From Dusk till Dawn: Localisation at Night using Artificial Light Sources

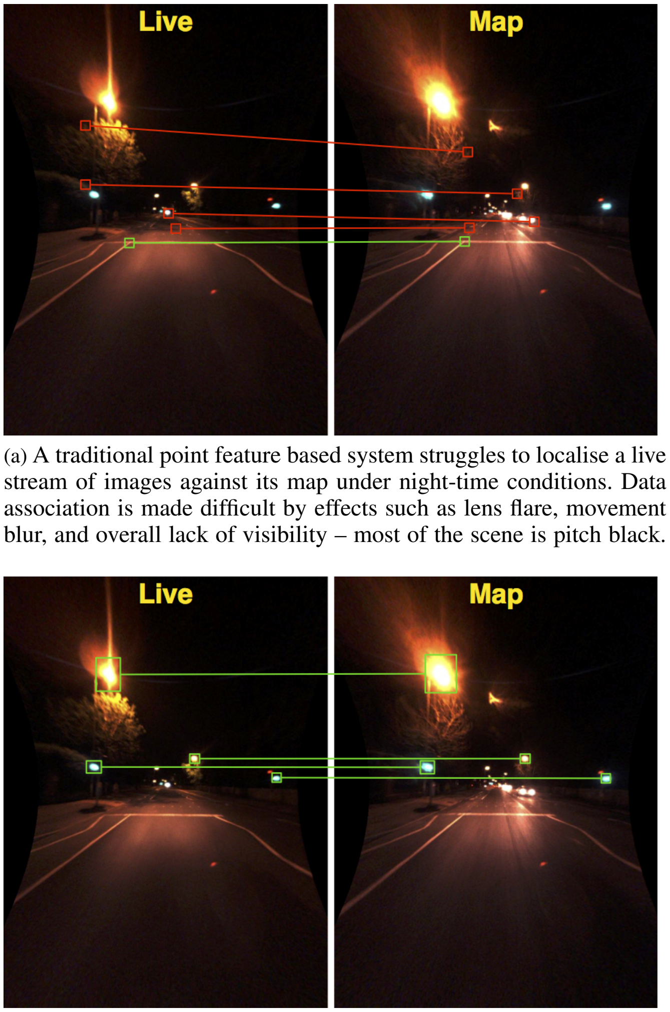
Peter Nelson, Winston Churchill, Ingmar Posner and Paul Newman

***Abstract*— This paper is about localising at night in urban environments using vision. Despite it being dark exactly half of the time, surprisingly little attention has been given to this problem. A defining aspect of night-time urban scenes is the presence and effect of artificial lighting – be that in the form of street or interior lighting through windows. By building a model of the environment which includes a representation of the spatial location of every light source, localisation becomes possible using monocular cameras. One of the challenges we face is the gross change in light appearance as a function of distance due to flare, saturation and bleeding – city lights certainly do not appear as point features. To overcome this, we model the appearance of each light as a function of vehicle location, using this to inform our data-association decisions and to regularise the cost function which is used to infer vehicle pose. In this way we develop a place-dependent but stable sensor model which is customised for the particular environment in which we are operating. We demonstrate that our system is able to localise successfully at night over 12 km in situations where a traditional point feature based system fails.**

I. Introduction

An ongoing challenge for autonomous vehicles is the problem of localisation: knowing where the vehicle is positioned relative to a map of its environment. While significant progress has been made in this area, the vast majority of research to date has only tackled the problem in daylight (or under similar lighting conditions). This is only half of the problem.

Unfortunately, conventional cameras typically perform poorly at night, often lacking the dynamic range required to capture scenes which can exhibit large luminance ranges when bright lights are present in otherwise total darkness. At this point we have a dilemma: either increase camera exposure time (and/or sensitivity) in order to better resolve poorly lit background, or decrease exposure time in order to reduce saturation and bleeding caused by light sources. Both of these compromises can increase noise and decrease information in an image, and the best strategy depends on the task at hand. For example, [1] uses the former approach to localise along dark rural roads while [2] uses the latter to detect car rear lamps. These problems can be avoided by using cameras that are designed specifically for use at night, however these are more expensive than consumer cameras and often have limited use in daylight. As one of our aims is to achieve cost effective autonomy using vision, we eschew these alternatives in favour of low-cost conventional cameras.



(a) A traditional point feature based system struggles to localise a live stream of images against its map under night-time conditions. Data association is made difficult by effects such as lens flare, movement blur, and overall lack of visibility – most of the scene is pitch black.

(b) By specifically detecting and matching lights in the live scene with those in a map, we are able to successfully localise. In addition to position in image space, we also take into account the expected appearance of each light, based on how far away it is. This greatly improves the robustness of our data association and serves to better inform our pose estimate.

Fig. 1. A visual comparison of how localisation at night fares using transient point features versus permanent, static light sources.

A consequence of increased noise in images is that low-level point features—widely used in visual mapping and pose estimation systems—are not as effective as they would be under optimal lighting conditions. Fortunately, many urban areas are brightly illuminated at night by artificial light sources of all kinds – street lights, traffic lights, road signs, billboards, and windows. In images taken with a conventional camera using typical exposure and shutter settings, these appear as large saturated blobs and hence are very easy

Authors are from the Mobile Robotics Group, Dept. Engineering Science, University of Oxford, UK. {peterdn, winston, hip, pnewman}@robots.ox.ac.uk

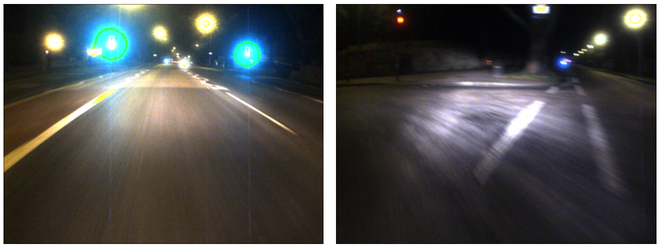


Fig. 2. Examples of scenes where point feature based methods fail. This is due to significant motion blur, saturation, lens flaring, and lack of well defined edges. In our approach we explicitly take advantage of the clearly visible light sources.

to detect and track. Most importantly these lights are often permanent, static, and usually always visible from the road. Therefore, they make suitable candidate landmarks of which a localisation system can take advantage. We must also however contend with dynamic light sources, for example headlamps of oncoming vehicles, and reflections.

In this paper we present a mapping and localisation system that: (i) automatically builds a map of artificial light sources from an offline sequence of images; and (ii) detects and matches observations of lights in an online sequence in order to localise within this map. Additionally, we take advantage of how the appearance of bright lights can radically change as distance from them increases or decreases. By remembering and modelling how a light's appearance changes as a function of distance during the map building process, we can refer back to these saved appearances in order to improve data association and better inform our pose estimate. We demonstrate that our system is able to localise under conditions where a point feature based teach-and-repeat system fails.

II. Related Work

Visual landmark-based mapping and pose estimation has been an active topic of research for decades. Many previous works use interest-point feature detectors (such as Harris Corners [3] or FAST [4]) to find small, distinctive image patches as landmarks for vehicle localisation and visual odometry (for example, [5], [6], [7], [8]). However, as these patches are often on the order of 10×10 pixels in size, they are highly sensitive to noise and image variability such as drastic illumination changes. In addition, many rely on sharp images in order to discern features such as corners and therefore are particularly affected by motion blur. Low-light conditions often exacerbate these effects for all but the most expensive and specialised cameras, and therefore systems that rely on these point features typically do not perform well in darkness. Figure 2 shows some examples of the conditions we must contend with at night.

One such previous work is a visual teach-and-repeat system that uses small point features as landmarks [9]. It works in a similar way to our system: a map of landmarks is built offline and localised against in a subsequent traversal. At each frame, the reprojection error of detected landmarks is minimised in order to correct a visual odometry frame-to-frame transformation estimate. We compare our new approach to our existing point feature based teach-and-repeat system in later sections.

In [10], multiple images captured at different exposures (exposure bracketing) are fused together in order to achieve a high dynamic range (HDR) effect. This allows a greater range of illumination to be captured but proves to be difficult when the camera moves during a capture sequence. They use the technique to improve robustness of SIFT key point detection for localisation under varying illumination conditions, however it is unknown how their system performs at night.

Our approach of using light blobs as landmarks has similarities to star tracking which is used in aerospace applications to determine a spacecraft's orientation. Modern star trackers are fully autonomous, have comprehensive star catalogs, and are highly accurate and robust [11]. Due to the extreme distances involved, star tracking is only able to determine orientation, not position. In contrast, most light sources visible from a vehicle navigating a well-lit urban area will be sufficiently close to give an indication of relative position. Moreover, we face very different environmental challenges navigating on Earth than in interplanetary space: landmarks can move, can be obstructed (possibly by other landmarks), and can disappear entirely without warning.

One of the few examples of work that addresses the problem of navigating a vehicle at night using consumer cameras is SeqSLAM [12], in which localisation is achieved by recognising coherent sequences of heavily down-sampled images. Their results showed they could recognise a brightly lit urban environment at night despite having previously seen it only during the day. In a subsequent work they were able to successfully localise even on unlit rural roads by maximising camera exposure and gain settings [1]. While effective, these techniques only provide an estimate of topological position: in order to control an autonomous vehicle we require a metric pose estimate. In addition, they require long sequences of images to be processed before a position can be determined.

III. Prerequisites

Here we give a brief overview of prerequisite techniques used in the following sections.

The problems of constructing a 3D map from a 2D image sequence, and estimating pose from a set of 3D-to-2D point correspondences are known as structure from motion (SfM) and perspective-n-point (PnP), respectively. In both cases our approach is based on minimising the reprojection error given detection of landmarks in 2D images. Detailed explanations of these problems can be found in [13].

Our optimisation approach is to use a slight variation of Levenberg Marquardt with a robust cost function (Huber kernel [14]). To greatly simplify expressions involving rotation matrices we use the linearisation technique described in [15]. In the case of pose estimation we introduce two other terms: (1) a weighted prior (on pose) by adding a regularisation term that penalises solutions that are far from the prior; and (2) distance constraints that take into account the estimated distance of the camera from each observed light. Hence our objective function $C(x)$, given n measurements, is:

$$C(\mathbf{x}) = \frac{1}{2}\left\|\Sigma_r^{\frac{1}{2}}\mathbf{r}(\mathbf{x})\right\|^2 + \frac{1}{2}\left\|\Sigma_s^{\frac{1}{2}}\mathbf{s}(\mathbf{x})\right\|^2 + \frac{1}{2}\left\|W\left(\mathbf{x}-\hat{\mathbf{x}}\right)\right\|^2 \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^6$ is the 6 degree-of-freedom (DoF) transformation we are solving for, parameterised by translational components x, y, z, and rotational components r, p, q. W is a 6×6 diagonal matrix that weights the contribution of the prior $\hat{\mathbf{x}}$. $\mathbf{r}(\mathbf{x}) : \mathbb{R}^6 \to \mathbb{R}^{2n}$ is a vector of reprojection error residuals and $\mathbf{s}(\mathbf{x}) : \mathbb{R}^6 \to \mathbb{R}^n$ is a vector of distance constraint error residuals. Σ_r and Σ_s are diagonal matrices containing the Huber weights for each measurement ($\frac{1}{2}$ denotes element-wise square root). Therefore our normal equations [16] for the least-squares minimisation are:

$$J^T \Sigma J \mathbf{x} = -J^T \Sigma \mathbf{y} \tag{2}$$

where J is the $(3n+6) \times 6$ Jacobian:

$$J = \begin{bmatrix} \frac{\delta \mathbf{r}}{\delta \mathbf{x}} \\ \frac{\delta \mathbf{s}}{\delta \mathbf{x}} \\ W \end{bmatrix} \tag{3}$$

$\mathbf{y}$ is the $(3n+6) \times 1$ vector:

$$\mathbf{y} = \begin{bmatrix} \mathbf{r}(\mathbf{x}) \\ \mathbf{s}(\mathbf{x}) \\ (\mathbf{x}-\hat{\mathbf{x}}) \end{bmatrix} \tag{4}$$

and $\Sigma = \mathrm{diag}(\Sigma_r, \Sigma_s, W)$.

The following sections describe the two main tasks our system is concerned with: (i) creating a map of light sources; and (ii) localising within a map of light sources. In both cases we process 2D image sequences from a number of appropriately calibrated monocular cameras for which we assume the standard pinhole camera model. We therefore also assume that as a preprocessing step the images are corrected to account for distortion.

IV. Mapping

A. Overview

Building a map of suitable light sources is fundamentally a structure from motion problem. Given a set of observations of landmarks from a camera stream and a corresponding set of 6 DoF vehicle poses, our main aims are twofold. Firstly, we need to estimate a 3D location for each static light, ideally discarding those that move – car head lamps, for example. Secondly, we wish to create an individual appearance model for every light – what it looks like and how its appearance changes as we move relative to it.

We describe the map creation process for a sequence of K frames $\mathcal{I} = \{\mathcal{I}_1, \ldots, \mathcal{I}_K\}$ from a single camera. From this point on we use the term *blob* to mean a 4-connected region of high intensity pixels. An overview of the steps for building a map are as follows:

1) Detect individual lights in each frame.
2) Track blobs between frames to obtain a set of light tracks.

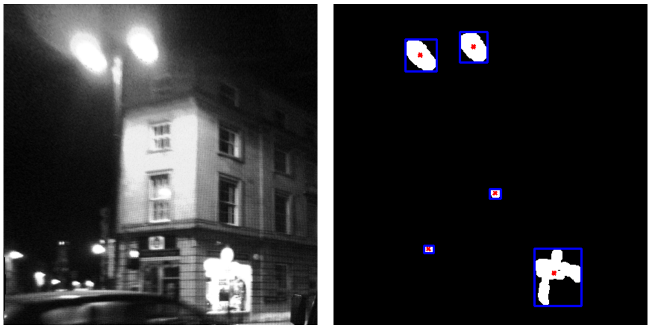

(a) (b)

Fig. 3. Light detection on a single frame. Figure (a) shows the raw image converted to greyscale. Figure (b) shows the binary thresholded image after erosion with bounding boxes and blob centroids.

3) For each light track, solve the structure from motion problem for every consecutive pair of observations to obtain a set of candidate 3D locations. Tracks that exhibit a large variance of candidate locations are discarded.
4) For the remaining light tracks, solve the structure from motion problem using all observations to obtain a final 3D location.
5) Extract and save appearance patches.

B. Light Detection and Tracking

For light detection we segment each image $\mathcal{I}_k \in \mathcal{I}$ using a fixed intensity threshold function to obtain binary image $\mathcal{I}_k^B$:

$$\mathcal{I}_k^B(u,v) = \begin{cases} 1 & \text{if } \mathcal{I}_k(u,v) > \lambda_{\mathrm{B}} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

Each $\mathcal{I}_k^B$ is eroded to remove noise and lights that are too dim or small to reliably observe. The blobs in each frame are then extracted from the eroded binary image and their bounding boxes and centroids are computed. This process is illustrated in Figure 3.

We track lights across multiple frames using a nearest-neighbour approach. A cost function that considers bounding box overlap and distance between blob centroids is used to find the best match for a blob in the previous frame. This works well for sparse lights that can be disambiguated very easily (for example solitary street lights) and less well for clusters that exhibit similar movement patterns within the image (for example groups of windows). We note however that this is not an issue as the latter are likely to also be ambiguous during localisation and make data association more difficult. In any case, lights that are tracked incorrectly in image space should not converge to a consistent location in 3D space and so will be culled in the next step.

Lights are tracked until they go out of view; occlusions, splittings, and mergings are not explicitly considered as these may indicate an unreliable light source. We therefore obtain

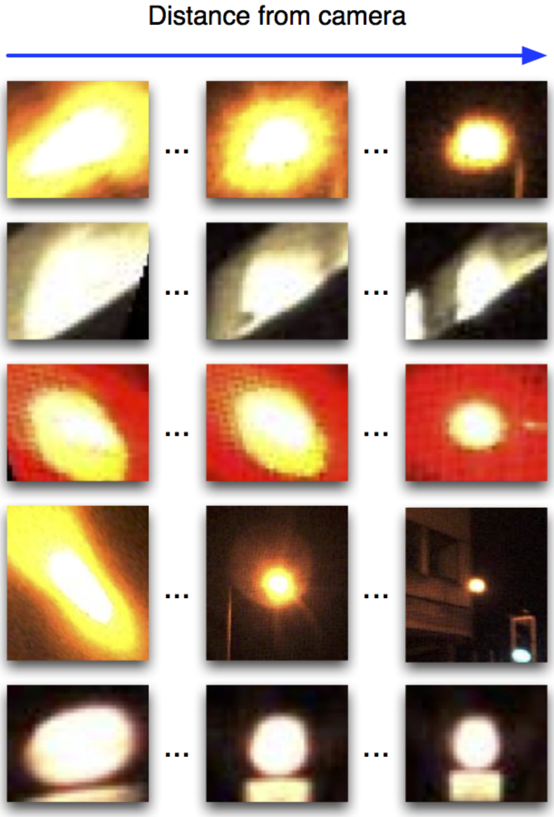


Fig. 4. Example appearances of lights at varying distances from the camera. Note how drastically their appearance changes as the camera moves towards them – particularly those exaggerated by flaring effects. In a point feature based system this would be a problem, however it allows our system to distinguish between different lights at varying distances with ease.

a set of n light tracks $\mathcal{T} = \{\mathcal{T}_1, \ldots, \mathcal{T}_n\}$. A track $\mathcal{T}_i = \langle \mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_{n_i} \rangle$ is an ordered sequence consisting of n_i blobs from consecutive frames.

C. Structure from Motion

Light tracks that contain fewer than λ_M observations are discarded immediately. For each blob track $\mathcal{T}_i$, we generate $n_i - 1$ estimates of the 3D position of the light using each consecutive pair of blobs. If the variance of these candidate locations is greater than a threshold λ_T, the blob track is discarded as this may indicate a moving light source or an otherwise incorrectly tracked object.

Surviving blob tracks are input to a final optimisation process to obtain a 3D position by minimising the reprojection error over all measurements. From this we obtain a set of N lights $\mathcal{L} = \{\mathcal{L}_1, \ldots, \mathcal{L}_N\}$ where each $\mathcal{L}_i \in \mathbb{R}^3$.

D. Modelling Appearance

For every observation in each light track, we extract and save a full-colour image patch from its corresponding frame, centred on the blob, resulting in a 'stream' of (appearance, camera pose) pairs for that light. Some examples of these appearance patches captured at different distances are shown in Figure 4. Additionally, the appearance stream is used to establish a 2-parameter functional relationship between the blob area and distance from the camera. For light $\mathcal{L}_i$, we posit that area A is related to distance d by:

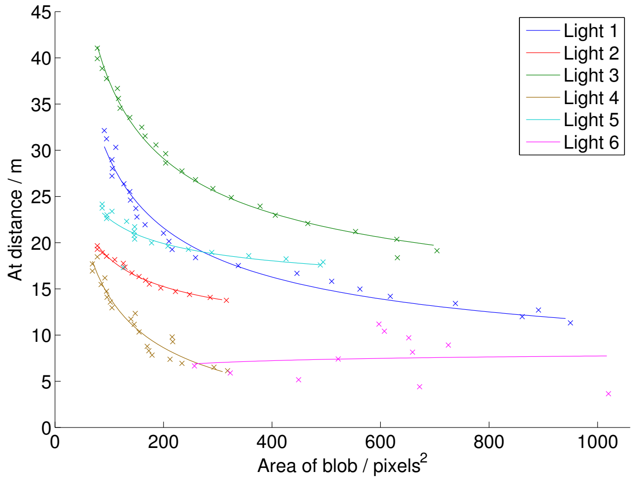


Fig. 5. Graphs of light blob area vs distance from camera for six different lights fitted to Equation 6. This illustrates in particular how by referring back to these models next time we see the corresponding lights, we can accurately infer how far we are from them. It also shows, along with Figure 4, just how diverse they look. This is very much to our advantage during data association. It is not perfect, however: light 6 is given as an example that *does not* fit the model correctly. This is mainly true for planar and irregular light sources, as well as those that are tracked incorrectly.

$$d = \frac{a_i}{\sqrt{A}} + b_i \tag{6}$$

as the inverse square law $A \propto \frac{1}{d^2}$ holds for the pinhole camera model. Values for the two parameters a_i and b_i are obtained by solving linear least squares. Figure 5 shows some examples of how light area relates to distance from the camera and the resulting graph fitted to Equation 6.

Both the appearance stream and functional relationship given by Equation 6 form the basis of our individual appearance models for each light.

V. Localisation

A. Overview

Given a set of observations of lights from a camera stream and a map of known light sources $\mathcal{L}$ we wish to estimate the camera pose. This is known as pose estimation or the perspective from n points problem. Although each camera has its own independent map, we are able to combine 2D-to-3D point correspondences from all cameras in order to solve for pose. It is assumed that all necessary camera calibration parameters are known in advance.

One of the key ideas that should be evident from Figure 4 is that not only do lights look very different from one another, but their individual appearances also change drastically as we move closer or farther away from them. We therefore firstly aim to use each light's individual appearance model to make our data association more robust. Secondly, Figure 5 shows us that we can infer valuable information about how far away lights are by their observed area. This relationship allows us to derive distance constraints from these appearance models that can influence our pose optimisation cost function.

An overview of the steps for localising with a new set of camera frames are as follows:

1) Predict the current vehicle pose using the previous pose and some estimate of incremental motion.
2) Detect lights in the current frame.
3) Using the current pose estimate, predict where lights in the map should appear in the current frame. With these predictions and the detected light blobs, use the joint compatibility branch-and-bound algorithm to compute a putative but mutually compatible set of 2D-to-3D point correspondences.
4) Perform a second pass rejection step based on light appearance, taking advantage of the fact that we know what these lights should look like.
5) Estimate distance from camera to each remaining light using its appearance model and Equation 6, providing distance constraints for use in pose optimisation.
6) Finally, optimise to correct the pose estimate.

B. *Pose Prediction*

Assume we are given an estimate of the vehicle pose at time $k-1$ which we write as $\hat{T}_{k-1} \in \mathbb{SE}(3)$. Additionally we are given an estimate of incremental motion between time $k-1$ and time k. We write a predicted vehicle pose at time k as:

$$\hat{T}_k = \hat{T}_{k-1} \oplus T^{VO}_{k-1,k} \tag{7}$$

Where $T^{VO}_{k-1,k}$ is an estimate of our incremental motion which in our case comes from a VO system. We have found that even at night this offers performance levels commensurate with those we required to produce a seed solution to the localiser.

C. *Points Correspondence*

Light detection is done as described in Section IV-B. From this we obtain a set of n_k candidate blobs $\mathcal{B}^k = \left\{\mathcal{B}^k_1, \ldots, \mathcal{B}^k_{n_k}\right\}$. All m_k lights $\mathcal{L}_k$ in the map within a distance λ_D of the current pose estimate $\hat{T}_k$ are considered as observable candidates. Their predicted positions in the camera image are computed by reprojecting them into the current frame.

The joint compatibility branch-and-bound algorithm (JCBB) [17] is used to obtain an initial set of point correspondences. JCBB computes the maximal set of jointly compatible correspondences and as such preserves the correlation between measurements and predictions. However, as its running time is exponential in the number of measurements, we must invoke a policy to limit the number of candidate blobs to $\lambda_J \approx 20$. Therefore, if more than λ_J blobs are detected in a frame, we use a nearest-neighbour approach to prioritise candidate blobs that are closest to predicted locations of lights.

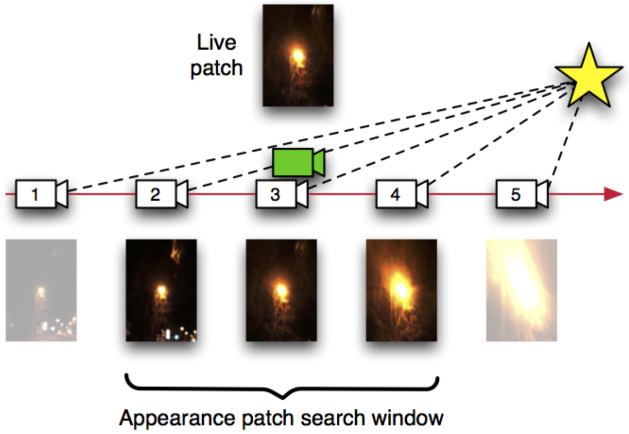


Fig. 6. As part of our outlier rejection step, performed after JCBB returns a set of tentative associations, a window of stored appearances is searched for a match for our live light observation. In this example, five stored appearances of the candidate light and corresponding camera poses at their time of capture are shown. Our *predicted* live camera pose, relative to the map, is shown in green. Map pose 3 is closest (by distance) to our predicted location and so we search a window centred at appearance 3.

1) Modelling Uncertainty: We consider the measurement uncertainties for each light $\mathcal{L}_j \in \mathcal{L}_k$ independently so that the full covariance matrix takes the form $R = \operatorname{diag}(R_1, \ldots, R_{m_k})$, where each $R_j = \operatorname{diag}(r_u, r_v)$ represents the uncertainty of that light's centre in image space. We reason that due to over-exposure and flaring effects, the observed centre of closer lights (i.e. larger blobs) is more uncertain than the observed center of distant lights. Therefore, for each light $\mathcal{L}_k$, we scale r_u and r_v by the width and height of the closest appearance patch, respectively. Our process covariance matrix P is chosen to simply allow for a small amount of variance in each pose parameter.

2) Appearance Matching: The set of 2D-to-3D correspondences returned by JCBB may still contain outliers, but we have yet to consider the crucial deciding factor: whether the lights look how we remember them.

Recall that for every light $\mathcal{L}_i$ in our map, we have a set of (appearance patch, camera pose) pairs $(A_{i,j}, X_{i,j})$, in temporal order, representing every observation of that light and where it was observed from. Intuitively, we would expect our observation to best match the appearance that was captured at a location closest to our predicted pose. However, due to the uncertainty in our pose estimate, we also consider nearby appearances. This is as a result of the potential for a light's appearance to change considerably in the space of a few metres.

Therefore, we consider a window of stored appearances:

$$\bar{\mathcal{A}}_i(\bar{\mathbf{x}}) = \left\{\mathcal{A}_{i,j+k} : j = \arg\min_j \|X_{i,j} - \bar{\mathbf{x}}\|, |k| \le \omega\right\} \tag{8}$$

Where parameter ω, determining the window size, can be fixed or can vary according to our pose uncertainty. $\bar{\mathbf{x}}$ is the current pose estimate. Figure 6 shows how this matching process works.

We quantify appearance similarity by comparing image patches in two different ways:

1) By normalised cross-correlation (NCC) between greyscale appearance patches. This ensures the observed blob roughly matches the expected size, shape, and intensity of the corresponding light. We select the appearance $\mathcal{A}_{i,j} \in \bar{\mathcal{A}}_i$ with the highest NCC score before testing it again in the next step:
2) By measuring appearance patch compatibility in colour space. A simple euclidean distance measure in YUV space is more than sufficient to distinguish between lights of different colours, for example red/green traffic lights and yellow sodium street lights.

If both the highest NCC score and corresponding YUV distance are below a certain threshold, the correspondence is finally classed as an inlier.

D. Pose Update

A delta pose estimate, ΔT, is computed by minimising the cost function given by Equation 1. This is where our individual appearance models come into play again – as well as taking reprojection errors into account, our cost function also considers how far away we appear to be from each observed light. These distances are calculated based on the area of each light's observed blob using Equation 6. These constraints allow the system to arrive at reasonable solutions for configurations not possible with reprojection alone.

We compute ΔT using Equation (2) with $\hat{\mathbf{x}} = \mathbf{0^{6\times1}}$ as a prior (as we have already applied the VO frame-to-frame transformation) and lastly update the current pose estimate to obtain a final pose for this frame:

$$T_k = \hat{T}_k \oplus \Delta T \tag{9}$$

VI. Results

In this section we compare our localisation system (NightNav) against our stereo point feature based teach-and-repeat localiser (VT&R) [18]. For both mapping and localisation we used our survey vehicle, a modified Bowler Wildcat equipped with a front-facing Bumblebee2 stereo camera (used by VO and VT&R) and a front-mounted Ladybug2 omni-directional camera system (used by NightNav). The Ladybug2 consists of six separate cameras: five are arranged radially providing 360° field of view and a sixth points directly up. We use only the five radially mounted cameras as these provide the best views of lights surrounding the vehicle but it is worth noting that our technique generalises to any number of monocular cameras in any configuration. The Automatic Multi-Camera Calibration Toolbox [19] was used for camera extrinsic calibration, with intrinsic parameters provided by the manufacturer.

For our results we used four datasets collected from a well-lit urban route around central Oxford (shown in Figure 7) measuring approximately 4km in length. Datasets 1 and 2 were collected in January 2014 and datasets 3 and 4 were collected in June 2014. A map was built with dataset 1 using the method described in Section IV. Our VO system [20] determined the map trajectory. We found it to be sufficiently locally accurate despite not being optimised for low-light conditions. The vehicle was then localised in subsequent traversals (datasets 2, 3 and 4) using the method described in Section V.

Threshold parameter values were determined experimentally. We found for our setup that $220 \leq \lambda_B \leq 240$ (where $\lambda_B \in [0, 255]$ is a greyscale value) was a suitable range for the image intensity threshold that preserved bright light centres but not reflections. During map construction, only lights tracked for a minimum of $\lambda_M = 10$ frames were considered reliable. Similarly, $\lambda_D = 80\text{m}$ was found to be suitable maximum distance for candidate lights as this is around the distance at which a typical streetlight becomes visible in the intensity-thresholded image. For the weighting of the prior in our cost function (Equation 1), a value of $W = I^{6\times6}$ was found to give the most robust results.

Figure 8 shows the proportion of the total distance traversed in each dataset during which each system travelled more than a particular distance without successfully localising. For NightNav, a successful localisation requires a minimum of two matching lights. For VT&R, a successful localisation requires a minimum of three feature correspondences. Note that both NightNav and VT&R were configured to exhaustively search from the last known good location in their maps in an attempt to reseed their locations if they travelled more than 30m by dead-reckoning. In most of these cases, NightNav managed to relocalise almost immediately whereas VT&R would continue for many metres before recovering. These failures occurred mainly along relatively dark sections of the route such as that highlighted in Figure 7. In several instances VT&R also gets lost when other vehicles obscure parts of the scene. It is clear from these results that because NightNav relies on permanent features in the environment rather than transient point features, it gets lost less often and for shorter distances.

Localisation is challenging for both NightNav and VT&R in datasets 3 and 4 compared to dataset 2, however NightNav continues to outperform VT&R. We believe the main reason for this degradation is the difference in foliage coverage between winter and summer. In many sections of the route, lights are partially or fully obscured by leaf-covered trees.

Figure 9 shows the computed trajectory for both NightNav and live VO projected onto the X-Y plane, in comparison to the map trajectory. This shows NightNav localising successfully within the map and demonstrates that it does not suffer from drift that affects uncorrected VO. Also shown is the cross track error in x, y, and z, measured relative to the map trajectory. The median x, y, and z errors over all three datasets are 0.35m, 0.36m, and 0.31m respectively. In computing these errors we assume that we precisely follow the map trajectory route and therefore these values can be partly attributed to variations in position when driving in lane. Large spikes in cross track error, however, occur when a lack of visible lights causes a localisation failure. Again, this happens most frequently when localising against datasets that were collected five months after the map dataset.

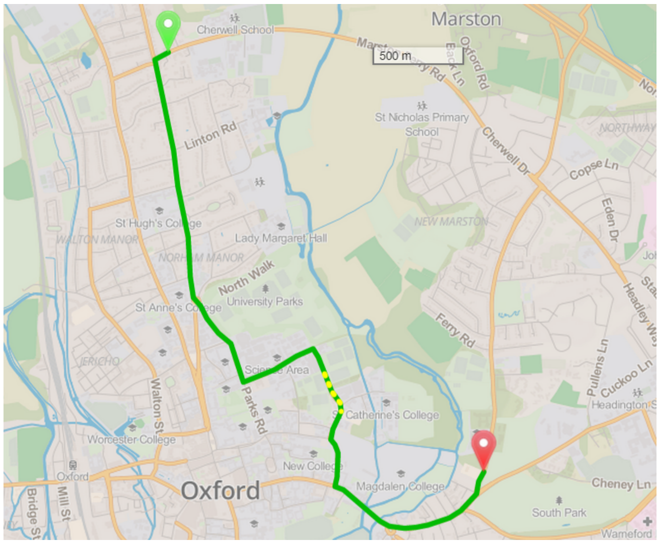
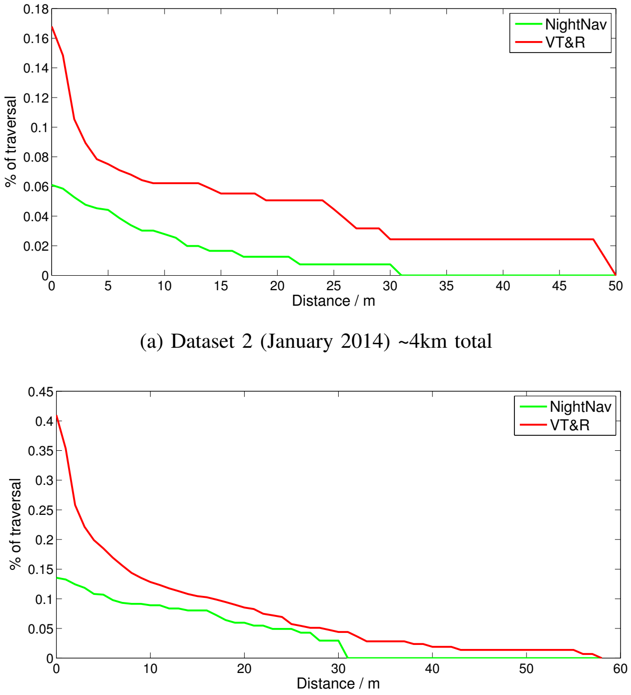


Fig. 7. A map showing the ~4km route taken through central Oxford in all four datasets. Street lighting is present on both sides of the road along its entire extent, with the exception of the section shown in dotted yellow where illumination is sparse and it is exceptionally dark. Consequently, both VT&R and NightNav had difficulties localising in this area. Map imagery from OpenStreetMap[2].



(a) Dataset 2 (January 2014) ~4km total

(b) Datasets 3 and 4 (June 2014) ~8km total

Fig. 8. Graphs showing the proportion of the total traversal during which each system has travelled more than a certain distance without successfully localising. For example in Figure (a), VT&R spends almost 17% of the total traversal in a state where it is lost for more than 0 metres. In contrast, NightNav spends only 6% of the total traversal in the same state. Localisation performance degrades in datasets 3 and 4 (Figure (b)), which were both captured five months after the map dataset. However, NightNav continues to outperform VT&R.

VII. Conclusions

We have demonstrated a novel system that builds maps of artificial light sources and subsequently uses that map to localise at night using vision only. By modelling (via a database of appearances) the behaviour of the light sources at night, we are able deal with saturation, blurring, and distance-dependent lens flare in images. We show that this system out performs a point feature based localiser over many kilometres of testing in a city at night.

VIII. Acknowledgements

Peter Nelson is supported by an EPSRC Doctoral Training Account. Paul Newman is supported by EPSRC Leadership Fellowship EP/I005021/1. Ingmar Posner is supported by the European Project V-Charge.

References

[1] Michael J Milford, Ian Turner, and Peter Corke. Long exposure localization in darkness using consumer cameras. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3755–3761. IEEE, 2013.

[2] Ronan O'Malley, Edward Jones, and Martin Glavin. Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions. *Intelligent Transportation Systems, IEEE Transactions on*, 11(2):453–462, 2010.

[3] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.

[4] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006*, pages 430–443. Springer, 2006.

[5] Andrew J Davison and David W Murray. Simultaneous localization and map-building using active vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):865–880, 2002.

[6] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23, 2006.

[7] Mark Maimone, Yang Cheng, and Larry Matthies. Two years of visual odometry on the mars exploration rovers: Field reports. *Journal of Field Robotics*, 24(3):169 – 186, March 2007.

[8] G. Sibley, C. Mei, P. Newman, and I. Reid. A system for large-scale mapping in constant-time using stereo. *International Journal of Robotics Research*, 2010.

[9] Paul Furgale and Timothy D Barfoot. Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics*, 27(5):534–560, 2010.

[10] Kiyoshi Irie, Tomoaki Yoshida, and Masahiro Tomono. A high dynamic range vision approach to outdoor localization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5179–5184. IEEE, 2011.

[11] Allan R Eisenman, Carl C Liebe, and John L Joergensen. New generation of autonomous star trackers. In *Aerospace Remote Sensing'97*, pages 524–535. International Society for Optics and Photonics, 1997.

[12] Michael J Milford and Gordon Fraser Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1643–1649. IEEE, 2012.

[13] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

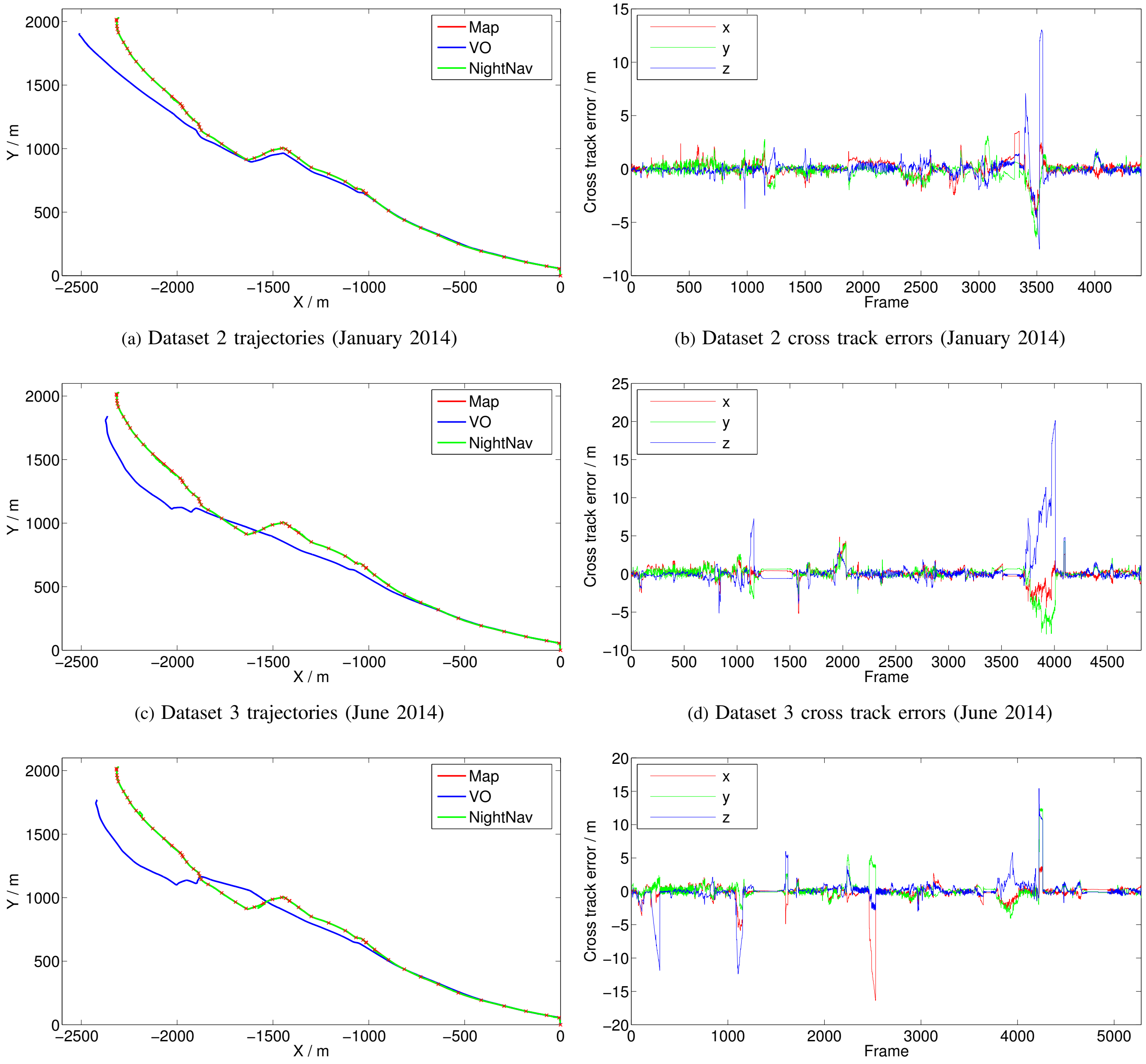
[14] Peter J Huber et al. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

[15] Timothy Barfoot, James R. Forbes, and Paul T. Furgale. Pose estimation using linearized rotations and quaternion algebra. *Acta Astronautica*, 68(1–2):101–112, 2011.

[16] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision algorithms: theory and practice*, pages 298–372. Springer, 2000.

[17] José Neira and Juan D Tardós. Data association in stochastic mapping using the joint compatibility test. *Robotics and Automation, IEEE Transactions on*, 17(6):890–897, 2001.

[2] http://www.openstreetmap.org/



(a) Dataset 2 trajectories (January 2014)

(b) Dataset 2 cross track errors (January 2014)

(c) Dataset 3 trajectories (June 2014)

(d) Dataset 3 cross track errors (June 2014)

(e) Dataset 4 trajectories (June 2014)

(f) Dataset 4 cross track errors (June 2014)

Fig. 9. Figures (a), (c), and (e) show NightNav (green) trajectories in the X-Y plane in comparison to the map trajectory (red) for datasets 2, 3, and 4 respectively, showing that it successfully localises. Uncorrected, integrated live VO (blue) is included for illustrative purposes and as expected, suffers from drift. Note that as our map is built using a purely relative pose framework, it is not globally consistent and hence does not precisely match the shape of the metrically-accurate trajectory shown in Figure 7. Figure (b), (d), and (f) show the cross track errors in x, y, and z for NightNav's trajectories in relation to the map trajectory. Large spikes are localisation failures caused by lack of visible lights. These failures occur more often in datasets 3 and 4 as street lights are obscured by increased foliage as a result of seasonal change between January and July.

[18] Winston Churchill and Paul Newman. Experience-based navigation for long-term localisation. *The International Journal of Robotics Research*, 32(14):1645–1661, 2013.

[19] Michael Warren, David McKinnon, and Ben Upcroft. Online calibration of stereo rigs for long-term autonomy. In *International Conference on Robotics and Automation (ICRA)*, Karlsruhe, 2013.

[20] Winston Churchill. *Experience Based Navigation: Theory, Practice and Implementation*. PhD thesis, University of Oxford, 2012.