Practice Makes Perfect? Managing and Leveraging Visual Experiences for Lifelong Navigation

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 three month period at different times of day, in different weather and lighting conditions. In all, we process over 136,000 frames captured from 37km of driving.**

I. Introduction

To achieve long term autonomy robotic systems must be able to function in changing environments - we see this as a big challenge. Change can come from many sources: sudden structural change, lighting conditions, time of day, weather and seasonal change. To illustrate, consider the problem of ego-motion estimation with a camera mounted on a robot operating outdoors. This is a richly mined area of research and immediately we reach for a visual navigation (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? 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 have to contain features from every possible scene modality. The things we see on a given tree in winter are simply not the things we see in summer; the details we see on a wet road at high noon are different to those we see at dawn when the road is dry. 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.

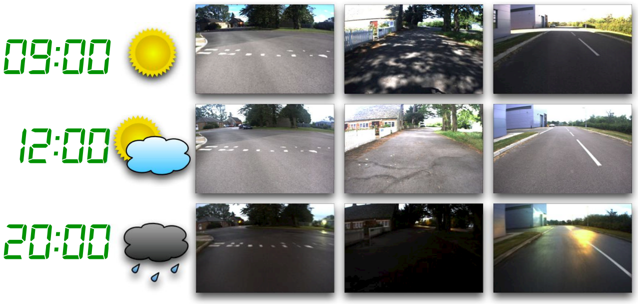


Fig. 1. The same place can look very different, depending on when it is observed. This variation may come from structural change, lighting conditions or through shifting seasons. Attempting to produce a single trajectory from multiple experiences may be difficult due to lack of correspondences.

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. On the initial visit to a new area we save a constellation of visual features like most systems. For reasons that will become clear we call this an "experience" rather than a map. 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. This is shown graphically in Fig. 2. Importantly this methodology causes the system to "remember" more representations for regions that change often, and fewer for regions that are more staid. 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.

A core competency on which we depend is a visual odometry (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

The authors are with the Oxford University Mobile Robotics Group {winston,pnewman}@robots.ox.ac.uk

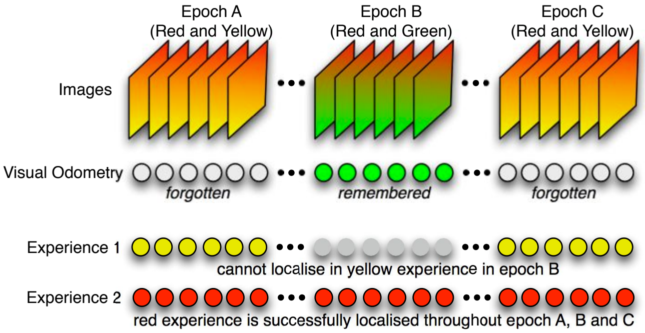


Fig. 2. An overview of our approach (best viewed in colour). A visual odometry (VO) system continuously consumes the live image stream. In parallel a series of localisers attempt to localise each live frame in their own experience. In epochs A and C both localisers successfully localise the frames in 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.

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 and Mei [1], which allows us 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 the ones described above are supplied.

We have tested our system on 53 runs of two laps of a 0.7km circuit, covering 37km in total and consisting of over 136,000 stereo frames. The data were collected over a three month period at many different times of day and in different weather conditions.

The remainder of the paper is organised as follows. Section II looks at related work. Section III presents the main idea in this paper and Section IV briefly outlines the implementation details. Section V presents the results.

II. Related Work

As localisation and mapping algorithms mature, the problem of truly long term navigation is becoming more urgent. A common problem in robotic navigation is changing environments and is a major obstacle to realising lifelong autonomy. Traditional mapping approaches often create their map once on the initial visit and hope that this will be sufficiently close in appearance to be useful on subsequent visits. Recently there have been some attempts to combat these problems.

Konolige and Bowman [2] develop their view-based maps system [3] to adapt to change in an indoor environment. The system creates a skeleton graph of keyframes from a VO system. The views stored at nodes are then updated and deleted based on a scheme designed to preserve view diversity while limiting the maximum number of views. Their skeleton map is in a single frame of reference and is incrementally optimised via Toro [4]. They show results for an office that includes moving people, furniture and changing lighting conditions. Milford and Wyeth's RatSLAM system [5] uses the notion of an "experience map" which stores a collection of experiences. Their experiences, unlike ours, are point places in the world, containing a pose estimate and a view of the scene, as well as metric transforms (from odometry readings) to other experiences. Over time the experience map is relaxed to minimise the difference between the absolute experience locations and their transitions to others. A new experience is created when the robot visits a new place or the world changes visually, experiences are also deleted to keep the density down. Note that an experience in our framework is not the same as theirs, ours is a chain of images and poses which can cover a large spatial area. Also experience creation is driven by localisation failure, not directly by visual changes. Biber and Duckett [6] sample laser maps at a series of time scales to create a single map that has both long term and short term structure. This allows them to model both short term dynamic objects as well as long term structural change. They show improved localisation accuracy over the period of weeks for a busy lab environment. We also note the work of Furgale and Barfoot [7] who develop a teach and repeat system using vision in an outdoor environment. During the teach phase the robot creates a series of submaps using VO. A localisation module is then used to retrace the original path. They show results over several kilometres, however they do not attempt to capture changing appearance from the original traverse.

In contrast to these previous approaches we only store new experiences when our current ones are insufficient. This results in each stored experience representing a different visual mode so we do not need to worry about deletion. As we allow each experience to exist in its own frame of reference, localisation is trivially parallelised and allows the use of multiple experiences at any time. This also allows significantly different visual modes to represent the same physical space.

III. Plastic Maps

The contribution of this paper is the development and demonstration of a framework for long-term navigation in a changing environment. With our proposed methodology, the robot constantly produces a 3D model of the environment

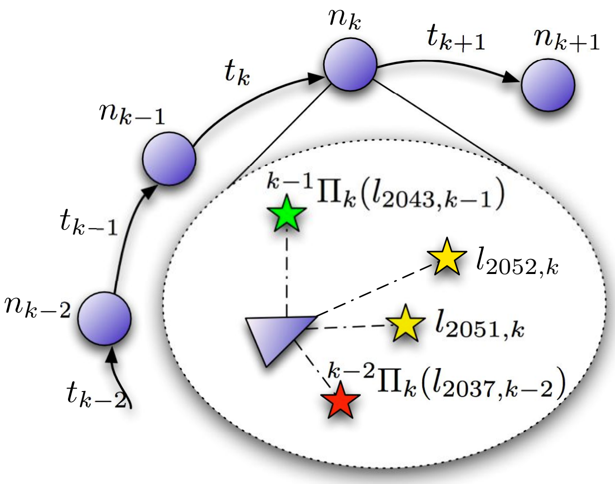


Fig. 3. An example of the output from our visual odometry system. Each frame is represented by a node, n_k. Nodes are linked by the transformation that describes the camera motion between $\mathcal{F}_{k-1}$ and $\mathcal{F}_k$ as a 6 DoF vector t_k. Each node also has a set of visible landmarks which can be transformed so they are relative to k via eq. (2).

from a VO system that takes the live image stream as its input. At the same time the robot opportunistically attempts to localise in saved experiences. If localisation is successful in a previous experience, we believe our representation of the current place to be sufficient. However if we cannot localise in previous experiences, we create a new one by saving the output of the VO system. We denote experiences by $\mathcal{E}$ and refer to the jth experience as ${}^j\mathcal{E}$. The set of all experiences creates the plastic map.

$$\mathcal{PM} = \{{}^j\mathcal{E}\} \ \forall j \tag{1}$$

Our method relies on the ability to navigate locally and being able to "close the loop" - recognising when we have returned to a previously visited place. Both of these are proven competencies. Metric trajectory estimation systems using laser [8] and vision [9], [10] have been demonstrated while large scale topological loop-closure systems using vision have been achieved [11] and crude GPS with a <10m error is often available in outdoor environments. In this paper we present our method using images, but it could be applied to other sensor types if equivalent competencies are provided. We now outline the requirements of the local navigation system before explaining how this is used to create the plastic map.

A. Experiences

Our VO system operates on 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 $t_k = [x, y, z, \theta_r, \theta_p, \theta_q]^T$ where θ_r, θ_p and θ_q are roll, pitch and yaw respectively. If new 3D landmarks are initialised as $\mathcal{F}_k$ is processed then these are also attached to n_k. We denote the ith such landmark attached to n_k, where i is a global index[1], as $l_{i,k} = [x, y, z]^T$ - a vector in the camera frame at time 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 - the ones in which they were initialised.

Often we wish to express a landmark described in frame p in a different frame q. This operation of transforming $l_{*,p}$ to $l_{*,q}$ is represented by ${}^p\Pi_q$ such that

$$l_{*,q} \leftarrow {}^p\Pi_q(l_{*,p}) \tag{2}$$

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

B. Localisation

We now introduce the idea of a localiser. Each localiser runs over a saved experience, 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 frame $\mathcal{F}_{k-1}$. The landmarks are taken from the local region surrounding the previous position in the experience. Additionally 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.

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

$$\mathbf{L}({}^j\mathcal{E}, \mathcal{F}_k) = \begin{cases} 1 & \text{if localised} \\ 0 & \text{if lost.} \end{cases} \tag{3}$$

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

$$m \leftarrow {}^j\bar{\mathcal{E}}(\mathcal{F}_k) \tag{4}$$

Once a localiser is declared lost, it stays in this state until receives outside assistance, discussed in Section III-E.2.

C. Experience Creation

We define N to be the minimum acceptable number of successful localisers at any time. For each frame, the set of active experiences $\mathcal{A}$ is computed (discussed in Section III-E.1). For each experience in $\mathcal{A}$, its associated localiser runs and the result is a binary vector S indicating the success

[1] So every landmark has a unique ID.

Algorithm 1 Plastic Mapping

```
bool Saving
while 1 do
    $\mathcal{F}_k = GetFrame()$
    $\{n_k, t_k\} \leftarrow \mathbf{VO}(\mathcal{F}_k)$
    $\mathcal{A} = ComputeActiveLocalisers(\mathcal{PM})$
    $S = Localise(\mathcal{A}, \mathcal{F}_k)$
    if $|S| < N$ then
        Saving = True
        ${}^j\mathcal{E} \leftarrow \{{}^j\mathcal{E}, {}^j\mathcal{E}_m\}$
    else if $|S| >= N$ && Saving==True then
        Saving = False
        $\mathcal{PM} \leftarrow \{\mathcal{PM}, {}^j\mathcal{E}\}$
    end if
end while
```

or failure of each. If $|S|$ falls below N a new experience ${}^j\mathcal{E}$ is created and the VO output is saved until $|S|$ returns to greater than or equal to N. This process is described by Algorithm 1.

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.

D. Interweaving Experiences

The plastic map stores many experiences covering an unbounded spatial area, in addition to capturing different appearances of the same area, thus they will not all be relevant all the time. The robots' spatial position in its environment will not be captured by all experiences. Therefore it is necessary to calculate which are applicable for $\mathcal{F}_k$. Experiences are not stored in a single frame of reference, so it is not possible to integrate local transforms to estimate the position in one from another. To overcome this problem we introduce the idea of places, which create topological links between experiences. We refer to all places as $\mathcal{P}$ and the zth place as $\mathcal{P}^z$. $\mathcal{P}^z = \{{}^j\mathcal{E}_m\}$, is the set of all nodes (taken from various experiences) which are known to have been concurrently localised against. This is a set of camera frames viewing the same physical place.

When $\mathcal{F}_k$ is simultaneously localised in more than one experience we can create a place. By querying each successful localiser for its nearest node, we can create a place with the set $\{{}^j\mathcal{E}_m\}_k$. However it is also possible a previously created place contains some of these nodes, in which case we merge the sets created from $\mathcal{F}_k$ and the previous place.

$$\mathcal{P}^z = \mathcal{P}^z \cup \underbrace{\left[\bigcup_j {}^j\bar{\mathcal{E}}(\mathcal{F}_k)\right]}_{\mathcal{F}_k} \quad (5)$$

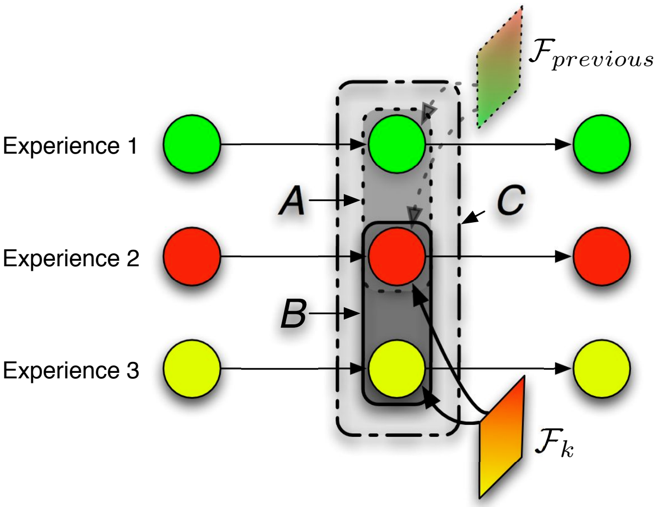


Fig. 4. A simple example of how places are created and updated. A previous frame, $\mathcal{F}_{previous}$, was able to simultaneously localise in experience 1 and 2 creating place *A*. $\mathcal{F}_k$ is able to simultaneously localise in experience 2 and 3, creating place *B*. As *A* and *B* share nodes (from experience 2), they are merged to create place *C*.

Where $\mathcal{P}^z$ on the right hand side is potentially an empty set depending on whether the nodes generated from $\mathcal{F}_k$ are in previous places. A simple example of place creation and merging is shown in Fig. 4. Note that the shared node in experience 2 causes places *A* and *B* to be merged into place *C*. A place is also created when a new experience begins saving, using the last successful localisation nodes.

Places are used to link experiences. We require the ability to query all places with a set of experience nodes, and receive the set of places which contain these nodes. This query is defined via Ω.

$$\{\mathcal{P}^z\} \leftarrow \Omega(\{{}^j\mathcal{E}_m\}) \quad (6)$$

E. Leveraging Places

1) Computing Relevant Experiences: As noted above, not all stored experiences will be relevant for $\mathcal{F}_k$ as they will not cover the robots current position. Therefore we allow localisers to take two states, active and inactive. Before processing $\mathcal{F}_k$, the set of successful localiser positions from the previous frame, $\{{}^j\mathcal{E}_m\}_{k-1}$, are used to generate the set of related places:

$$\{\mathcal{P}^z\}_{related} \leftarrow \Omega(\{{}^j\mathcal{E}_m\}_{k-1}) \quad (7)$$

Each inactive localiser then queries $\{\mathcal{P}^z\}_{related}$ for the presence of its own nodes, and if any are present, it activates itself at the retrieved position. When the robot moves beyond the range of an experience, the associated localiser sets itself to inactive.

2) Reinitialising Lost Localisers: Localisers get lost because they can no longer track the current frame, however this may only be a temporary state. They may become relevant again a short while later, and therefore it is important to attempt to reinitialise lost localisers. Failure to do so results

Fig. 5. The group's vehicle, the Wildcat, was used to collect 37km of visual data.

in the system saving more experiences than necessary. This problem is identical to computing the relevant experiences discussed in Section III-E.1. Given a known location in one experience do we know our position in another through places? Periodically lost localisers query $\{\mathcal{P}^z\}_{related}$ to see if they can be reinitialised.

F. 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 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.

IV. Implementation

A key property of our system is that once the set of relevant localisers has been computed, localisation of the current live frame in each one is independent and so can be run in parallel. Given that the data association and trajectory estimation steps dominate the computation time, by parallelising them we are able to process frames at 15Hz. To achieve robust data association we use Binary Robust Independent Elementary Features (BRIEF) descriptors [12]. These features are very fast to compute and match, while providing comparable performance to SURF [13], yet they only require the CPU. We note that GPU implementations of descriptors like SURF are available [14] and allow frame rate performance. However if we used a GPU implementation of such a descriptor (to achieve frame rate processing), the matching step in each localiser, and the live VO system, would require access to the GPU. Note that feature extraction on the incoming frame is independent and can be performed once at the start. As most systems only have one or two GPUs at best, and the number of active localisers and VO system is often larger than this, the dependancy on a GPU makes parallelisation difficult, compared to running a CPU-only program on a multi-core or multi-process system.

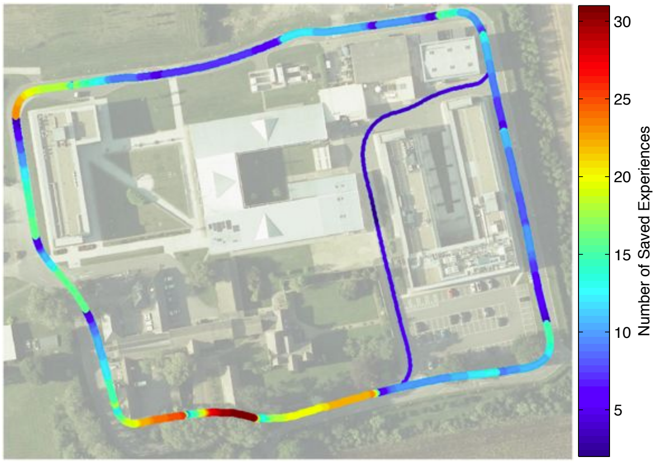


Fig. 7. Overhead of the two routes visited during data collection. The outer loop, indicated by the thicker line was driven 47 times, while the inner loop, shown by the thinner line was driven 6 times. The intensity of the trace shows how many experiences have been laid down at each point. See Fig. 6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

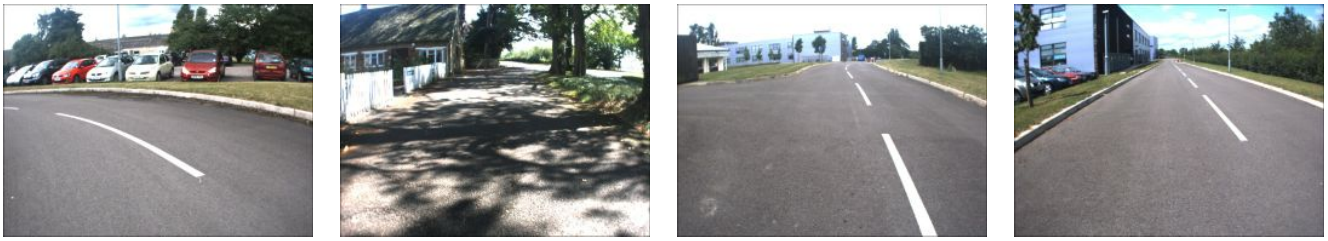
We use the FAST corner extractor [15] to compute points of interest for BRIEF. Matched landmarks are refined to sub-pixel precision using efficient second-order matching, described by Mei [16]. The 6 DoF transforms t_k computed by the VO system, when compared to the same relative transforms computed by the vehicle INS (which we assume is ground truth) have a mean error of $[-0.0093, -0.0041, -0.0420]$ meters and $[-0.0239, 0.0021, 0.0040]$ degrees and standard deviation of $[0.0225, 0.0245, 0.0155]$ meters and $[0.0918, 0.0400, 0.0383]$ degrees.

To determine if a localiser is successfully localising, i.e. $\mathbf{L}({}^{j}\mathcal{E}, \mathcal{F}_k) = 1$, we use two metrics. Firstly we require that the number of landmarks successfully matched from the previous experience to the live frame be at least 5% of the total number of landmarks searched for. Secondly we require that the translation from $\mathcal{F}_{k-1}$ to $\mathcal{F}_k$ computed by the localiser be within 15% of the same translation computed by the VO system running on the live stream.

V. Results

A. A Three Month Experiment

To test our framework we collected 53 traverses of two semi-overlapping 0.7km routes around Begbroke Science Park. Data was collected over a three month period at different times of day and with different weather conditions using the group's survey vehicle, the Wildcat. Fig. 7 shows an overhead of the site, along with the two routes driven. 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. The intensity of the plot indicates how many experiences have been laid down at each point. 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.



(a) (b) (c) (d)

Fig. 6. (a) and (b) are examples of places that have a large number of experiences associated with them. In (a) this is because the configuration of the car park changes daily, meaning past experiences are difficult to localise in. (b) suffers from strong shadowing effects due to overhanging trees and the road often has a large amount of fallen foliage on it. Both of these cases motivate the use of running the VO system continuously in parallel with the localisers. (c) and (d) are examples of places that have low visual variation and a relatively low number of saved experiences are needed to stay localised.

Some regions require more experiences than others. In Fig. 6 we show examples of places which exhibit both low and high visual variation. One example of a region that has high visual variation is the car park, Fig. 6(a). The contents and configuration of this space varies daily, so experiences stored on previous days are unlikely to be useful. Another is a section of road covered by overhanging trees, Fig. 6(b). Sunlight causes strong and intricate shadowing effects. While these are useful for the live VO system, they are often not encountered again meaning previous experiences are not useful. This encourages the use of the live VO system at all times.

Fig. 8 shows how much of each traverse is saved, along with the time of day it was captured. Here the visit numbers are in the order in which the data was collected. 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. Fig. 12 shows examples of localisation failures on visit 4, where strong shadowing effects are encountered for the first time. Fig. 13 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 (like driving at dusk or a different route), we find that as we revisit the route we typically need to remember less and less each time.

B. Are Multiple Experiences Needed?

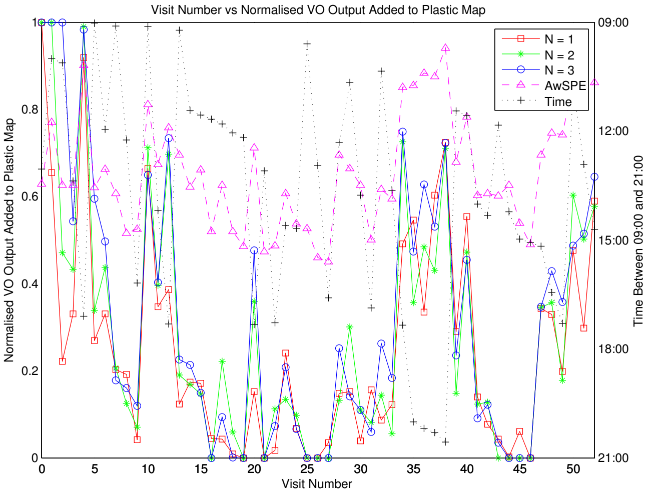


Fig. 8. For each traverse along the route, we show the fraction of the VO output that is saved for future use. Results shown for minimum number of localisers $N = 1$, $N = 2$ and $N = 3$. For the first 47 visits the outer loop was driven. On the final 6 traverses the inner loop was driven, resulting in the spike as a new area is saved. Note that the rest of the experience is relevant so only the new fraction is added. Visits 35-38 were done as dusk fell with light rain present. Both of these conditions had not been experienced before, resulting in large additions to the plastic map. See Fig. 13 for examples of Visit 38. For each visit we also computed the fraction of the saved VO output when every other visit was used as a single available experience and averaged the results. We refer to this as Average with Single Previous Experience (AwSPE).

To demonstrate the value of saving multiple experiences and building the plastic map, we evaluated what would happen if we made only one previous experience available. This represents the model most traditional localisation methods use, where a single previous view of the world is privileged and assumed correct (often the first experience of the world), and no attempt is made to capture the changing environment. For every visit we used every other visit as the only available saved experience. We calculated the amount of VO output that needed to be saved in each case, indicating localisation failure, and averaged the results. These are shown in Fig. 8, referred to as Average with Single Previous Experience (AwSPE). Across all visits, the average VO output saved is 65.8%, and no one visit is particularly “good” as a prior for other visits. This motivates the use of multiple experiences to represent the environment.

Over time, baring new routes, our system produces a $1/(TraverseCount)$ decay as the system captures the typical variation of the route. The order in which the data were collected is a privileged sequence. However it could be ordered in 53! ways to produce different performance graphs. To make the point we performed a greedy re-ordering. We moved the 10 most surprising traverses of the outer route and the 6 inner traverses to the beginning and re-ran the system. The resulting performance graph is shown in Fig. 10. Now we have moved the most “interesting” and “surprising” visits to the beginning of the plastic map creation we see that

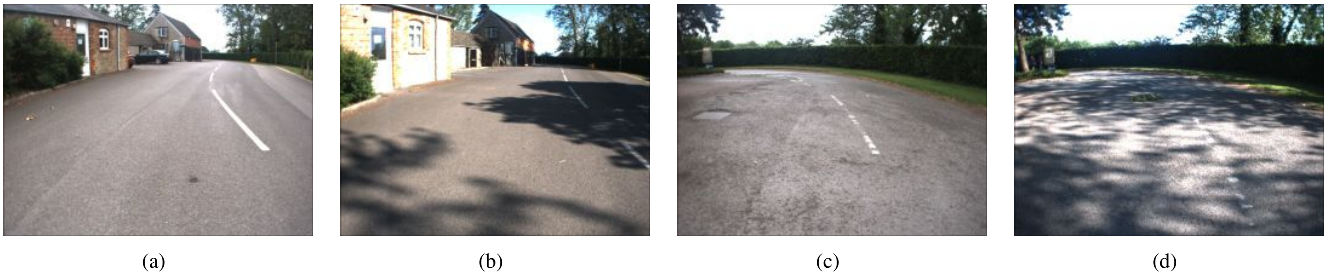


Fig. 12. Examples of localisation failure leading to the creation of new saved experiences on visit 4. Frames (a) and (c) are from previously saved experiences and frames (b) and (d) are the live stream.

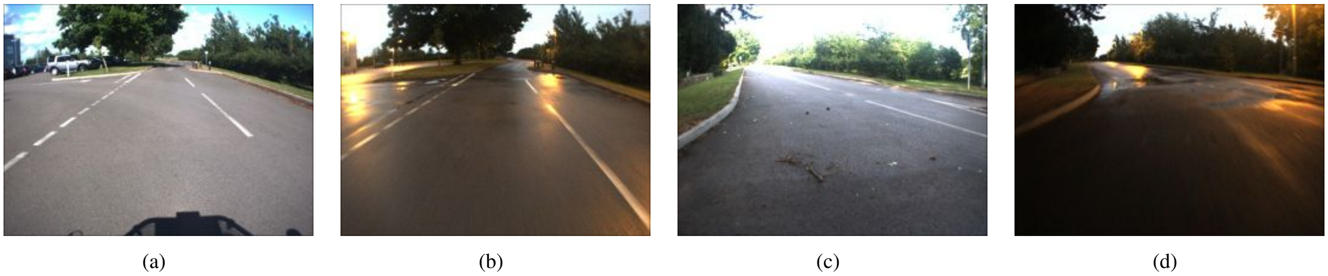


Fig. 13. Examples of localisation failure leading to the creation of new saved experiences on visit 38. Frames (a) and (c) are from previously saved experiences and frames (b) and (d) are the live stream.

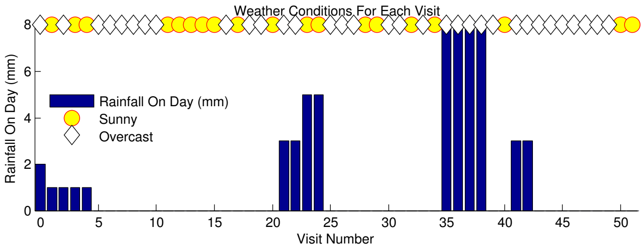


Fig. 9. Weather statistics for each traverse [17].

accumulation of experiences stays high until visit 18 before dropping off significantly.

We also classified each traverse as either overcast or sunny (the weather for each visit is shown in Fig. 9). We ran the system using only overcast or sunny experiences, the results of which are shown in Fig. 11. (We removed the 4 dusk and 6 inner loop traverses.) Interestingly with the overcast only visits, we quickly accumulate sufficient experiences to manage cloudy conditions, while the sunny traverses have a slight offset bias. We believe this to be caused by shadowing effects making localisation in previous experiences difficult.

Finally we show the performance of the system running on the Wildcat hardware in Fig. 14. The Wildcat houses 2 Intel Xeon X5570 2.93GHz CPUs, offering 16 cores (with hyper-threading). 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. At this point a new experience begins. 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 100ms, while the average for the successful localisation part of the sequence (i.e. not including frames 1296-2239) is 53ms. This is possible due to the ability to parallelise the localisation processing across the Wildcat CPU cores.

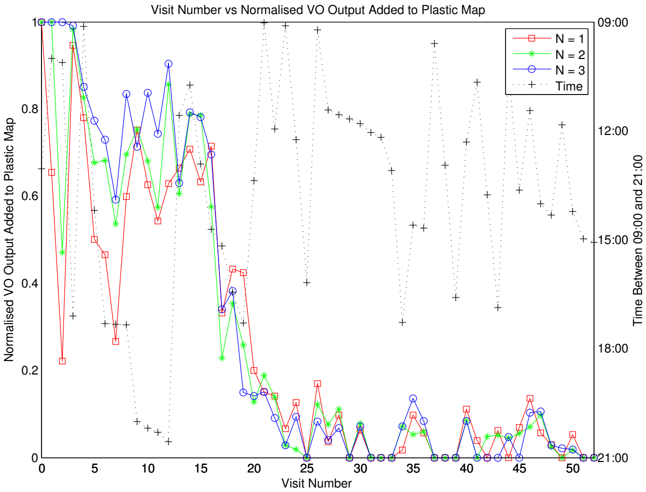


Fig. 10. The traverse order has been greedily re-arranged. The 10 worst performing visits and 6 inner loop visits from Fig. 8 were shifted to the beginning and the system re-run.

As noted previously, our approach has been implemented using a stereo camera, but would also be applicable to lidar sensors. It is likely that the higher number of experiences saved during sunny conditions, suggested by Fig. 11 and demonstrated in Fig. 6(b), could be moderated by using a lidar as it would be less affected by the changing lighting conditions. However 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 Fig. 6(a).

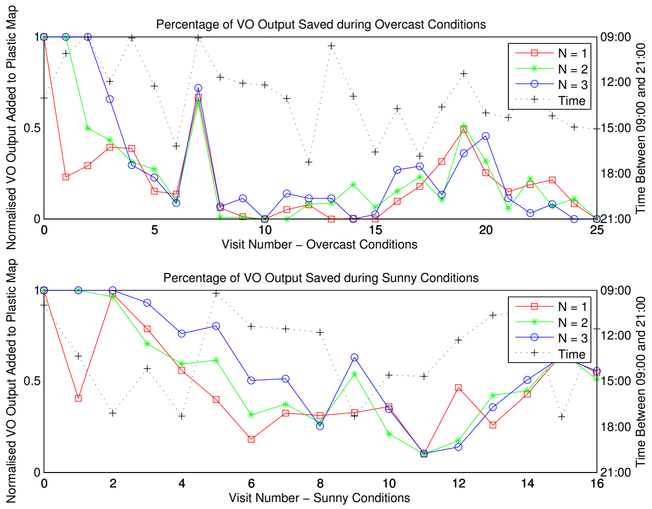


Fig. 11. Performances of only overcast (top) vs. only sunny (bottom) traverses. Notice the constant offset for sunny conditions compared to overcast conditions. This is probably caused by strong shadows making localisation difficult in previous experiences.

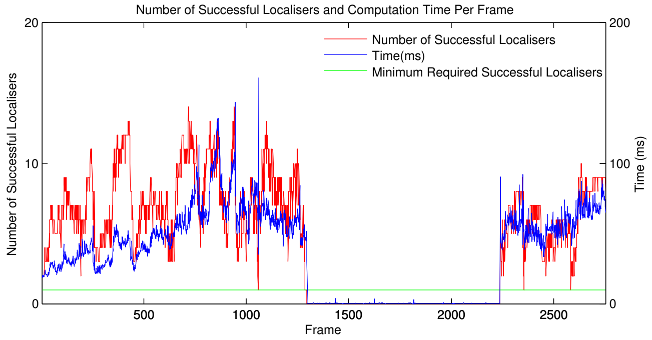


Fig. 14. Timing performance (using the Wildcat hardware) and number of successful localisers for each frame on visit 47, the first traverse of the inner loop. Only when the robot is driving on the previously unseen road is the VO saved, otherwise localisation is always successful. In the regions of previously seen road the average number of successful localisers is 7, and the average localisation time per frame during this period is 53ms.

VI. Conclusion

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, using the 3 months of data we possess at the time of writing, 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. We hope that practice will make perfect.

VII. Acknowledgements

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

The authors would like to thank Gabe Sibley and Chris Mei for discussions on RSLAM which lead to the development of the VO system used in this paper.

The authors would also like to thank the members of MRG who aided in the collection of the data used in this paper.

References

[1] G. Sibley, C. Mei, I. Reid, and P. Newman, "Adaptive relative bundle adjustment," *Robotics Science and Systems*, 2009.

[2] K. Konolige and J. Bowman, "Towards lifelong visual maps," in *IROS*, 2009, pp. 1156–1163.

[3] K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua, "View-based maps," *International Journal of Robotics Research (IJRR)*, vol. 29, no. 10, 2010.

[4] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard, "A Tree Parameterization for Efficiently Computing Maximum Likelihood Maps using Gradient Descent," *Robotics Science and Systems*, 2007.

[5] M. Milford and G. Wyeth, "Persistent navigation and mapping using a biologically inspired slam system," *The International Journal of Robotics Research*, 2009.

[6] P. Biber and T. Duckett, "Dynamic maps for long-term operation of mobile service robots," in *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

[7] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *Journal of Field Robotics*, vol. 27, no. 5, pp. 534–560, 2010.

[8] M. Bosse and R. Zlot, "Map Matching and Data Association for Large-Scale Two-dimensional Laser Scan-based SLAM," *International Journal of Robotics Research*, vol. 27, pp. 667–691, 2008.

[9] D. Nister, O. Naroditsky, and J. Bergen, "Visual Odometry for Ground Vehicle Applications," *Journal of Field Robotics*, vol. 23, 2006.

[10] G. Sibley, C. Mei, I. Reid, and P. Newman, "Vast scale outdoor navigation using adaptive relative bundle adjustment," in *International Journal of Robotics Research*, vol. 29, no. 8, July 2010, pp. 958–980.

[11] M. Cummins and P. Newman, "Highly Scalable Appearance-Only SLAM FAB-MAP 2.0," in *Robotics Science and Systems*, 2009.

[12] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," in *European Conference on Computer Vision*, September 2010.

[13] H. Bay, A. Ess, T. Tuytelaars, and L. V. Goo, "Surf: Speeded up robust features," *Computer Vision and Image Understanding (CVIU)*, vol. 110, pp. 346–359, 2008.

[14] N. Cornelis and L. V. Gool, "Fast Scale Invariant Feature Detection and Matching on Programmable Graphics Hardware," in *Computer Vision and Pattern Recognition*, 2008.

[15] E. Rosten, G. Reitmayr, and T. Drummond, "Real-time video annotations for augmented reality," in *Advances in Visual Computing. LNCS 3840*, December 2005, pp. 294–302.

[16] C. Mei, S. Benhimane, E. Malis, and P. Rives, "Efficient homography-based tracking and 3-d reconstruction for single-viewpoint sensors," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1352–1364, Dec. 2008.

[17] www.weatheronline.co.uk.