# Made to Measure: Bespoke Landmarks for 24-Hour, All-Weather Localisation with a Camera

Chris Linegar, Winston Churchill and Paul Newman

***Abstract*— This paper is about camera-only localisation in challenging outdoor environments, where changes in lighting, weather and season cause traditional localisation systems to fail. Conventional approaches to the localisation problem rely on point-features such as SIFT, SURF or BRIEF to associate landmark observations in the live image with landmarks stored in the map; however, these features are brittle to the severe appearance change routinely encountered in outdoor environments. In this paper, we propose an alternative to traditional point-features: we train place-specific linear SVM classifiers to recognise distinctive elements in the environment. The core contribution of this paper is an unsupervised mining algorithm which operates on a single mapping dataset to extract distinct elements from the environment for localisation.**

**We evaluate our system on 205 km of data collected from central Oxford over a period of six months in bright sun, night, rain, snow and at all times of the day. Our experiment consists of a comprehensive N-vs-N analysis on 22 laps of the approximately 10 km route in central Oxford. With our proposed system, the portion of the route where localisation fails is reduced by a factor of 6, from 33.3% to 5.5%.**

## I. Introduction

This paper addresses the problem of camera-only, metric localisation in difficult outdoor environments. In outdoor environments, the scene's appearance changes frequently, and often unpredictably, as a function of weather, lighting and season. Traditional approaches rely on point-features (such as SIFT, SURF and BRIEF) for metric localisation, however these point-features are not robust to severe appearance change. This paper presents an alternative to point-features. We propose an unsupervised mechanism to extract mid-level distinctive features from the environment, training place-specific linear SVM classifiers to fire on these distinctive features. These classifiers are used at run-time to associate image patches in the live and map images to perform robust localisation.

We emphasise that the problem of metric localisation is different to that of place recognition. SeqSLAM [1] and FAB-MAP [2] are examples of place recognition systems – they output the image (or place) to which the robot's live image is most similar, however there is no information about the robot's 6-DOF pose in the map.

We approach the problem of camera-based localisation from the context of our previous work in Experience-Based Navigation [3][4]. In this previous work, the robot incrementally built up a map of overlapping experiences, where each experience could be thought of as a distinct visual snapshot of the world under particular conditions (e.g. sunny, rainy, night, etc). At the core of this work was a localiser. The localiser subsystem took two images as input, and if possible, returned a 6-DOF transformation between the two images. In this previous work, a traditional point-feature localiser performed this function. However, because point-features are brittle to appearance change, the robot could only localise when the live image was visually similar to one of the experiences contained in the map. For example, if the map contained cloudy and rainy experiences, localisation would fail on the first time the robot encountered a sunny day. Therefore, the robot needed to survey the environment many times, under many different states of appearance, before achieving robust localisation. This presents a problem for systems in which a high level of autonomy must be reached with limited training data.

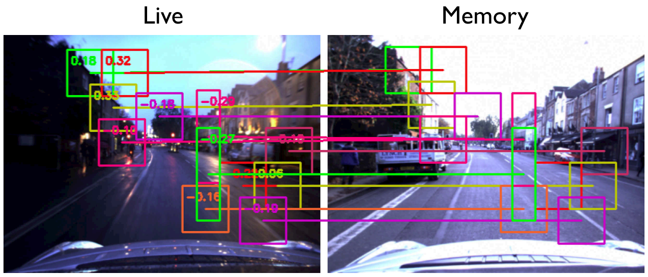


Fig. 1: We present an alternative to traditional point-feature localisation systems. Our system extracts distinctive elements from the environment using an unsupervised mining technique. We present a comprehensive N-vs-N analysis on 205 km of driving in central Oxford, showing that the proposed system is significantly more robust to appearance change than traditional approaches.

Our work is similar in spirit to McManus *et al.* [5], where they trained SVM classifiers to identify distinctive landmarks in a scene. However, our work differs in a number of key areas:

1) Our method trains classifiers on a single dataset, whereas [5] requires a high number of traverses through the environment.
2) We propose a new training algorithm based on inexpensive geometric tests, resulting in a significantly faster training algorithm. As a result, we can train much larger maps with fewer computational resources.
3) Our system does not require GPS to manually align datasets before training.

For the purposes of clarity in this paper, we set aside our previous work in experiences [4] and concentrate solely on the localiser subsystem. We present a comprehensive N-vs-N

Authors are from the Mobile Robotics Group, University of Oxford, Oxford, England; {chrisl, winston, pnewman}@robots.ox.ac.uk

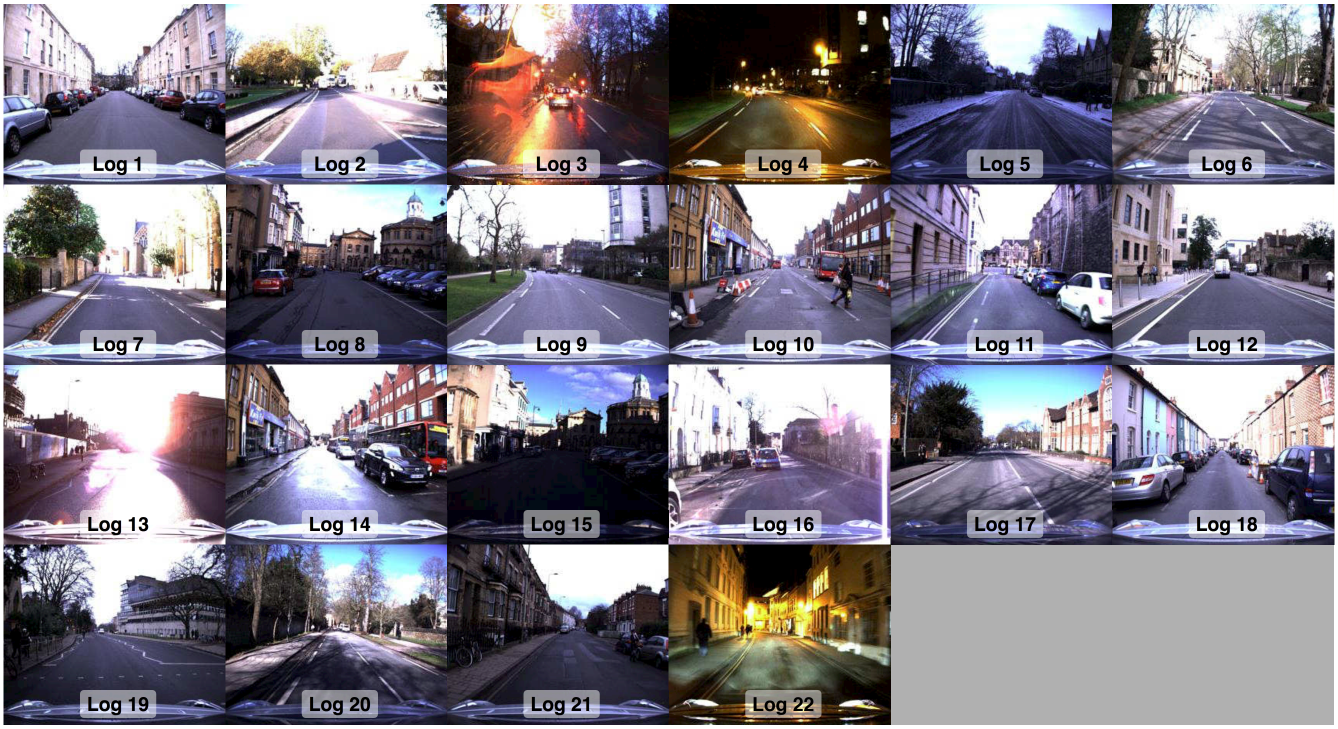


Fig. 2: Sample images from 205 km of data from the Oxford 10 km dataset. Each image is taken from one of the 22 datasets used in this paper, illustrating the challenging weather and lighting conditions present in the dataset.

comparison on 205 km of data from central Oxford, where we compare the performance of a traditional point-feature localiser to our proposed method. Sample images from each log are shown in Figure 2 and classified by weather conditions and time of day in Table I. Our experiments show that during the day, the portion of the route where localisation fails is reduced from 33.3% to 5.5%, a factor of 6 improvement. This remaining 5.5% would be resolved using an experience-based approach as in [4].

## II. Related Work

Outdoor localisation using vision has received significant attention in recent years. We make the distinction here between topological methods which produce localisations with respect to a collection of images, or places, versus the metric methods which report a numeric pose relative to the map. While the focus of this paper is metric localisation, we review recent works in both areas.

Many prior works in this area use local, low-level image features such as corners [6][7], blobs [8], or lines [9] as the underlying point of interest detector. These are then described with a local feature descriptor such as SIFT [10], SURF [11] or one of the binary descriptors [12][13][14][15]. Numerous SLAM approaches using combinations of these feature detectors and descriptors have been demonstrated [16][17][18]. Furgale and Barfoot used SIFT features in their visual Teach and Repeat system and noted their lack of robustness to changes in the time of day [19]. Valgren examined the effect of seasonal change on SIFT and SURF features for topological localisation, but did not examine metric localisation [20].

With these limitations of point-based methods for robust long-term localisation, recent approaches use mid-level or whole-image descriptors. SeqSLAM [1] and BRIEF-GIST [21] are two examples of topological localisers that use whole-image descriptors. SeqSLAM, by exploiting the sequence information, demonstrated day-to-night localisation, and noted that FAB-MAP [2] (a point-based feature system) performed poorly in such conditions. More recently Sunderhauf *et al.* used Edge Boxes to generate landmarks from each image [22][23]. Edge Boxes produce candidate edge-based bounding boxes for objects in the scene, which can be ranked with an "objectness" score. Combined with features from ConvNet [24], Sunderhauf *et al.* demonstrated a topological localiser robust to viewpoint and appearance change.

The SLAM++ approach developed by Salas-Moreno *et al.* used full objects in their framework rather than typical low-level features [25]. Objects such as chairs and tables were detected and tracked, however they required detailed 3D models of the objects used for localisation.

Our work is similar to that of McManus *et al.* who proposed Scene Signatures [5]. Rather than use corners or line-primitives for features, they used mid-level patches that were distinctive in their local setting. The localisation performance was metric and more robust than point-feature methods, but the metric accuracy was reduced, leading to the term "weak localiser". This was inspired by the work of Doersch *et al.* whose method distinguished between images of Paris and London by learning distinctive mid-level patches of each city [26]. A similar approach is also used by Li *et al.* who also use higher-level visual features for underwater place recognition [27].

| Log | Weather | Time of day | Log | Weather | Time of day |
|---|---|---|---|---|---|
| 1 | Cloud | Mid-day | 12 | Cloud | Morning |
| 2 | Sun | Mid-day | 13 | Sun | Morning |
| 3 | Rain | Dusk | 14 | Cloud | Morning |
| 4 | Clear | Night | 15 | Sun | Afternoon |
| 5 | Snow | Morning | 16 | Sun | Afternoon |
| 6 | Sun | Morning | 17 | Sun | Afternoon |
| 7 | Sun | Afternoon | 18 | Sun | Morning |
| 8 | Partly cloudy | Afternoon | 19 | Cloud | Morning |
| 9 | Partly cloudy | Afternoon | 20 | Sun | Morning |
| 10 | Cloudy | Morning | 21 | Cloud | Morning |
| 11 | Clear | Dusk | 22 | Night | Night |

TABLE I: Table categorising the 22 datasets used for evaluation by the weather conditions and time of day. Sample images are shown in Figure 2.

Our work differs from [5] in a number of ways. Most notably, we use only a single dataset to create our map, whereas McManus *et al.* require a number of training datasets. Additionally, our training procedure discovers distinctive elements through inexpensive geometric tests, which is significantly faster than the method in [26]. Lastly, we do not require GPS to manually align the datasets before training.

## III. Traditional approaches

Before discussing our method in more detail, we provide a brief overview of the core components of a traditional localisation system.

### A. *Visual odometry*

Visual odometry is used to provide an estimate of the robot's motion through the environment [28][29][30]. In our implementation, visual odometry consumes a stream of image pairs from a stereo camera. For each stereo frame, a set of 3D landmarks is extracted. The robot's trajectory is estimated by associating observations in the live stereo frame with landmarks stored in the previous frame and minimising the reprojection error of the observed landmarks.

### B. *Localisation and mapping*

The map is stored as a graph, where nodes in the graph contain images and extracted 3D landmarks, and edges contain the 6-DOF transformations obtained from visual odometry [4]. At run-time, localisation is performed between the live image and landmarks stored on a node in the graph. Landmarks are described by a feature descriptor (e.g. SIFT, SURF, BRIEF). These feature descriptors are used to perform data associations between observations in the live image and landmarks stored on a node. The pose optimisation is performed in a similar manner to [28][29][30].

We find that appearance change makes localisation fail in two ways. Firstly, *the feature extractor breaks down* – i.e. the extracted features in the live image are not the same as those in the map image. This often happens when shadows create strong gradients and corners in the image. Secondly, *the feature descriptors are not invariant* to the level of appearance change encountered in outdoor environments. This makes the data-association step fail. Our method addresses both of these failure modes.

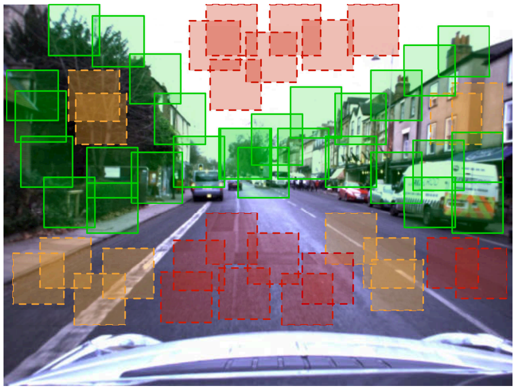


Fig. 3: Our unsupervised mining algorithm iteratively prunes and retrains a bank of robust detectors. The first phase trains a set of seed detectors (green, red and orange). By triangulating the landmark observations across a set of nearby images, we can reject landmarks which do not optimise to a consistent position (shown in red). A test for aliasing over a larger radius from the origin removes detectors which are not unique within the environment (shown in orange). The remaining detectors (shown in green) are saved.

## IV. Training method

The training method is an offline mining procedure which extracts distinctive elements from the environment and describes them so that they can be detected at run-time for localisation.

A distinctive element in the environment might be a postbox alongside the road, or the particular shape of a building's roof on the horizon. However, while it is helpful to think of these distinctive elements as belonging to semantically meaningful objects, we are not limited to them. Figure 1 illustrates a subset of the distinctive elements present in the scene.

We train a linear SVM classifier to detect each distinctive element. SVMs provide a powerful way to describe the appearance of landmarks since they can be trained using multiple observations of the landmark from different images. In this paper, we discover and train robust detectors using a single training dataset, with sufficient generality to localise across the extreme changes in appearance shown in Figure 2.

While this paper focuses on how to train classifiers using a single training dataset, the method naturally extends to learning from multiple passes through the environment – provided that the relative poses between images in datasets can be obtained. This would improve the robustness of the landmark detectors, however we only consider the case that a single mapping run is available, as in the Teach and Repeat systems used by Furgale and Barfoot [19].

### A. *Banks of place-specific detectors*

Landmarks in traditional point-feature localisation systems usually correspond to precise 3D points in the world – for example, the corner of a windowsill or signpost. However, the features we are interested in are more complex. For example, they may contain multiple overlapping objects with planes at varying depths in the image. In spite of this, we still require our landmarks to "behave" in the way a conventional 3D point-feature landmark would: as the robot moves through the world, the appearance of the landmark should project into the camera frame consistently.

The $i^{th}$ distinct landmark is referred to as:

$$\mathcal{L}_i = \{d_i, p_i\}$$

where $d_i$ is the linear SVM detector trained to detect the landmark, and $p_i$ is the homogeneous coordinates of the landmark relative to a chosen coordinate frame.

For a particular place $k$ in the map, a bank of place-specific landmarks $\mathcal{B}_k$ are trained, where:

$$\mathcal{B}_k = \{\mathcal{L}_0, \mathcal{L}_1, \mathcal{L}_2 ....\}$$

The bank of landmarks is stored on the node corresponding to place $k$.

### B. Mining distinctive landmarks

This paper presents a new technique for mining distinctive landmarks from the environment. The technique curates a bank of robust detectors by iteratively applying geometric consistency tests on the detectors. It is an unsupervised, camera-only technique that does not rely on GPS or manual alignment of training images.

The method below is for a single place $k$ and window size $s$. It is run in parallel for multiple places and window sizes. The training method is as follows:

1) ***Train seed detectors***. Select a single image $I_0$ from place $k$. Slide a window of size $s$ over image $I_0$. As the window moves, train detector $d_i$ for each new position of the window as in Section IV-D. We refer to these detectors as seed detectors and store them in $\mathcal{B}_k$. Many of these seed detectors will not represent distinctive elements – the following steps iteratively prune $\mathcal{B}_k$ until only the set of distinctive landmarks remain.
2) ***Test detectors in nearby images***. Query the graph structure for images close to image $I_0$ and store them in set $\mathcal{I}_1$. In our implementation, stereo images within a 1 m radius of image $I_0$ are included. Each detector in $\mathcal{B}_k$ is tested on the training images in $\mathcal{I}_1$, as in Section IV-E. We require a landmark to be visible in all images in $\mathcal{I}_1$, so we do not threshold the detection scores. This forces the detector to give us its "best estimate", yielding the vector of observations $\mathbf{z}$. The following step detects false positives.
3) ***Perform geometric tests for consistency***. Optimise for the 3D position of each detector using the observations $\mathbf{z}$ and the relative transformations $\mathbf{T}$ from visual odometry as in Section IV-F. If the detector has fired incorrectly, this will either prevent the optimisation from converging, or will result in outliers. False detections imply the underlying element in the environment is not unique, and so the landmark is rejected from $\mathcal{B}_k$. The red patches in Figure 3 illustrate the kinds of patches that typically fail this test – usually they correspond to featureless patches of road or sky. Only landmarks with good position estimates now remain in $\mathcal{B}_k$.
4) ***Retrain detectors from multiple images***. Since the observation of each landmark in each image in $\mathcal{I}_1$ is now known from the previous step, retrain the linear SVM classifiers in $\mathcal{B}_k$. In other words, we use the successful detections from the previous two steps to make the data associations between landmarks in different images. This makes the linear SVM detectors more robust. Every time the linear SVM classifiers are updated, the corresponding landmark positions are recalculated to ensure the detector still behaves as a consistent landmark.
5) ***Test for aliasing***. Consider a detector trained to fire on a lamp post. It may initially appear unique because training images have only been sampled from a small region in the map – however, as the robot moves outside of this radius, it may detect other lamp posts. This is a problem for localisation, as we want to avoid the data association problem of knowing which lamp post we have detected. The orange patches shown in Figure 3 are other examples of elements prone to aliasing.
   To this end, retrieve a set of test images $\mathcal{I}_2$ nearby the origin image $I_0$. In our implementation, images within a 5 m radius are used. Run the detectors in $\mathcal{B}_k$ on the images in $\mathcal{I}_2$ and note the detection locations. If the landmark is not visible in an image, the detector should return a low detection score. For this reason, we threshold on the maximum detection scores to allow for the event that the landmark is not visible. Since the relative transformations between images are available from visual odometry, and the 3D positions of landmarks are known, project each landmark in $\mathcal{B}_k$ into each image in $\mathcal{I}_2$. If a detector has fired incorrectly, remove it from $\mathcal{B}_k$.

The output of the training method is a bank of robust detectors for each place in the map which can be used for online localisation.

We include some additional implementation details below.

### C. ACF features

ACF features [31] are used as the underlying feature representation for input images. HOG features were used by McManus et al. [5], however we find that ACF features provide better performance.

### D. Training a detector

This section describes our procedure for training linear SVM classifiers. We define a function which accepts a vector of training images $\mathbf{I}$ and corresponding vector of landmark locations in the image $\mathbf{u}$. The function extracts image patches corresponding to the landmark locations in $\mathbf{u}$. These patches are labelled as members of the positive class. The function randomly samples the remainder of the image to populate the negative class. Liblinear [32] is used to train a linear SVM using the positive and negative classes.

To avoid overfitting, we artificially augment the training data [24]. In our implementation, the images in $\mathbf{I}$ are darkened, lightened and blurred.

### E. Using detectors to look for landmarks

Detections are performed using a sliding window approach. Since the training method requires landmarks to be unique in the scene, the detector returns only the maximum detection score and corresponding image coordinates – i.e. we do not allow multiple detections of a single landmark in the scene. If detection scores are unanimously low across the image (governed by threshold $t$), the landmark is declared not visible and no data association is made.

### F. Estimating landmark positions

An accurate localisation system relies on good estimates of landmark position. The position $p_i$ of a landmark $\mathcal{L}_i$ is modelled as a homogeneous coordinate vector:

$$p_i = (x, y, z, q)$$

where $q = 1$ if the landmark was close enough to the camera to observe its 3D position accurately, and $q = 0$ if the landmark was modelled at infinity. The robot's ability to estimate the depth of landmarks far away is a function of the baseline of the stereo camera. Observations of landmarks at infinity constrain the robot's orientation, but do not constrain its position.

We optimise for the homogeneous coordinate vector $p_i$ by triangulating observations of the landmark from multiple camera frames. Multiple observations of the same landmark may come from a stereo image pair and/or from multiple frames of a camera as the robot moves through the environment. The position of the landmark is solved for using Ceres [33], a non-linear least squares solver. If the optimisation converges successfully with no observations marked as outliers, and a low RMS reprojection error, we return the position of the homogeneous coordinate to $p_i$.

The current CPU implementation takes approximately 15 seconds to extract a bank of landmarks $\mathcal{B}_k$ on a 16-core 2.6GHz Intel Xeon machine. A GPU implementation would be significantly faster, since a significant portion of the processing work is in performing image convolutions. In our testing, we extract landmarks of size 64 x 64 and 32 x 128 using images of size 480 x 680. The training method typically extracts between 100 and 300 robust detectors per place.

## V. Localisation

The landmarks extracted in Section IV-B are used at runtime for localisation instead of traditional point-features such as SIFT, SURF and BRIEF. The localisation process is as follows:

1) ***Find the nearest bank of landmarks***. Query the graph structure to find the nearest bank of landmarks $\mathcal{B}_k$. Recall that the robot's pose in the graph is given by an external place recognition system such as FAB-MAP, SeqSLAM or GPS, or by using visual odometry to update the robot's previous pose estimate.
2) ***Run detectors on the live image***. Test each detector in $\mathcal{B}_k$ on the live image as described in Section IV-E. Since landmarks may genuinely not be visible in the image, we threshold detections by their detection score. This outputs a vector of observations $\mathbf{z}$ – the observations of the landmarks in the live image. This implicitly associates observations in the live image with landmarks in the map.

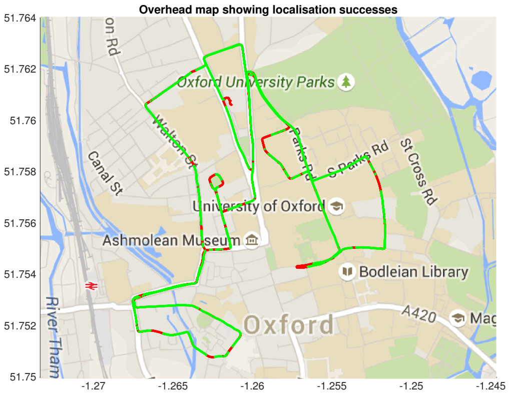


Fig. 4: The output from a single localisation run, where a sunny dataset was used to create a map and a cloudy dataset was used to localise in the map. The trajectory is plotted on an overhead map, and the success or failure of the localisation attempts are plotted in green and red, respectively.

| Datasets | Baseline | Proposed | Improvement |
|---|---|---|---|
| All datasets | 0.509 | 0.208 | 2.45 |
| Daytime datasets | 0.379 | 0.085 | 4.46 |
| Night datasets | 0.468 | 0.212 | 2.21 |
| Cloudy datasets | 0.297 | 0.018 | 16.6 |
| Sunny datasets | 0.469 | 0.148 | 3.17 |
| Map cloudy, localise at day | 0.333 | 0.055 | 6.05 |

TABLE II: Table with the average localisation performance from the N-vs-N experiment. This can be thought of as an average over the values in Figure 5, except that we exclude the diagonal $x = y$ where the same dataset is used for mapping and localisation.

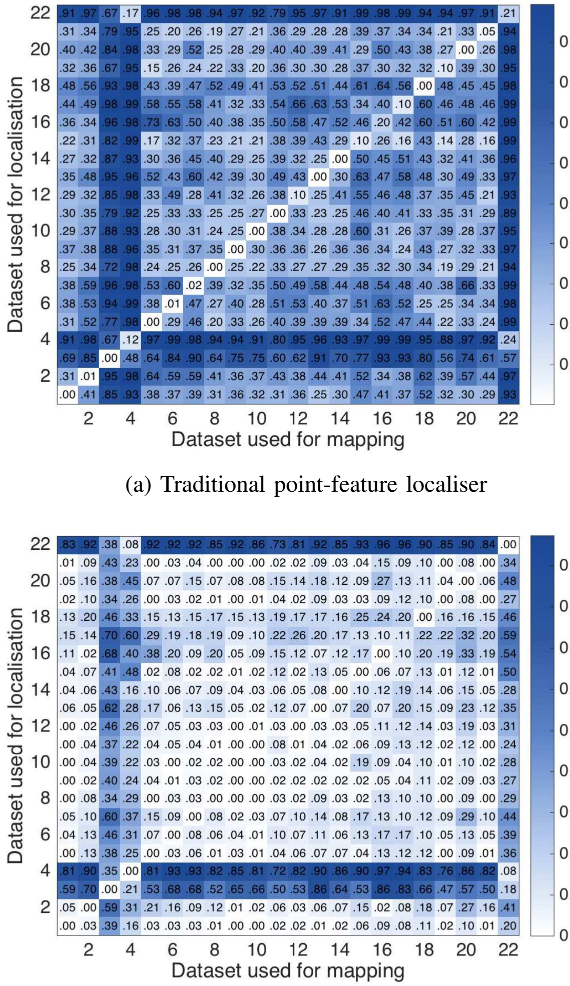
3) ***Do pose optimisation***. Optimise for pose in a similar manner to traditional point-feature localisers. We solve for pose with a 3D-2D pose optimisation [29][30] using Ceres [33].
4) ***Verify the pose estimate***. Verify that the pose estimate is reasonable by comparing successive localisation estimates with the ego-motion estimate from visual odometry, similar to [3][34][4][35]. This prevents poor localisation estimates propagating through the system.

## VI. Evaluation

We evaluate our system using the Oxford 10 km dataset. The dataset consists of 22 laps of an approximately 10 km route in central Oxford. Data is collected in bright sun, cloud, rain, snow and night, and at all times of the day. Figure 2 shows the extreme appearance change being considered and Table I categorises the datasets by weather and time of day.

### A. N-vs-N comparison

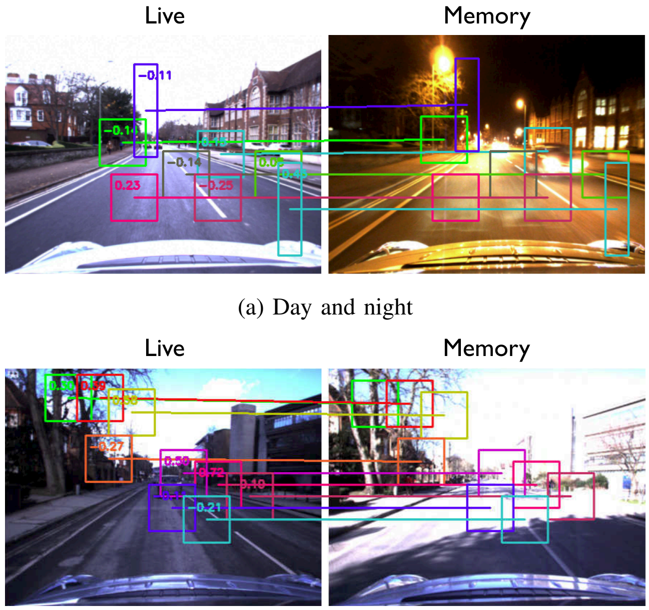
The primary experiment presented in this paper is a rigorous N-vs-N comparison which tests our system's ability



(a) Traditional point-feature localiser

(b) Proposed system

Fig. 5: Using 22 traverses of the Oxford 10 km route, we build 22 independent maps (listed along the x-axis). For each map, we attempt to localise all 22 traverses in that map (shown on the y-axis). The value in each cell is the portion of the localisation run in which localisation failed for longer than 20 m. We aim to minimise this distance travelled in open-loop, where the robot must estimate its pose in open loop using visual odometry. The figure shows our method consistently outperforming the baseline approach.



(a) Day and night

(b) Harsh lighting conditions

Fig. 6: Figures showing successful localisation in spite of severe appearance change. Only a subset of feature matches are plotted.

to localise across severe appearance change. We build 22 independent maps using each of the 22 traverses of the Oxford 10 km dataset using the training method in Section IV. For each map, we attempt to localise all of the 22 traverses against that map. This N-vs-N comparison presents a significant computational effort. To generate the results, we process the full 205 km of data in 22 different combinations, totalling 4510 km of processed data. Figure 4 shows the route on a map.

We measure localisation success by comparing the sequential localisation estimates with the estimates from visual odometry, in a similar manner to [3][34][35][4]. Our metric for success is the portion of the localisation run in which localisation fails for more than 20 m. This is an important metric, since during periods of localisation failure the robot must estimate its pose in open-loop using visual odometry which is prone to drift over large distances. Ground truth for the localisation runs is difficult to obtain since both localisers are more accurate than high-end INS systems.

Figure 5 presents the full results of this experiment. Figure 5a presents the performance of the baseline system, a traditional point-feature localiser, and Figure 5b shows the results of our proposed system. Each cell in the table corresponds to the portion of the route where localisation failed for more than 20 m, for a single mapping and localisation combination. For example, cell $[x, y] = [1, 8]$ corresponds to the localisation performance when Dataset 1 was used to create a map, and Dataset 8 was used to localise against that map. The diagonal $x = y$ corresponds to the event where the same dataset is used for mapping and localisation. The figure shows that localisation performance is consistently more robust using our proposed method.

Table II presents the mean performance of the two localisers. The mean is calculated by averaging the respective matrices from the N-vs-N experiment in Figure 5. Note that we exclude the diagonal elements from the calculation since these represent the case where the same dataset is used for mapping and localisation.

In Figure 5, it is clear that there is structure in the matrices. This implies that certain datasets performed consistently better (or worse) than others during localisation and mapping. We group the datasets in various combinations according to Table I and record the corresponding mean performance in Table II. We discuss these results in more detail below.

### B. *Localisation during the day*

This section analyses localisation performance during the day. Table II shows that when we exclude night datasets from the experiment, the portion of the route where localisation failure occurs is reduced from 37.9% to 8.5%, an improvement by a factor of 4.46. Figure 6b shows successful localisation under harsh lighting conditions.

However, our system does not only show improvement when severe appearance change is present. Table II shows the average localisation performance over a set of datasets captured in cloudy conditions. Cloudy datasets provide the most favourable conditions for the traditional point-feature localiser since they are the most visually similar (no shadows or direct sun). Despite this, the average portion of the route where localisation failed remains high at 29.7%. This is likely due to poor translation invariance, where lateral movement of the vehicle across the road causes localisation to fail, rather than appearance change. Here, our system outperforms the baseline system by a factor of 16.6, with only an average 1.8% of the route suffering localisation failure. As can be seen in 5b, a number of the localisation runs completed with zero localisation failures.

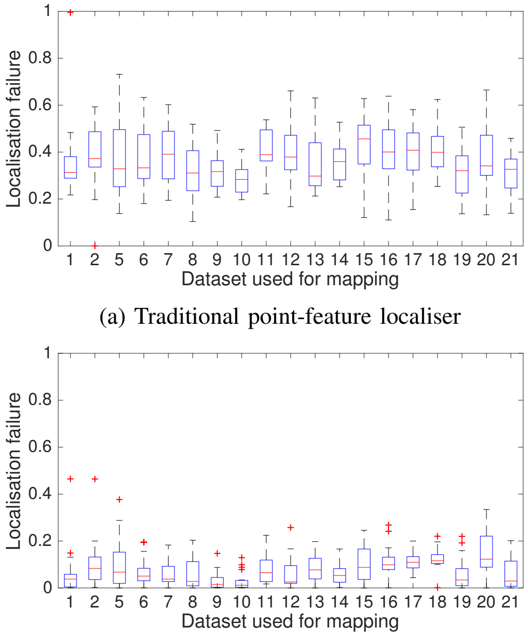
### C. *Localisation between night and day*

Consider the matrices in Figure 5. The matrix of the traditional localiser is roughly symmetric: mapping with Dataset A and localising with Dataset B results in similar performance to the converse of mapping with Dataset B and localising with Dataset A. In the case of localising between night and day, we see that localisation fails for a significant portion of the route regardless of whether the map is created during the day or night. Night datasets are Datasets 3, 4, and 22.

However, the matrix corresponding to the proposed localiser is not symmetric. When the map is created during the day, and localisation is performed at night, we see that localisation performs poorly in the same way as the baseline system. Interestingly however, mapping during the night and localising during the day results in a reduction in the percentage of localisation failure from approximately 90% to 40%. Figure 6a shows successful localisation between day and night. While localisation still fails for a large portion of the route, it is comparable with the best performance offered by the baseline system during the day. This means that to localise between night and day, the map should be created during the night. This may be because during the day there is more clutter in the scene, whereas at night only the most distinct landmarks are visible – nevertheless, it is certainly an interesting result.

### D. *Which datasets are best for mapping?*

Figure 7 plots the distribution of the localisation results on a per-map basis for datasets during the day. We assert that the average localisation performance on a given map is an indicator of underlying map quality. From Figure 7, cloudy datasets (Datasets 1, 9, 10, 19, and 21 in particular) appear to provide slightly better map quality than those created when sun or snow is present. This may be due to sun blinding the camera, or creating scenes with a high dynamic range which the sensor cannot capture. Table II shows the mean performance of creating a map with a cloudy dataset, and localising in that map using all of the daytime datasets (including the datasets with sun and snow). Under these conditions, the average localisation failure of our proposed system is reduced from 8.5% (when any daytime dataset is used for mapping) to 5.5% (when only cloudy datasets are used for mapping).



(a) Traditional point-feature localiser

(b) Proposed system

Fig. 7: Boxplot showing the distribution of the localisation results on a per-map basis for daytime datasets. The median result is marked with a horizontal red line. Better localisation performance is observed when the map is created under cloudy conditions (Datasets 1, 9, 10, 19, and 21).

### E. *Experience-Based Navigation*

In previous work [4], we leveraged the notion of "experiences" to perform robust localisation across appearance change. We used a traditional point-feature localiser which was brittle to appearance change, meaning that the system required a high number of experiences to map the environment. Additionally, the system could not generalise to unseen experiences – for example, if the map only contained sunny experiences, localisation would fail on the first encounter with snow.

We maintain that the framework of experiences is beneficial even with a more robust localiser. Figure 4 shows an overhead plot of the 10 km route, marking points along the trajectory where localisation failed and succeeded, in red and green respectively. A point of concern with the proposed system may be that there are certain parts of the world where it is simply not possible to extract distinctive landmarks, leaving "dead zones" in the map. Rather, we observe that given multiple maps and a single dataset for localisation, that localisation failures occur in *different areas* of the map.

This means that an approach using multiple experiences [4] would likely result in improved localisation performance.

## VII. Conclusion

This work has demonstrated an unsupervised mechanism for extracting distinctive mid-level features from the environment. We show that by applying inexpensive geometric checks for consistency in landmark position, we are able to extract the most distinctive elements from a scene. We describe these elements using linear SVM detectors, trained using a single mapping dataset. We leverage these distinctive landmarks at run-time to perform robust, metric localisation across extreme appearance change. The system is evaluated in an exhaustive N-vs-N comparison across 22 traverses of an approximately 10 km route, totalling 205 km of driving in central Oxford. We show that localisation failures during the day are reduced by a factor of 6, from 33.3% to 5.5%, when compared with a traditional point-feature localiser.

## VIII. Acknowledgements

The authors wish to acknowledge the following funding sources. Chris Linegar is funded by the Rhodes Trust. Both Paul Newman and Winston Churchill are supported by EPSRC Leadership Fellowship Grant EP/J012017/1. Additionally, the authors acknowledge the support of this work by the European Community's Seventh Framework Programme under grant agreement FP7-610603 (EUROPA2).

## References

[1] M. Milford and G. Wyeth, "SeqSLAM : visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE International Conferece on Robotics and Automation (ICRA 2012)*. IEEE, 2012, pp. 1643–1649.

[2] M. Cummins and P. Newman, "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance," *The International Journal of Robotics Research*, vol. 27, 2008.

[3] W. Churchill and P. Newman, "Experience-based navigation for long-term localisation," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1645–1661, 2013.

[4] C. Linegar, W. Churchill, and P. Newman, "Work Smart, Not Hard: Recalling Relevant Experiences for Vast-Scale but Time-Constrained Localisation," in *Proc. IEEE International Conference on Robotics and Automation (ICRA2015)*, 2015.

[5] C. McManus, B. Upcroft, and P. Newman, "Scene signatures: Localised and point-less features for localisation," in *Proceedings of Robotics Science and Systems (RSS)*, July 2014.

[6] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proceedings of IEEE European Conference on Computer Vision (ECCV)*, May 2006.

[7] C. Harris and M. Stephens, "A Combined Corner and Edge Detection," in *Proceedings of The Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[8] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2002, pp. 36.1–36.10.

[9] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 6, pp. 679–698, Nov 1986.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," in *International Journal of Computer Vision*, 2004.

[11] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features," *Computer Vision and Image Understanding (CVIU)*, vol. 110, pp. 346–359, 2008.

[12] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "Brief: Computing a local binary descriptor very fast," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1281–1298, 2012.

[13] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, November 2011, pp. 2564 –2571.

[14] S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints," in *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, November 2011, pp. 2548 –2555.

[15] R. Ortiz, "Freak: Fast retina keypoint," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ser. CVPR '12. IEEE Computer Society, 2012, pp. 510–517.

[16] 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.

[17] K. Konolige, J. Bowman, J. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua, "View-based maps," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 941–957, 2010.

[18] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, November 2007.

[19] 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.

[20] C. Valgren and A. J. Lilienthal, "Sift, surf and seasons: Long-term outdoor localization using local features." in *EMCR*, 2007.

[21] N. Sunderhauf and P. Protzel, "Brief-gist - closing the loop by simple means," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, Sept 2011, pp. 1234–1241.

[22] N. Suenderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, "Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free," in *Proceedings of Robotics: Science and Systems*, July 2015.

[23] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *ECCV*. European Conference on Computer Vision, September 2014.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012, pp. 1097–1105.

[25] H. S. P. H. J. K. Renato F. Salas-Moreno, Richard A. Newcombe and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," *IEEE Proc. Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[26] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. A. Efros, "What makes paris look like paris?" *ACM Transactions on Graphics (SIGGRAPH)*, vol. 31, no. 4, pp. 101:1–101:9.

[27] J. Li, R. M. Eustice, and M. Johnson-Roberson, "High-level visual features for underwater place recognition," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 2015, pp. 3652–3659.

[28] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007.

[29] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2006.

[30] D. Scaramuzza and F. Fraundorfer, "Visual Odometry: Part I - The First 30 Years and Fundamentals," *IEEE Robotics Automation Magazine*, 2011.

[31] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *PAMI*, 2014.

[32] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[33] S. Agarwal and K. Mierle, "Ceres solver: Tutorial & reference," *Google Inc*, vol. 4, 2012.

[34] C. McManus, W. Churchill, W. Maddern, A. D. Stewart, and P. Newman, "Shady dealings: Robust, long-term visual localisation using illumination invariance," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014.

[35] C. McManus, B. Upcroft, and P. Newman, "Learning place-dependant features for long-term vision-based localisation," *Autonomous Robots, Special issue on Robotics Science and Systems 2014*, pp. 1–25, 2015.