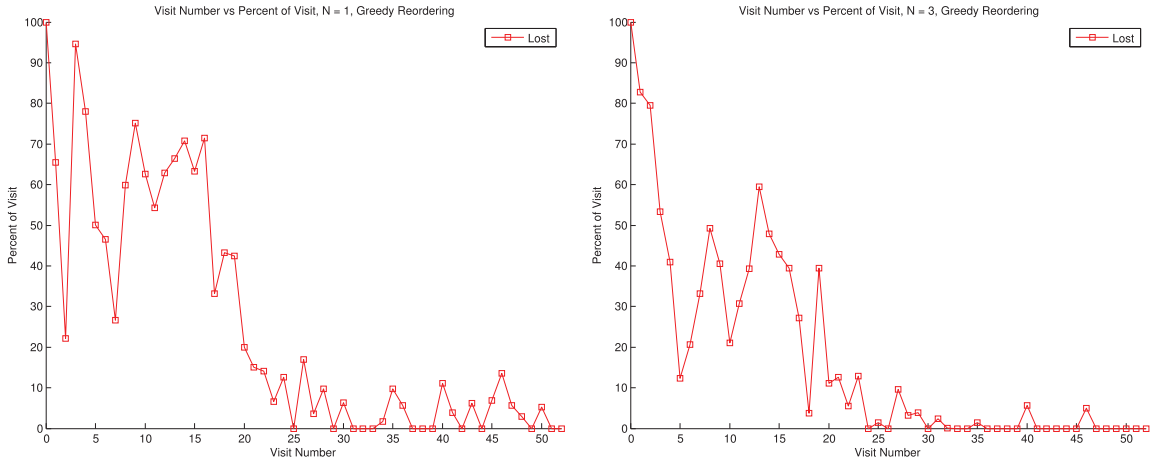


Fig. 18. This graph shows the performance of single experience priors. For each visit we used that as the single experience available. Localisation was then performed on all of the other visits using just the single available prior and the time spent lost, i.e. localisation had failed, was recorded. Here we plot the median performance of each visit when used as the single prior experience along with the 25th and 75th percentile bounds. We note that no visit performs well as a prior with the average time spent lost being 65.8%. This motivates the use of multiple experiences.

often this is the first time the world is experienced. We took each visit in turn and declared it the only available prior experience of the world. Using this single experience, we attempted to localise against it for all other visits. For each run we recorded the amount of time spent lost, i.e. localisation had failed. Results are shown in Figure 18. We plot the median performance for each visit and mark the 25th and 75th percentile bounds. Interestingly no single visit is particularly "good" as a prior for most other visits. Also we see the visits conducted around dusk (35–38) are also less useful as they share less similarity to the majority of the dataset. Across all visits the average time spent lost is 65.8%. This result suggests that reliable localisation results from a single prior will be difficult to achieve and motivates the use of multiple experiences working in concert to represent the environment.

6.5. *Long-term trends*

Over time, barring new routes, our approach produces an inversely proportional decay as the system captures the typical variation of the route. The order in which the data were collected is a biased sequence. Once we saw a signal of the system working we intentionally collected data we knew would cause problems to see how it would respond—such as driving at night or new routes. However the data could be ordered in 53! ways to produce different performance graphs. To make this point and demonstrate we can achieve an inversely proportional decay we performed a greedy reordering. We moved the 10 most surprising traverses of the



(a) (b)

Fig. 19. The system performance when the traverse order has been greedily re-arranged to present the system with the most surprising and diverse set of experiences first: (a) $N = 1$; (b) $N = 3$. The percentage of each visit spent lost is plotted in both cases.

outer route and the 6 inner traverses to the beginning and re-ran the system. We measured surprisal by the amount of traversal that was saved in the original configuration.

The resulting performance graphs for $N = 1$ and $N = 3$ are shown in Figure 19. By moving the most "interesting" and "surprising" visits to the beginning of the plastic map creation we see that accumulation of experiences is high before dropping off significantly. For $N = 1$ the system is still getting lost occasionally: this is due to the fragility of single localisation failures. When increasing N to 3 we see an improved robustness with many visits achieving no localisation failures.

6.6. Descriptor choice

BRIEF is a descriptor which describes the result of a set of t binary comparisons. Standard sizes for t are 128, 256 or 512, which corresponds to 16, 32 or 64 bytes per descriptor. We refer to these as BRIEF-{16,32,64} or B-{16,32,64}. The longer the descriptor the more discriminative it is. We find that when performing VO we can use a relatively short BRIEF descriptor as the sequential images from the camera are largely similar in appearance: the camera is unlikely do have undergone large translations or rotations between captures. However, when attempting to navigate against a prior experience, this assumption no longer holds true. To this end we investigate the performance results for different choices of BRIEF descriptor: 16, 32 and 64.

We present a plot of both the saving and lost performance of our system when using the different BRIEF lengths, for varying values of N in Figure 20. To create this graph we re-ran the whole system from scratch three times with a different descriptor length each time. We see that BRIEF-16 is significantly worse than BRIEF-32 and BRIEF-64 due to its shorter and less discriminative nature. It ends up saving more experiences and is localised for less time.

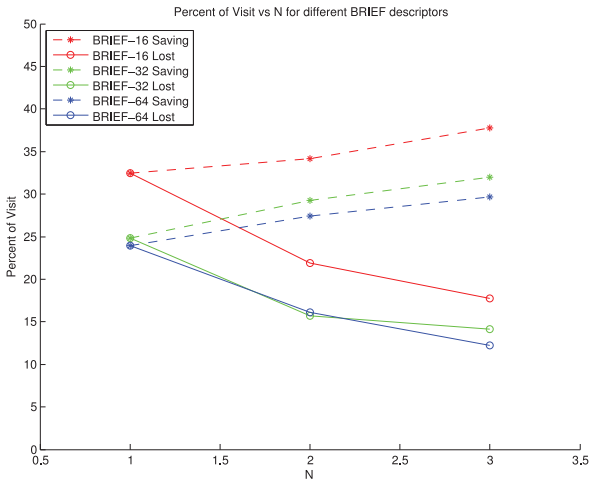


Fig. 20. Here we plot both the amount of time spent saving and lost for different values of N, when using different BRIEF descriptor lengths. The least discriminative is BRIEF-16 (red) which spends the most time both saving and lost. BRIEF-32 (green) and BRIEF-64 (blue): the more discriminative descriptors perform better, with BRIEF-64 achieving the best results.

Overall BRIEF-64 is best, it always saves the least, and apart from $N = 2$ is lost less than BRIEF-32. We also show the system performance for $N = 3$ with BRIEF-16 and BRIEF-64 in Figure 21. Clearly the BRIEF-64 graph is more suppressed (better) with several traverses achieving no localisation failures.

These results make sense. The data association problem when attempting to localise in an experience is more difficult due to changing environments, larger and more persistent spatial deviations, and having to find matches in a larger pool of candidates. In this setting the more discriminative descriptors perform better.

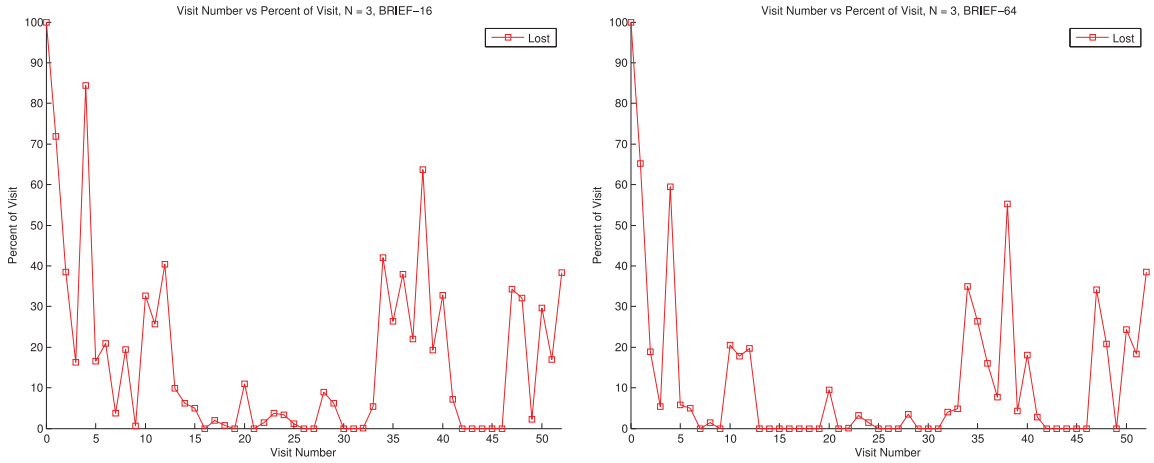


Fig. 21. Performance results for two different BRIEF descriptors: (a) BRIEF-16 and (b) BRIEF-64. Here we see increased performance when using more discriminative descriptors. The percent of each visit spent lost is plotted in both cases.

6.7. Connectivity of the experience graph

To evaluate how the connectivity of G affects the localisation performance we describe four approaches to discovering these connections and test each. The four approaches are No Discovery, Live Discovery, GPS Discovery and Refined Discovery. The first two approaches do not require any extra computation. The final two are introspective processes which are run between vehicle sorties. We now describe each in more detail.

6.7.1. No Discovery To demonstrate the importance of G's connectivity levels, in this approach we only allow inter-experience edges to be created which include the start of an experience. This enables experiences to be started if they are not active (otherwise they would never be used), but prevents the re-initialisation of lost experiences.

6.7.2. Live Discovery Here edges are created when the live stereo stream is localised in more than one experience. The resulting links are likely to be accurate due to the strong geometry required for successful localisation. However, by only using the live stream, G can only be changed when the vehicle is driving, and only in the regions covered by the vehicle on that trip. For example if the previous sortie included the car park, but it is not included on the next outing, the car park section cannot be connected to other experiences of that area. This dependency on the live system to create edges is undesirable.

6.7.3. GPS Discovery In this variant, for every experience created we also store the GPS position of every node. When an experience is created, for each new node, we find the nearest position in all other experiences via GPS. If the distance is less than some threshold, we introduce an edge connecting them.

6.7.4. Refined Discovery In this final version we attempt to recover from the potential variations that GPS can suffer from when recorded over long periods of time. As our experiences are collected over several months, our data are affected by the potentially wandering nature of the GPS signal. We achieve an initial match to other experience nodes using the same approach as GPS Discovery. We then look to refine this estimate by matching the new node against a sequence of nodes surrounding the GPS suggestion, based on the same stereo frame to frame estimation techniques used in the VO system.

Given the new node n^* and a candidate node proposed from GPS, n_c, we take a small window of nodes either side of n_c, denoted $\{n_c\}_w$ and compute the transform from n^* to each node in the window. Next, assuming all transforms are valid, we compute the translation from n^* to each $\{n_c\}_w$. If we find a local minimum which is not on the bounds, i.e. n^* really did pass by the window $\{n_c\}_w$, we take the candidate node in $\{n_c\}_w$ with the smallest translation to n^* to be the same place. The appropriate edge is then created in G.

6.7.5. Impact of discovery choice We present results for $N = 1$ and 3 in Figure 22 and Figure 23. The saved and lost statistics for each variant, along with varying N is shown in Table 2. Performing Refined Discovery results in a 26%, 17% and 10% improvement for $N = 1$, 2 and 3, respectively, when compared with Live Discovery. This decrease in improvement makes sense. When operating at $N = 3$, the Live Discovery approach is able to create a greater number of links, reducing the number of new ones that the introspective approaches can introduce. Interestingly GPS Discovery does not perform as well as Refined

Table 2. Percentage of saved output and time spent lost (lower is better) for the different graph discovery methods. GPS and Refined Discovery methods work best.

	Discovery type			
	Saving			
	No Discovery	Live	GPS	Refined
$N = 1$	36.33%	24.80%	21.32%	18.47%
$N = 2$	46.07%	29.25%	27.19%	24.13%
$N = 3$	50.53%	31.95%	30.32%	28.51%
	Lost			
$N = 1$	36.33%	24.80%	21.32%	18.47%
$N = 2$	26.55%	15.73%	12.83%	12.01%
$N = 3$	23.28%	14.12%	11.96%	9.09%

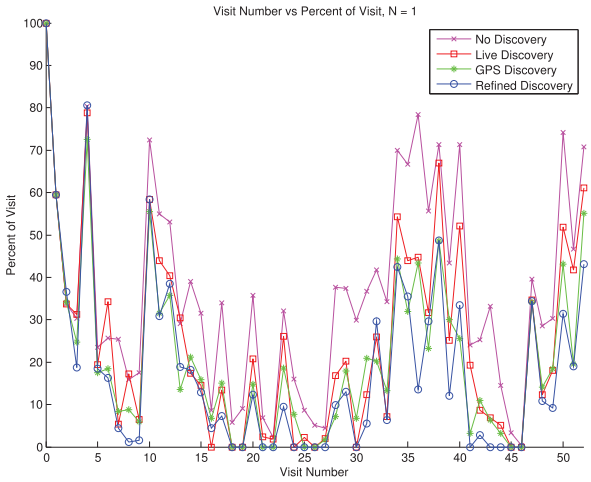


Fig. 22. System performance when using different graph structure discovery approaches. For each mode of operation the percentage of visit spent lost is plotted. Here $N = 1$ for all runs.

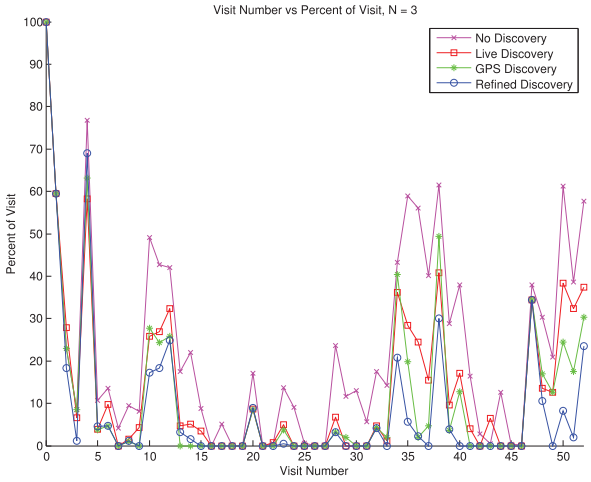


Fig. 23. System performance when using different graph structure discovery approaches. For each mode of operation the percentage of visit spent lost is plotted. Here $N = 3$ for all runs.

Discovery. This is likely to be caused by the quality of the inter-experience edges in G being substandard due to the slightly inconsistent nature of the GPS signal.

Note how in both Figure 22 and Figure 23, GPS and Refined Discovery do not always out perform the original Live Discovery variant. This is because performance on a particular run is directly tied to what has been saved previously. As the Live Discovery stores more than is necessary in earlier runs, on some later outings it has more experiences to draw from and sometimes stays localised for longer than the other systems. However, Table 2 shows that Refined Discovery performs best overall, followed by GPS Discovery and then Live Discovery. No Discovery always performs worst and saves significantly more experiences than the other variants. This of course is expected as the system becomes brittle to localisation failures and is wholly dependent on the external loop closer to put it back on the right path.

We see from Table 2 that the offline methods store fewer experiences and are lost less, this implies they are making better use of the information they already have stored. By increasing the quantity and quality of edges in G, we have shown that we can stay localised for longer and are required to save less, as we are making better use of the information we currently have.

6.8. Sensitivity to lost threshold

The experience based navigation approach requires some metric to measure if a localiser is lost. We chose to compare the localiser's motion estimate with that of the VO. Given this, we must then decide on some threshold at which the two disagree. To investigate our systems sensitivity to this parameter we ran it with thresholds requiring 5%, 10%, 15% and 20% similarity. The saving and lost percentages for $N = 2$ is given in Table 3. We also plot the saving and lost graphs in Figure 24.

We see that for more stringent (lower) thresholds the general performance degrades. This is because the system is demanding that the inter-frame motion estimates of the localiser and the VO agree more closely. In reality this is difficult to achieve as we are trying to match image features to different sets of landmarks, each of which will have their own errors. We find that the average inter-frame error of the VO system itself is around 5% and that it attempted to match two frames adjacent in time. For thresholds between 10% and 20% we see the difference in performances are less. We also note that the general shape of the graphs in 24 are the same, with the stricter 5% threshold generally being slightly higher than the rest. This suggests that while the choice of this threshold will impact the performance of the system, it should not affect the overall functionality provided it is reasonable.

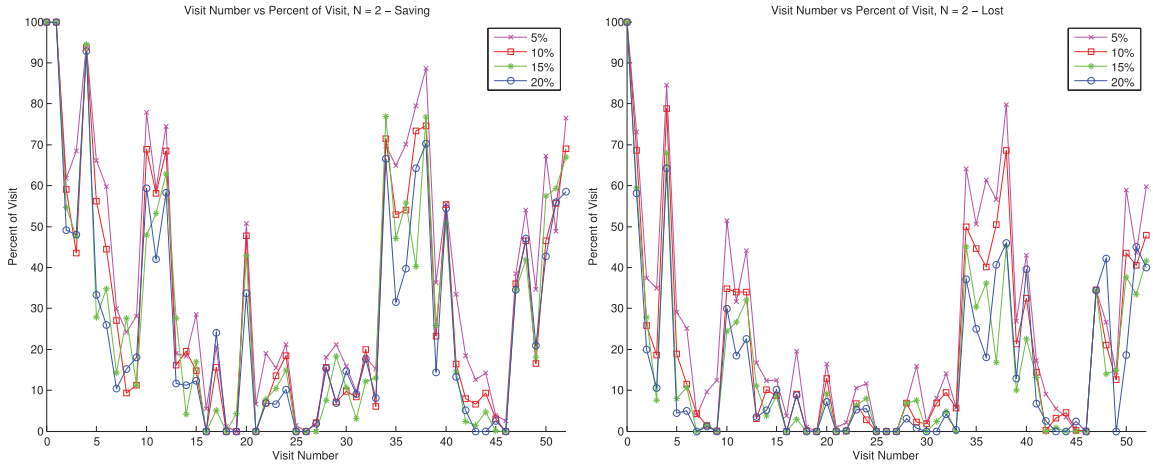


Fig. 24. Plots show the saving (a) and lost (b) performance graphs for varying lost thresholds. We see that the general shape of the graphs are largely similar for different thresholds.

Table 3. The systems sensitivity to the motion similarity threshold. To decide whether the system is lost, the localisation performance is compared against the VO estimate. The localiser is declared lost if its estimate differs by a certain percentage to the VO. Here we show the saving and lost performance of the system with different similarity thresholds for $N = 2$.

	Motion similarity threshold			
	5%	10%	15%	20%
Saving	38.11%	31.71%	29.25%	27.62%
Lost	25.53%	19.76%	15.73%	15.08%

6.9. Weather conditions

We also classified each traverse as either overcast or sunny (the weather for each visit is shown in Figure 13). We ran the system using only overcast or sunny experiences, the results of which are shown in Figure 25. (We removed the four dusk and six inner loop traverses.) Interestingly, when provided with only overcast visits, the system quickly accumulates sufficient experiences to navigate successfully. Conversely when presented with the purely sunny traverses localisation is more difficult and the system suffers a reduction in performance. We believe this is happening because the direct sunlight casts different shadowing effects which make localisation against previous experiences difficult.

6.10. Timing performance

Finally we show the performance of the system running on the Wildcat hardware in Figure 26. Shown is the number of successful localisers and timing performance for each frame on visit 47, which is the first traverse of the inner loop. Localisation is successful until frame 1296, at which point the vehicle turns onto the inner loop. When this happens a new experience begins as localisation fails in all previous experiences. At frame 2239 the external loop closer fires and results in successful localisation, so saving of the new experience stops. Despite varying numbers of active localisers the timing per frame typically stays under 100 ms, while the average for the successful localisation part of the sequence (i.e. not including frames 1296–2239) is 53 ms. This is possible as the localisation steps of the live image to each experience are independent of each other, making it possible to parallelise the process across multiple CPUs. Also note that in areas with many experiences, the majority of these will not be similar to the live image stream, meaning they will not all be used, reducing the potential computational load.

7. Discussions and conclusions

7.1. Sensor choice

As noted previously, our approach has been implemented using a stereo camera, but would also be applicable to other sensor modalities, assuming that motion estimation and localisation modules are available. One obvious choice would be lidar sensors. Choosing a different sensor may provide more robustness to certain types of scene change. For example, lidar sensors will clearly perform better than vision sensors when scene change occurs from varying lighting effects. This would allow it to deal better with the sunny traverses that the vision-based system struggled with in Figure 25. This change in sensor modality is exactly the approach McManus et al. (2012) took to extend their original teach and repeat system (Furgale and Barfoot, 2010). By using a lidar-based system they were able to reduce the impact of lighting on the performance of the system. However, lidar does not completely solve this issue. McManus

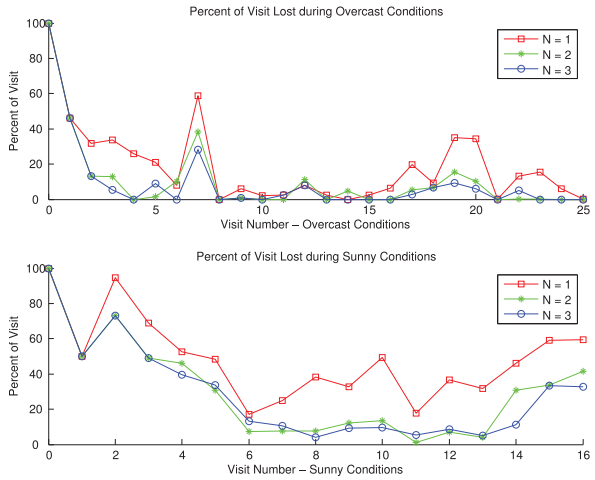


Fig. 25. Performances of only overcast (top) versus only sunny (bottom) traverses. Note the constant offset for sunny conditions compared with overcast conditions, this is probably caused by strong shadows making localisation difficult in previous experiences. The percentage of visit spent lost is plotted in both cases.

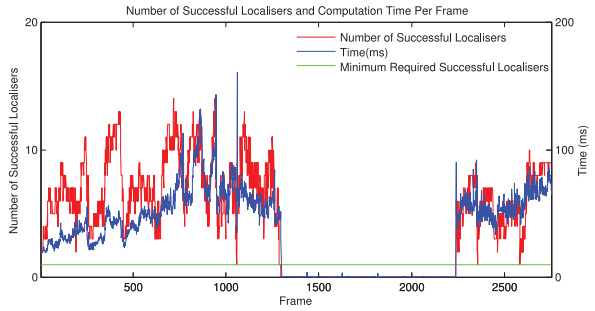


Fig. 26. Timing performance (using the Wildcat hardware) and number of successful localisers for each frame on visit 47, the first traverse of the inner loop. Only when the robot is driving on the previously unseen road is the VO saved, otherwise localisation is always successful. In the regions of previously seen road the average number of successful localisers is 7, and the average localisation time per frame during this period is 53 ms.

et al. (2012) noted dips in performance caused by the high noon sun, and rain soaked ground producing different reflectance values compared to when dry. Others have also noticed lidar systems suffer when surfaces become wet or icy (Borrmann et al., 2010). Further, such a system would still need to lay down multiple experiences in regions which undergo gross structural change, such as the car park shown in Figure 10. It should also be noted the lidar sensor used by McManus et al. (2012) can only operate at 2 Hz and due to its scanning nature, if used while the robot is moving produces warped images.

7.2. Forgetting experiences

We find for the majority of the route used in our experiments, the number of experiences saved tends to a constant, as the low number of visual modes are quickly discovered. Significant sections of Figure 7 are captured with just a handful of experiences. However, there are notable areas that have highly unstable appearance, resulting in significant numbers of experiences being saved in these regions. The prime example being shown in Figure 11, overhanging trees casting a unique set of shadows each time the sun is shining. After 53 runs, over 30 experiences have been saved in the most difficult region. Another troublesome region previously discussed is when passing the car park shown in Figure 10. To prevent new experiences being created on almost every visit to these areas, these regions could be marked as un-mappable after a certain number of attempts. When entering this region in future, the system could switch over to purely relying on the VO module, which is very accurate over shorter distances, and the external loop closure could be used when returning from longer problematic sections.

In this work we have not considered deletion policies for experiences. The policies covered in the previous work that focus on maintaining diversity and recency would be obvious choices if it was required to keep computation costs down. Another approach would be to select only *appropriate* experiences. If we know we are navigating in sunny conditions in the middle of summer, there is little point attempting to localise in an experience captured on a snowy winter morning. By judiciously selecting experiences, computational savings could easily be made.

7.3. Dynamic objects

The overwhelming majority of ego-motion estimation algorithms used in exploration or localisation assume the world is static, and so the issue of dynamic objects in the scene is one the SLAM community has been aware of for some time (Bailey and Durrant-Whyte, 2006). In this context "dynamic objects" refers to other things moving in the world, e.g. humans and cars. Most modern approaches use probabilistic filters to reject measurements that do not fit with the consensus as outliers. For example our VO system uses random sampling consensus (RANSAC) to compute an initial guess of the camera motion, enabling the system to be robust to outlier matches provided by the front-end. Some systems go further, explicitly tracking the moving objects (Wolf and Sukhatme, 2004; Schindlera et al., 2010). These solutions are successful provided the amount of dynamic motion in the scene is sufficiently small.

In the context of long-term navigation, objects which are not necessarily dynamic when they are first encountered, can easily move between visits, causing dynamic scenes. An obvious example are parked vehicles. These cases can be solved to an extent by the same techniques described above, assuming that outliers can be detected and rejected. However, if the scene changes sufficiently this approach begins to break down. For example, consider the case of a car park. The configuration of the parked vehicles will

change daily, leading to regularly varying and ephemeral scene appearance.

8. Conclusions

In this paper we have demonstrated continuous localisation of a road vehicle in drastically changing lighting and weather conditions over a 3-month period. This was possible because we adopted the notion of plastic mapping. We focussed not on building a single monolithic map or inferring a latent underlying state which explains all observations of the workspace but on creating a composite representation constructed from multiple overlapping experiences. This representation is only as complex and rich as it needs to be. It handles both drastic and creeping changes in the same way: as soon as prior experiences fail to adequately describe the present a new experience is saved for future reference. We have shown our system working in real-time embedded on a vehicle. We have shown the advantages of plastic maps in localisation performance (robustness) and have demonstrated the asymptotic behaviour of plastic map maintenance. Starting with a core competency (in our case VO) day on day, week on week, we are extending our vehicle's operating envelope; gradually making the extraordinary ordinary.

Funding

Winston Churchill is supported by an EPSRC Case Studentship with Oxford Technologies Ltd. Paul Newman is supported by an EPSRC Leadership Fellowship (number EP/I005021/1). This work has also been supported by BAE SYSTEMS.

Notes

1. So every landmark has a unique ID.
2. We note that certain parts of the environment require more experiences than others, on account of greater visual variation.

References

Bailey T and Durrant-Whyte H (2006) Simultaneous localisation and mapping (SLAM): part II state of the art. *Robotics and Automation Magazine* 13(3): 108–117. DOI: 10.1109/mra.2006.1678144.

Bay H, Ess A, Tuytelaars T and Gool LV (2008) SURF: Speeded Up Robust Features. *Computer Vision and Image Understanding* 110: 346–359.

Biber P and Duckett T (2005) Dynamic maps for long-term operation of mobile service robots. In: *Proceedings of IEEE Robotics: Science and Systems*, Cambridge, MA.

Borrmann D, Elseberg J, Rauniyar SS and Nuchter A (2010) Lifelong 3D mapping – monitoring with a 3D scanner. In: *IEEE International Conference on Intelligent Robots and Systems (IROS) - Workshop on Robotics for Environmental Monitoring*.

Burgard W, Stachniss C and Haehnel D (2007) Mobile robot map learning from range data in dynamic environments. In: *Autonomous Navigation in Dynamic Environments* (*Springer Tracts in Advanced Robotics*, vol. 35). New York: Springer, p. 27.

Calonder M, Lepetit V, Ozuysal M, Trzinski T, Strecha C and Fua P (2011) BRIEF: computing a local binary descriptor very fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(7): 1281–1298.

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

Dayoub F, Cielniak G and Duckett T (2011) Long-term experiments with an adaptive spherical view representation for navigation in changing environments. *Robotics and Autonomous Systems* 59(5): 285–295. DOI: 10.1016/j.robot.2011.02.013.

Dayoub F and Duckett T (2008) An adaptive appearance-based map for long-term topological localization of mobile robots. In: *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS)*, pp. 3364–3369.

Furgale P and Barfoot TD (2010) Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics* 27(5): 534–560.

Konolige K, Agrawal M and Solà J (2007) Large scale visual odometry for rough terrain. In: *International Symposium on Research in Robotics (ISRR)*.

Lategahn H and Stiller C (2012) City GPS using stereo vision. In: *Proceedings of IEEE International Conference on Vehicular Electronics and Safety*.

McManus C, Furgale P, Stenning B and Barfoot TD (2012) Visual teach and repeat using appearance-based lidar. In: *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*.

Mei C, Benhimane S, Malis E and Rives P (2008) Efficient homography-based tracking and 3-D reconstruction for single-viewpoint sensors. *IEEE Transactions on Robotics* 24(6): 1352–1364. DOI: 10.1109/TRO.2008.2007941.

Milford M and Wyeth G (2009) Persistent navigation and mapping using a biologically inspired SLAM system. *The International Journal of Robotics Research* 29(9): 1131–1153.

Milford M and Wyeth G (2012) SeqSLAM: visual route-based navigation for sunny summer days and stormy winter nights. In: *IEEE International Conferece on Robotics and Automation (ICRA 2012)*, River Centre, Saint Paul, MN. IEEE, pp. 1643–1649.

Newman P, Sibley G, Smith M, et al. (2009) Navigating, recognising and describing urban spaces with vision and laser. *The International Journal of Robotics Research* 28(11–12): 1406–1433. DOI: 10.1177/0278364909341483.

Rosten E, Porter R and Drummond T (2008) Faster and better: a machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(1): 105–119.

Schindlera K, Essb A, Leibec B and Gool LV (2010) Automatic detection and tracking of pedestrians from a moving stereo rig. *ISPRS International Journal of Photogrammetry and Remote Sensing* 65: 523–537.

Sibley G, Mei C, Reid I and Newman P (2010) Vast scale outdoor navigation using adaptive relative bundle adjustment. *The International Journal of Robotics Research* 29: 958–980.

Taneja A, Ballan L and Pollefeys M (2011) Image based detection of geometric changes in urban environments. In: *Proceedings of IEEE International Conference on Computer Vision (ICCV)*.

Weather Online UK (2012) http://www.weatheronline.co.uk.

Wolf D and Sukhatme GS (2004) Online simultaneous localization and mapping in dynamic environments. In: *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*.