Vision-Only Localisation Under Extreme Appearance Change

Chris Linegar

New College

Supervisor:

Professor Paul Newman

Mobile Robotics Group

Department of Engineering Science

University of Oxford

August 2016

Chris Linegar
New College

Doctor of Philosophy
August 2016

Vision-Only Localisation Under Extreme Appearance Change

Abstract

Robust localisation is a key requirement for autonomous vehicles. However, in order to achieve widespread adoption of this technology, we also require this function to be performed using low-cost hardware. Cameras are appealing due to their information-rich image content and low cost; however, camera-based localisation is difficult because of the problem of appearance change. For example, in outdoor environments the appearance of the world can change dramatically and unpredictably with variations in lighting, weather, season and scene structure. We require autonomous vehicles to be robust under these challenging environmental conditions.

This thesis presents Dub4, a vision-only localisation system for autonomous vehicles. The system is founded on the concept of experiences, where an "experience" is a visual memory which models the world under particular conditions. By allowing the system to build up and curate a map of these experiences, we are able to handle cyclic appearance change (lighting, weather and season) as well as adapt to slow structural change. We present a probabilistic framework for predicting which experiences are most likely to match successfully with the live image at run-time, conditioned on the robot's prior use of the map. In addition, we describe an unsupervised algorithm for detecting and modelling higher-level visual features in the environment for localisation. These features are trained on a per-experience basis and are robust to extreme changes in appearance, for example between rain and sun, or day and night.

The system is tested on over 1500 km of data, from urban and off-road environments, through sun, rain, snow, harsh lighting, at different times of the day and night, and through all seasons. In addition to this extensive offline testing, Dub4 has served as the primary localisation source on a number of autonomous vehicles, including the Oxford University's RobotCar, the 2016 Shell Eco-Marathon, the LUTZ PathFinder Project in Milton Keynes, and the GATEway Project in Greenwich, London.

Statement of Authorship

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfilment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Chris Linegar, New College

Funding

Chris Linegar is funded by the Rhodes Trust.

Acknowledgements

I would like to thank my supervisor, Professor Paul Newman, for his insights, advice and encouragement throughout my time in Oxford. Being part of the Mobile Robotics Group has been exciting from Day 1, and to see our software commanding fleets of robots and autonomous cars has been tremendously rewarding – this would not have been possible without Paul's energy and enthusiasm for robotics. I would like to thank everyone in the lab for making it such an enjoyable place to work, from the integration weeks and field trials, to the lunch-time football games.

I would also like to thank my parents and my brother, for the years and years of love and support which got me here. They taught me to think and ask questions, to challenge my ideas, and not to shy away from hard work. I'd also like to mention my grandparents, for their unwavering support and encouragement.

Most importantly, I would like to thank my wife, Caterina. What an incredible adventure – none of this would have been nearly as much fun without you.

Chris Linegar

August 2016

Contents

Chapter 1

Introduction

1.1 Motivation

An autonomous vehicle must address three main challenges in order to enter an autonomous mode. Firstly, it must determine where it is in the world. Secondly, it must be able to perceive its local environment, identifying obstacles and free space. Finally, it must use this information to determine a safe action which will move the vehicle towards its goal. This thesis addresses localisation, the first of these three challenges.

More specifically, this thesis is concerned with 6-DOF vision-only localisation in outdoor environments using a single stereo camera. Camera-based localisation is appealing due to the low sensor cost and information-rich image content, however it is a challenging task because of the problem of appearance change. In outdoor environments, the appearance of the world can change dramatically and unpredictably with changes in weather, lighting, season, and scene structure. Autonomous vehicles are required to be robust under these changing conditions.

The research presented here has been motivated by a large-scale, long-term data collection effort. Over the past three years, we have logged more than 1500 km of

Figure 1.1: This figure demonstrates the challenge of performing robust vision-only localisation in outdoor environments. Each image in the montage is a snapshot from the same place as the robot traversed a single lap of the Oxford 10 km route (note that the blue sign post is visible in all images). The appearance of the scene is subject to sudden and unpredictable changes due to varying lighting, weather and seasonal conditions.

outdoor driving, in rain, sun, snow, at all times of the day, across different cities and challenging terrains. This data has been instrumental in the design, testing and validation of the algorithms described here. Most notable of these datasets is the Central Oxford dataset, a dataset containing 100 repeats of a 10 km route through busy central Oxford. Figure 1.1 illustrates the diversity of appearance change encountered over the 18 month period, demonstrating the challenge faced by a vision-only localisation system. This thesis is motivated by the need for autonomous vehicles which are able to operate anywhere, at any time of the day, and in all weather conditions.

1.2 Contributions and Publications

This thesis makes the following contributions in the field of vision-only localisation:

1. During the development of the algorithms described in this thesis, we have collected over 1500 km of data from different areas in the UK. Most notable is the Central Oxford dataset which contains 100 repeats of a 10 km route in central Oxford, totalling over 1000 km of data. While other datasets may exist of similar or greater size, this dataset is unique in the high number of traversals of the same route under a challenging set of appearance conditions. A public release of the Oxford 1000 km dataset was made earlier this year (Maddern et al., 2017)[1].

2. As one of the first outputs of this thesis, we present a real-time, C++ implementation of Experience-Based Navigation (EBN). This implementation performs robust localisation, unsupervised mapping, automatic loop closure detection, and automatic initialisation using FAB-MAP (Cummins and New-

[1]This thesis acknowledges the contributions of a number of MRG members in vehicle systems integration, testing, and data collection.

man, 2008). In addition, an intelligent memory management technique is presented which allows the robot to learn which experiences will be relevant at run-time. This method was submitted by Linegar et al. (2015), where it won the ICRA 2015 Best Robotic Vision Paper Award. A patent has been filed on the algorithm with patent application number PCT/GB2015053723. Further implementation details are discussed by Nelson et al. (2015b).

3. We describe and demonstrate a new algorithm for extracting and modelling distinctive landmarks in the environment using high level visual information. We refer to the generated detectors as "patch detectors", due to their shape and relatively large size when compared with traditional point feature methods. This method operates on a single dataset and does not rely on GPS or manual alignment of training images. It enables localisation across extreme changes in appearance by capturing the underlying structure in the environment. This technique was published by Linegar et al. (2016).

4. The primary contribution of this thesis is a new vision-only localisation system called Dub4. The system can be thought of as a physical instantiation of the theory and algorithms presented in this thesis. It levers a suite of algorithms and techniques to perform robust pose estimation in spite of extreme appearance changes. These techniques include the multi-experience paradigm, whereby the map is able to store multiple, overlapping representations of the environment under different conditions; the illumination invariant transform to reduce the effect of shadows; the use of multiple feature types such as BRIEF and SURF; as well as patch detectors for coarse, but robust, data associations between images. We test and validate this system on over 1500 km of outdoor driving.

5. Finally, the Dub4 localiser is integrated within a full autonomy stack. Dub4

Figure 1.2: Photos of the LUTZ PathFinder trials which took place along pedestrian walkways in Milton Keynes. Dub4, the vision-only localiser described in this thesis, was used to localise the vehicle for the duration of this week-long trial.

has been used as the primary localisation source on a number of autonomous vehicles over the past year, including the Oxford University RobotCar, the 2016 Shell Eco-Marathon, the LUTZ PathFinder Project in Milton Keynes, and the GATEway Project in London. Figure 1.2 displays photos from the LUTZ PathFinder Pod trials in Milton Keynes[2].

The theory, algorithms and software described here continue to be used extensively within the Oxford Robotics Institute, as well as more widely with commercial partners in the automotive industry.

[2]Note that this thesis is concerned solely with the localisation system. Additional subsystems in the full autonomy stack, for example the perception and planning subsystems, are not described here.

1.3 Thesis Roadmap

This thesis describes the development and testing of a robust, real-time localisation system for autonomous vehicles. This work is structured as follows.

Chapter 2 provides a summary of preliminary concepts and theory. This includes the fundamentals relating to camera geometry, visual odometry and pose estimation. Additionally, the chapter reviews state-of-the-art techniques in performing robust localisation under appearance change.

Chapter 3 presents the large-scale datasets used in later chapters, as well as the key performance metrics used to measure localisation success and failure.

Chapter 4 presents our implementation of the experience-based paradigm for vast-scale mapping and localisation. The notion of experiences is fundamental in our approach to robust localisation, where the robot's map of the world evolves over time to adapt to a changing world. Chapter 5 presents a probabilistic framework for retrieving relevant experiences to support resource-constrained localisation.

Chapter 6 describes a new technique for localisation, where large, distinctive landmarks in the environment are extracted and modelled using linear SVM classifiers. We show that these "patches" are significantly more robust to appearance change than traditional point feature approaches.

Chapter 7 presents Dub4, a vision-only localisation system for autonomous vehicles operating in outdoor environments. The system draws on a number of complementary techniques to perform robust localisation across extreme appearance change. We present localisation results on over 1000 km of driving in central Oxford, as well as a further 200 km of driving in off-road terrain.

Finally, Chapter 8 presents a summary of the work and results presented here.

Chapter 2

Preliminaries

2.1 Camera Geometry

This section presents a summary of fundamental concepts in camera geometry. Hartley and Zisserman (2003) provide a detailed description of multi-view geometry which is particularly relevant to the work here.

2.1.1 Homogeneous Co-ordinates

A camera projects points in $\mathbb{R}^3$ onto an image sensor plane in $\mathbb{R}^2$. Points on the plane are represented by the vector $(x, y)^T$ in $\mathbb{R}^2$, although it is often convenient to express the same co-ordinate in projective space $\mathbb{P}^2$. The point in $\mathbb{R}^2$ can be represented in $\mathbb{P}^2$ using the homogeneous co-ordinate $(x, y, 1)^T$, or more generally $(kx, ky, k)^T$ for non-zero k, since scale is unimportant in projective space. So, given a homogeneous co-ordinate $(x_1, x_2, x_3)^T$ in $\mathbb{P}^2$, the equivalent co-ordinate in $\mathbb{R}^2$ is $(\frac{x_1}{x_3}, \frac{x_2}{x_3})^T$. Figure 2.1 shows how the projective space $\mathbb{P}^2$ can be thought of as a set of rays in $\mathbb{R}^3$ passing through the origin O.

This concept extends to higher-dimensional spaces. For example, a point $(x, y, z)^T$ in $\mathbb{R}^3$ can be represented in $\mathbb{P}^3$ as a homogeneous co-ordinate $(x, y, z, 1)^T$, which is

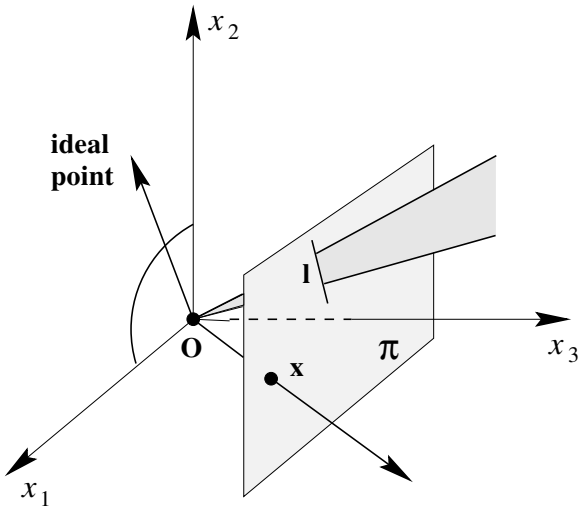


Figure 2.1: Figure demonstrating how a point $(x_1, x_2, x_3)^T$ in $\mathbb{P}^2$ can be thought of as a line in $\mathbb{R}^3$. The equivalent point in $\mathbb{R}^2$ is thus $(\frac{x_1}{x_3}, \frac{x_2}{x_3})^T$. Image credit: Hartley and Zisserman (2003).

equivalent to $(kx, ky, kz, k)^T$ for any non-zero k.

One of the advantages of homogeneous co-ordinates is that they provide a natural mechanism for handling points at infinity. Given a homogeneous co-ordinate $(x_1, x_2, x_3, x_4)^T$ in $\mathbb{P}^3$, setting $x_4 = 0$ corresponds to a point at infinity in $\mathbb{R}^3$. We will make use of this in later chapters.

2.1.2 $\mathbb{SE}(3)$ Transformations

A rigid body transformation is a transformation which preserves the distances between points, and the angle between vectors. By extension, a rigid body transformation maps a right-handed, orthogonal co-ordinate frame to a right-handed, orthogonal co-ordinate frame, such that angles and distances are preserved (Murray et al., 1994).

A rigid body transformation $\mathbf{G}_{A,B} \in \mathbb{SE}(3)$ is a 4×4 homogeneous transformation matrix which transforms a homogeneous point $\mathbf{x}_B$ in reference frame B, to homogeneous point $\mathbf{x}_A$ in frame A:

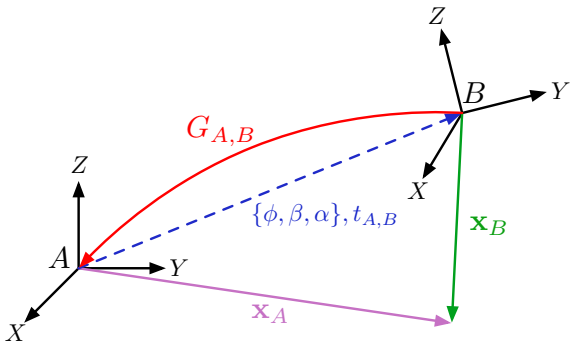


Figure 2.2: Figure illustrating how a rigid body transformation $\mathbf{G}_{A,B}$ transforms the point $\mathbf{x}_B$ in co-ordinate frame B, to $\mathbf{x}_A$ in co-ordinate frame A.

$$\mathbf{x}_A = \mathbf{G}_{A,B}\mathbf{x}_B$$

where $\mathbf{x} = (x, y, z, 1)^T$ is the homogeneous co-ordinate form. This is illustrated in Figure 2.2. The homogeneous matrix $\mathbf{G}_{A,B}$ is of the form:

$$\mathbf{G}_{A,B} = \begin{bmatrix} \mathbf{R}_{A,B} & \mathbf{t}_{A,B} \\ \mathbf{0} & 1 \end{bmatrix}$$

where $\mathbf{R}_{A,B} \in \mathbb{SO}(3)$ is the rotation matrix, and $\mathbf{t}_{A,B} \in \mathbb{R}^{3x1}$ is the translation vector, also illustrated in Figure 2.2. The NASA convention for co-ordinate frames is used, where X is forward, Y is right, and Z is down. The translation and orientation components of the transformation are formed by the position and orientation of the source frame B, in the destination frame A.

The rotation matrix $\mathbf{R}_{A,B} \in \mathbb{SO}(3)$ is a 3×3 matrix. It is an over-parameterised representation where six constraints ensure that its columns are mutually orthogonal and of unit length. This rotation can be represented in different ways, depending on how the rotation is being used. An overview of different representations is presented below.

Euler angles consist of a sequence of elemental rotations about each axis in the co-ordinate frame. In our implementation, a rotation in $\mathbb{SO}(3)$ is parameterised

by a sequence of three such rotations about the Z-Y-X axes, corresponding to the yaw (α), pitch (β) and roll (ϕ) axes, respectively. The rotation matrix is then constructed:

$$\begin{aligned}
\mathbf{R}_{A,B} &= \mathbf{R}_z(\alpha) \cdot \mathbf{R}_y(\beta) \cdot \mathbf{R}_x(\phi) \\
&= \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \\
&= \begin{bmatrix} c_\alpha c_\beta & (c_\alpha s_\beta s_\phi - s_\alpha c_\phi) & (c_\alpha s_\beta c_\phi + s_\alpha s_\phi) \\ s_\alpha c_\beta & (s_\alpha s_\beta s_\phi + c_\alpha c_\phi) & (s_\alpha s_\beta c_\phi - s_\phi c_\alpha) \\ -s_\beta & c_\beta s_\phi & c_\beta c_\phi \end{bmatrix}
\end{aligned} \tag{2.1}$$

where the abbreviations s_α and c_α refer to $\sin(\alpha)$ and $\cos(\alpha)$, respectively. The ranges of the yaw (α), pitch (β) and roll (ϕ) angles are defined to be $\{\phi, \alpha\} \in [-\pi, \pi], \beta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. Euler angles are a minimal representation of the rotation matrix. This is appealing, however the mapping into Euclidean space contains singularities (Stuelpnagel, 1964). These singularities are sometimes referred to as the *gimbal lock* problem, where two of the three axes of rotation align, resulting in the loss of one degree of freedom.

Quaternions have also been used to parameterise 3D rotations (Hamilton, 1844). A quaternion is represented by the 4-vector $\mathbf{q} = (q_x, q_y, q_z, q_w)$ with unit length $\|\mathbf{q}\| = 1$. This representation has the advantage that because it is over-paramterised, it is not subject to the gimbal lock problem as Euler angles are. Additionally, it is a more compact representation than the 3×3 rotation matrix, facilitating faster calculations. However, as with all over-parameterised representations, care needs to be taken to ensure that the constraints on the representation are maintained.

Lastly, the exponential map provides another parameterisation of the rotation (Grassia, 1998). The exponential map interprets a non-zero vector in $\mathbb{R}^3$ as a rotation, where the axis and magnitude of rotation is specified by the direction and magnitude of the vector, respectively. The zero vector is used to represent the identity rotation, such that the space is continuous. The exponential map is a minimal representation so it is not subject to additional constraints. Singularities arising from the minimal representation can be avoided by dynamic re-parameterisation.

We discuss our choice of parameterisation later in this chapter within the context of an optimisation for camera pose.

2.1.3 Camera Model

A projective camera $\mathbf{P}$ maps points in the world onto the camera's image plane according to:

$$\mathbf{x} = \mathbf{P}\mathbf{G}_{c,w}\mathbf{X}_w \tag{2.2}$$

where $\mathbf{x}$ is the homogeneous co-ordinates of the projection onto the image plane, $\mathbf{P}$ is the camera's projection matrix, $\mathbf{G}_{c,w}$ describes the position and orientation of the camera frame c with respect to the world frame w, and $\mathbf{X}_w$ is the homogeneous co-ordinates of a point in the world.

The projection matrix $\mathbf{P}$ is a function of the internal characteristics of the camera, and can be expressed as $\mathbf{P} = \mathbf{K}[\mathbf{I}|\mathbf{0}]$ where $\mathbf{K}$ is the 3×3 camera calibration matrix. The matrix $\mathbf{K}$ is a 3×3 matrix which describes the internal characteristics of the camera:

$$\mathbf{K} = \begin{bmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

where f_x and f_y are the focal lengths in the x- and y-axes, s is a parameter for skew

distortion in the image, and x_0 and y_0 are the principal point offsets. Focal length is measured in pixels here, so the focal length in x- and y-axes will be different if the pixels are not square in shape. The skew parameter s is zero for most cameras, but can be non-zero if the x- and y-axes are not perpendicular. The principal point offsets x_0 and y_0 describe the alignment between the sensor plane and the camera centre. Camera calibration can be performed as a once-off offline task, as discussed by Warren et al. (2013); Hartley and Zisserman (2003). For the purposes of this thesis, we will assume the intrinsic calibration matrix $\mathbf{K}$ is known for all cameras. Equation 2.2 can now be expressed as:

$$\mathbf{x} = \mathbf{K}[\mathbf{I}|\mathbf{0}]\mathbf{G}_{c,w}\mathbf{X}_w \tag{2.4}$$

This relationship is used extensively in vision-only localisation, as it describes how 3D landmarks in the world should correspond to 2D pixel observations on the image plane.

2.1.4 Epipolar Geometry

Epipolar geometry describes the intrinsic projective geometry between two overlapping camera views. It is independent of the scene being observed and depends only on the cameras' internal parameters and the relative pose between the cameras.

The fundamental matrix $\mathbf{F}$ describes this relationship. It is a 3×3 matrix which stipulates that if a point in the world $\mathbf{X}$ is observed by both cameras as $\mathbf{x}$ and $\mathbf{x}'$, the image points must satisfy the relation $\mathbf{x}'^T\mathbf{F}\mathbf{x} = 0$.

The epipolar plane π demonstrates this geometric relationship, as shown in Figure 2.3. An epipolar line is the intersection of an epipolar plane with the image plane. The points $\mathbf{x}$ and $\mathbf{x}'$ must lie on their respective epipolar lines. An application of epipolar geometry is in determining stereo correspondences. If a feature

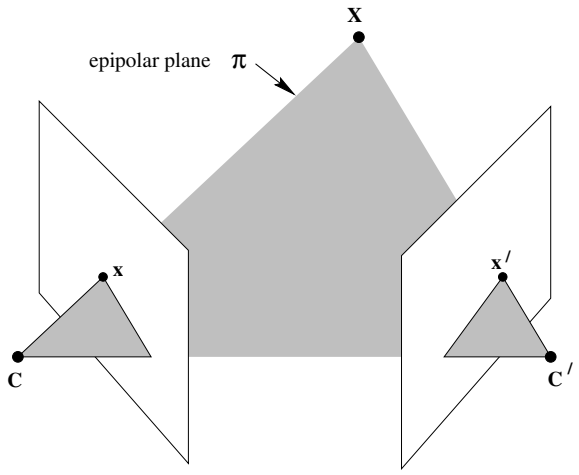
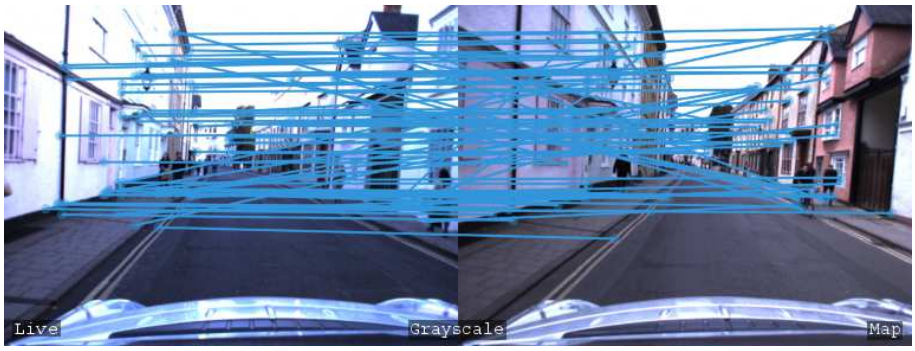


Figure 2.3: Two camera views of a scene are shown. An observation $\mathbf{x}$ in the first frame generates a ray in $\mathbb{R}^3$ which intersects the camera centre $\mathbf{C}$, $\mathbf{x}$ and world point $\mathbf{X}$. Similarly, a second ray is generated by the observation $\mathbf{x}'$ and camera centre $\mathbf{C}'$ in the second frame. Both rays intersect the world point $\mathbf{X}$, and together with the baseline between $\mathbf{C}$ and $\mathbf{C}'$, define the epipolar plane π. Image credit: Hartley and Zisserman (2003).

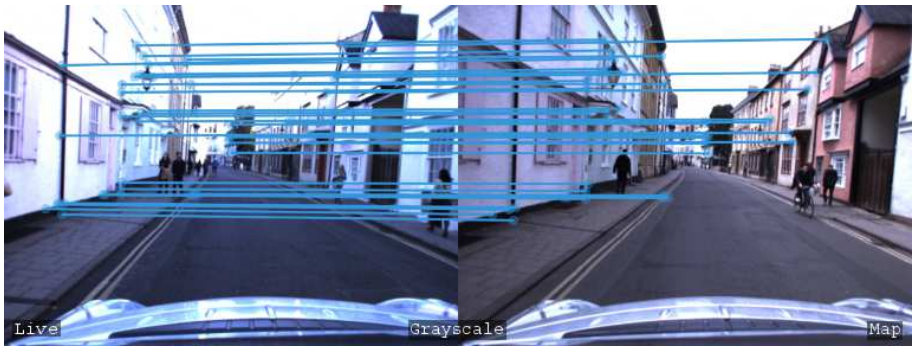
is observed in one frame, the search for that feature in the second frame can be constrained along the epipolar line.

2.2 Robust State Estimation

An important competency within a localisation system is the ability to estimate the relative 6-DOF pose between stereo image pairs. Within the context of visual odometry, we estimate the motion of the camera by solving for the relative transformations between consecutive image pairs. During localisation, the relative pose between the live camera image and the map is estimated to provide closed-loop updates. This section describes the algorithms used for model fitting, outlier rejection and non-linear least squares refinement, with reference to the pose estimation problem.



(a) All data associations



(b) Inliers only

Figure 2.4: Pose estimation is performed between the live image (left) and an image from the map (right). Figure 2.4a shows the raw data associations obtained from feature matching, where BRIEF features are used in this example. RANSAC is used to reject outliers, with the remaining inliers plotted in Figure 2.4b.

2.2.1 RANSAC

RANdom Sample Consensus (RANSAC) (Fischler and Bolles, 1981) is an algorithm for robust model fitting given data which may contain outliers. It is an iterative method which generates a set of candidate models using a minimal number of randomly sampled data points for each model. Each candidate model is scored by the total number of data points in agreement with the candidate model, such that the best model is the one with the greatest number of inliers. As such, RANSAC can be used to perform outlier rejection as well as model fitting. The algorithm is as follows:

1. The data is randomly sampled for the minimal number of data points n required to generate a model. At this point, it is unknown whether the sampled data points are inliers or outliers.

2. A candidate model is generated using the sampled data points. The model hypothesises that the sampled data points are all inliers.

3. The remainder of the data points (those not part of the minimal sample) are tested for consensus with the candidate model. A data point is marked as an inlier if it lies within a threshold distance from the value predicted by the candidate model.

4. The candidate model is scored by the number of data points in consensus with the candidate model, where a higher scoring model is preferred.

These steps are repeated for k trials to generate k different candidate models. The highest scoring candidate model is output as the model which best fits the data, together with the set of data points marked as inliers. Increasing the number of trials increases the probability of generating a high-scoring candidate model. In order to ensure a probability z of randomly sampling only from the inlier set, the

number of trials required is $k = \frac{\log(1-z)}{\log(1-w^n)}$ where w is the probability of selecting an inlier from the set of data points.

RANSAC requires a model on which to fit the data. Within the context of pose estimation, the data points are correspondences between 3D landmarks in the map and 2D observations in the image. Figure 2.4 demonstrates how RANSAC is used to reject incorrect data associations from a pose estimation attempt between a live image (left) and an image from the map (right).

The *perspective from n points* (PnP) is a class of problem which attempts to determine the position and orientation of a camera in closed form using n point correspondences. The minimal solution to this problem uses three point correspondences and is referred to as the *perspective from three points* (P3P) method (Fischler and Bolles, 1981). P3P may return multiple solutions, so in practice a fourth point correspondence may be used to remove the ambiguity. However, within the context of RANSAC, multiple solutions are simply treated as multiple candidate models, where low-scoring candidate models are rejected. The P3P method is used to generate the candidate models for RANSAC since the fewer data associations required to generate the model, the higher the probability of drawing a set of data associations which are all inliers, and therefore generating a high-scoring model.

2.2.2 Non-Linear Least Squares Refinement

RANSAC outputs the highest scoring model $\mathbf{x}_0$, where $\mathbf{x}_0$ describes the 6-DOF pose of the camera. While RANSAC is effective in performing outlier rejection, the output model is susceptible to noise in the input data since $\mathbf{x}_0$ is generated using a minimal set of measurements (three data associations when using P3P). An additional output of the RANSAC model-fitting algorithm is the set of M data associations $\{\mathbf{z}_0, ..., \mathbf{z}_M\}$ marked as inliers. We seek to refine the initial pose estimate $\mathbf{x}_0$ using all data associations marked as inliers in order to determine the model

$\mathbf{x}_{ml}$ which best explains the observations $\mathbf{z}$. This is represented by the likelihood function:

$$\mathcal{L} = \prod_{i=1}^{M} p(\mathbf{z}_i|\mathbf{x}) \tag{2.5}$$

which describes the probability of observing the measurements $\mathbf{z}$ given state $\mathbf{x}$. The model $\hat{\mathbf{x}}_{ml}$ is the one which maximises this likelihood function, and is known as the maximum likelihood estimator:

$$\hat{\mathbf{x}}_{ml} = \arg\max_{\mathbf{x}} \prod_{i=1}^{M} p(\mathbf{z}_i|\mathbf{x}) \tag{2.6}$$

We model the measurements, or data associations, as being subject to Gaussian noise:

$$\mathbf{z}_i = \bar{\mathbf{z}} + \mathbf{w}_i, \quad \mathbf{w}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{P_i}) \tag{2.7}$$

where $\bar{\mathbf{z}}$ is the true measurement and $\mathbf{w}_i$ is zero mean Gaussian noise with covariance $\mathbf{P}_i$. The covariance is an $m \times m$ matrix, where m is the size of the measurement vector. We assume that all correspondences have the same covariance $\mathbf{P}_i = \mathbf{P}_C$.

The relationship between the true measurement $\bar{\mathbf{z}}$ and the state vector $\mathbf{x}$ is described by the prediction function:

$$\mathbf{h}_i(\mathbf{x}) = \bar{\mathbf{z}}_i \tag{2.8}$$

We discuss the choice of $\mathbf{h}_i(\mathbf{x})$ later in this section. Given the Gaussian noise model, the likelihood function is also Gaussian and can be represented by:

$$p(\mathbf{z}_i|\mathbf{x}) = \prod_{i=1}^{M} \frac{1}{\sqrt{(2\pi)^m \det(\mathbf{P}_C)}} \exp\Big(-\frac{1}{2}(\mathbf{z}_i - \mathbf{h}_i(\mathbf{x}))^T \mathbf{P}_C^{-1} (\mathbf{z}_i - \mathbf{h}(\mathbf{x}_i))\Big) \tag{2.9}$$

The probability $p(\mathbf{z}_i|\mathbf{x})$ is maximised when the difference between the predicted measurement $\mathbf{h}_i(\mathbf{x})$ and observed measurement $\mathbf{z}_i$ is minimised. We refer to this

difference as the error function:

$$\mathbf{e}_i = \mathbf{z}_i - \mathbf{h}_i(\mathbf{x}) \tag{2.10}$$

Alternatively to minimising $\mathcal{L}$, we can minimise the negative log likelihood:

$$\begin{aligned} \hat{\mathbf{x}}_{ml} &= \arg\min_{\mathbf{x}}(-\log\mathcal{L}) \\ &= \arg\min_{\mathbf{x}} \sum_{i=1}^{M} \frac{1}{2}(\mathbf{z}_i - \mathbf{h}_i(\mathbf{x}))^T \mathbf{P}_C^{-1}(\mathbf{z}_i - \mathbf{h}_i(\mathbf{x})) \end{aligned} \tag{2.11}$$

We introduce the function $f(\mathbf{x})$ as the function to be minimised:

$$\arg\min_{x} f(\mathbf{x}) = \frac{1}{2}\mathbf{e}^T\mathbf{P}^{-1}\mathbf{e} \tag{2.12}$$

where $\mathbf{e} = [\mathbf{e}_1, ..., \mathbf{e}_M]^T$, and $\mathbf{P}^{-1} = \text{diag}\{\mathbf{P}_C^{-1}, ..., \mathbf{P}_C^{-1}\}$. This is known as least squares optimisation since the best solution $\mathbf{x}_{ml}$ is the one which minimises the weighted sum of the squared errors.

In our application, $\mathbf{h}(\mathbf{x})$ is a non-linear vector function. The Gauss-Newton (GM) method is an algorithm which can be used to solve non-linear least squares problems. Recall that RANSAC outputs a coarse estimate of the vehicle pose $\mathbf{x}_0$. An iterative optimisation is performed, where at each iteration $\delta\mathbf{x}$ is calculated where $\mathbf{x}_1 = \mathbf{x}_0 + \delta\mathbf{x}$, such that $f(\mathbf{x}_1) < f(\mathbf{x_0})$. We approximate $\mathbf{h}(\mathbf{x})$ using the first order Taylor series expansion:

$$\mathbf{h}(\mathbf{x}_0 + \delta\mathbf{x}) = \mathbf{h}(\mathbf{x}_0) + \mathbf{J}\delta\mathbf{x} \tag{2.13}$$

where $\mathbf{J}$ is the Jacobian of $\mathbf{h}(\mathbf{x})$ and takes the form:

$$\mathbf{J} = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\Delta \mathbf{h}_1}{\Delta \mathbf{x}_1} & \cdots & \frac{\Delta \mathbf{h}_1}{\Delta \mathbf{x}_m} \\ \vdots & \ddots & \vdots \\ \frac{\Delta \mathbf{h}_n}{\Delta \mathbf{x}_1} & \cdots & \frac{\Delta \mathbf{h}_n}{\Delta \mathbf{x}_m} \end{bmatrix} \tag{2.14}$$

Substituting the prediction function $\mathbf{h}(\mathbf{x})$ from Equation 2.13 into the least squares expression in Equation 2.11 yields:

$$\begin{aligned} f(\mathbf{x}_0 + \delta\mathbf{x}) &= \frac{1}{2}(\mathbf{z} - (\mathbf{h}(\mathbf{x}_0) + \mathbf{J}\delta\mathbf{x}))^T \mathbf{P}^{-1}(\mathbf{z} - (\mathbf{h}(\mathbf{x}_0) + \mathbf{J}\delta\mathbf{x})) \\ &= \frac{1}{2}(\mathbf{e} - \mathbf{J}\delta\mathbf{x})^T \mathbf{P}^{-1}(\mathbf{e} - \mathbf{J}\delta\mathbf{x}) \\ &= \frac{1}{2}(\mathbf{e}^T\mathbf{P}^{-1}\mathbf{e} - \mathbf{e}^T\mathbf{P}^{-1}\mathbf{J}\delta\mathbf{x} - (\mathbf{J}\delta\mathbf{x})^T\mathbf{P}^{-1}e + (\mathbf{J}\delta\mathbf{x})^T\mathbf{P}^{-1}\mathbf{J}\delta\mathbf{x}) \\ &= \frac{1}{2}(\mathbf{e}^T\mathbf{P}^{-1}\mathbf{e} - 2\delta\mathbf{x}^T\mathbf{J}^T\mathbf{P}^{-1}\mathbf{e} + \delta\mathbf{x}^T\mathbf{J}^T\mathbf{P}^{-1}\mathbf{J}\delta\mathbf{x}) \end{aligned} \tag{2.15}$$

Deriving with respect to $\delta\mathbf{x}$ results in:

$$\frac{\partial f(\mathbf{x}_0 + \delta\mathbf{x})}{\partial \delta\mathbf{x}} = -\mathbf{J}^T\mathbf{P}^{-1}\mathbf{e} + \mathbf{J}^T\mathbf{P}^{-1}\mathbf{J}\delta\mathbf{x} \tag{2.16}$$

By setting $\frac{\partial f(\mathbf{x}_0+\delta\mathbf{x})}{\partial \delta\mathbf{x}} = \mathbf{0}$, we compute:

$$\delta\mathbf{x} = (\mathbf{J}^T\mathbf{P}^{-1}\mathbf{J})^{-1}\mathbf{J}^T\mathbf{P}^{-1}\mathbf{e} \tag{2.17}$$

This equation is known as the normal equation. It allows us to iteratively refine the estimate $\mathbf{x}$ to output the state $\mathbf{x}_{ml}$, which best estimates the camera's pose from the set of observed data associations. Additionally, the covariance of the least squares solution is:

$$\mathbf{C} = (\mathbf{J}^T\mathbf{P}^{-1}\mathbf{J})^{-1} \tag{2.18}$$

which can be used as a measure of uncertainty in the solution.

2.2.3 M-Estimation

The non-linear least squares optimisation described above estimates the system state $\mathbf{x}$ by minimising the squared errors of the input data:

$$\begin{aligned} \arg\min_{x} f(\mathbf{x}) &= \sum_{i=1}^{M} \mathbf{e}_i^T \mathbf{P}_C^{-1} \mathbf{e}_i \\ &= \sum_{i=1}^{M} r_i^2 \end{aligned} \tag{2.19}$$

where r_i^2 is the squared error of each data point used in the optimisation. The quadratic cost means that corrupted input data significantly degrades the quality of the resulting estimate.

M-estimation introduces a cost function $\rho(\cdot)$ which alters the behaviour of the optimisation to reduce the impact of large errors. The least squares optimisation is re-formulated as:

$$\arg\min_{x} f(\mathbf{x}) = \sum_{i=1}^{M} \rho(r_i) \tag{2.20}$$

where the function $\rho(\cdot)$ is chosen based on the nature of the problem being solved. A common choice in pose estimation problems is the Huber function (Huber, 2011), which is defined as:

$$\rho(r) = \begin{cases} r^2 & \text{if } |r| \leq k \\ k(|r| - \frac{k}{2}) & \text{otherwise} \end{cases} \tag{2.21}$$

where k is a constant. The function enforces a quadratic cost on small errors, while only linear cost on large errors, reducing the impact of corrupted input data on the result.

2.2.4 Application to Pose Optimisation

Throughout this thesis, we are frequently required to solve for the pose of the camera with respect to a set of 3D landmarks. In order to use the non-linear least squares optimisation above, an appropriate parameterisation for the state vector $\mathbf{x}$ and the prediction function $\mathbf{h}(\mathbf{x})$ needs to be chosen. These choices are described as follows.

Given a landmark $\mathbf{p}_s^g$ stored in frame s, and an observation $\mathbf{z}_i^g$ of that landmark from frame i, we calculate the error of that observation using the prediction function:

$$h_{i,s}^g(\mathbf{x}) = \mathcal{K}(\mathcal{C}(\mathcal{T}_{i,s}^g(\mathbf{x}))) \tag{2.22}$$

where $\mathcal{T}_{i,s}^g$ is a function which transforms the landmark $\mathbf{p}_s^g$ into the frame i, $\mathcal{C}$ is a function which transforms the landmark into the camera frame (including the transform between robot and computer vision co-ordinate frames), and $\mathcal{K}$ is the camera projection function. For the purposes here, we will assume a single image is being used for localisation. The derivative of this function with respect to $\mathbf{x}$ is:

$$\frac{\partial h_{i,s}^g}{\partial \mathbf{x}} = \frac{\partial \mathcal{K}}{\partial \mathcal{C}} \frac{\partial \mathcal{C}}{\partial \mathcal{T}_{i,s}^g} \frac{\partial \mathcal{T}_{i,s}^g}{\partial \mathbf{x}} \tag{2.23}$$

The components of this derivative are examined below.

System State

The state vector $\mathbf{x}$ is parameterised using Euler angles. As discussed earlier in this chapter, Euler angles have the advantage of being a minimal representation, however they are susceptible to the problem of *gimbal lock*. We avoid this problem by solving for the small perturbation error $\mathbf{\Delta} = [\delta x, \delta y, \delta z, \delta\theta_r, \delta\theta_p, \delta\theta_q]^T$, ensuring that $\mathbf{\Delta} \approx 0$ by aggregating $\mathbf{\Delta}$ into $\mathbf{x}$ on each iteration.

Transform Derivative

With all landmarks stored against a single frame j, the transform function is described as:

$$\mathcal{T}_{i,j}^{g}(\boldsymbol{\Delta}) = \mathbf{T}(\boldsymbol{\Delta})\mathbf{T}_{i,j}\mathbf{p}_{j}^{g} \tag{2.24}$$

where $\mathbf{T}(\boldsymbol{\Delta})$ is a 4×4 homogeneous matrix describing the error transform, parameterised by $\boldsymbol{\Delta}$:

$$\mathbf{T}(\boldsymbol{\Delta}) = \begin{bmatrix} \mathbf{R}_z(\delta\theta_r)\mathbf{R}_y(\delta\theta_p)\mathbf{R}_x(\delta\theta_q) & [\delta x, \delta y, \delta z]^T \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{2.25}$$

where $\mathbf{R}_\gamma(\theta)$ is a 3×3 rotation matrix which performs a rotation of θ about the axis γ. Deriving Equation 2.24 with respect to $\boldsymbol{\Delta}$, the transform derivative is:

$$\frac{\partial \mathcal{T}_{i,j}^{g}}{\partial \boldsymbol{\Delta}} = \frac{\partial \mathbf{T}(\boldsymbol{\Delta})}{\partial \boldsymbol{\Delta}}\mathbf{T}_{i,j}\mathbf{p}_{j}^{g} \tag{2.26}$$

which is evaluated at zero $\frac{\partial \mathbf{T}}{\partial \boldsymbol{\Delta}}|_{\boldsymbol{\Delta}=0}$ to yield the following Jacobian, which takes the form of a $4 \times 4 \times 6$ tensor:

$$\frac{\partial \mathbf{T}}{\partial \boldsymbol{\Delta}} = \left[\frac{\partial \mathbf{T}}{\partial \delta x}, \frac{\partial \mathbf{T}}{\partial \delta y}, \frac{\partial \mathbf{T}}{\partial \delta z}, \frac{\partial \mathbf{T}}{\partial \delta \theta_r}, \frac{\partial \mathbf{T}}{\partial \delta \theta_p}, \frac{\partial \mathbf{T}}{\partial \delta \theta_y}\right] \tag{2.27}$$

Evaluating each component at zero leads to the following generator matrices:

$$\begin{aligned}
\frac{\partial \mathbf{T}}{\partial \delta x} &= \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \frac{\partial \mathbf{T}}{\partial \delta y} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\
\frac{\partial \mathbf{T}}{\partial \delta z} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \frac{\partial \mathbf{T}}{\partial \delta \theta_r} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \\
\frac{\partial \mathbf{T}}{\partial \delta \theta_p} &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \frac{\partial \mathbf{T}}{\partial \delta \theta_y} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}
\end{aligned} \tag{2.28}$$

Right multiplying by a 3×1 vector $\mathbf{v} = [x, y, z]^T$ produces:

$$\begin{aligned}
\frac{\partial \mathbf{T}}{\partial \boldsymbol{\Delta}} \mathbf{v} &= \left[\frac{\partial \mathbf{T}}{\partial \delta x}, \frac{\partial \mathbf{T}}{\partial \delta y}, \frac{\partial \mathbf{T}}{\partial \delta z}, \frac{\partial \mathbf{T}}{\partial \delta \theta_r}, \frac{\partial \mathbf{T}}{\partial \delta \theta_p}, \frac{\partial \mathbf{T}}{\partial \delta \theta_y} \right] \\
&= \begin{bmatrix} \mathbf{I}^{3\times 3} & [\mathbf{v}]^{\times} \\ \mathbf{0}^{1\times 3} & \mathbf{0}^{1\times 3} \end{bmatrix}
\end{aligned} \tag{2.29}$$

where $[\mathbf{v}]^{\times}$ is the skew-symmetric matrix:

$$[\mathbf{v}]^{\times} = \begin{bmatrix} 0 & z & -y \\ -z & 0 & x \\ y & -x & 0 \end{bmatrix} \tag{2.30}$$

With $\mathbf{v} = \mathbf{T}_{i,j}\mathbf{p}_j^g$ from Equation 2.24, the transform derivative is then:

$$\frac{\partial \mathcal{T}_{i,j}^g}{\partial \mathbf{\Delta}} = \begin{bmatrix} \mathbf{I} & [\mathbf{v}]^\times \\ 0 & 0 \end{bmatrix} \tag{2.31}$$

Robot to Vision Transform Derivative

In this thesis, we use the "aero" co-ordinate frame, which is commonly used in robotics. This frame uses X forward, Y right and Z down. We transform this into the standard computer vision co-ordinate frame, with X right, Y down and Z forward. The rotation matrix $\mathbf{C}_{v,r}$ is defined which will be used to transform between the computer vision co-ordinate frame v and the robotics co-ordinate frame r:

$$\mathbf{C}_{v,r} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \tag{2.32}$$

such that

$$\mathcal{C}(\mathbf{p}_i) = \begin{bmatrix} \mathbf{C}_{v,r} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} \mathbf{p}_i \\ 1 \end{bmatrix} \tag{2.33}$$

The derivative is calculated as:

$$\frac{\partial \mathcal{C}}{\partial \mathbf{p}_i} = \begin{bmatrix} \mathbf{C}_{v,r} & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{2.34}$$

Perspective Projection Derivative

Given a point in the camera frame $\mathbf{p}_c = (x_c, y_c, z_c)$, we can write the projection

function as:

$$\mathcal{K}(\mathbf{p}_c) = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \end{bmatrix} \begin{bmatrix} \frac{x_c}{z_c} \\ \frac{y_c}{z_c} \\ 1 \end{bmatrix} \tag{2.35}$$

To ease the computation of the derivative, we re-formulate $\mathcal{K}$ into two function, $\mathcal{K}_1$ and $\mathcal{K}_2$:

$$\mathcal{K}(\mathbf{p}_c) = \mathcal{K}_2(\mathcal{K}_1(\mathbf{p}_c)) \tag{2.36}$$

where $\mathcal{K}_1$ and $\mathcal{K}_2$ are defined as:

$$\begin{aligned} \mathcal{K}_1(\mathbf{p}_c) &= \begin{bmatrix} x'_c \\ y'_c \end{bmatrix} = \begin{bmatrix} \frac{x_c}{z_c} \\ \frac{y_c}{z_c} \end{bmatrix} \\ \mathcal{K}_2(\mathbf{p}'_c) &= \begin{bmatrix} u_c \\ v_c \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \end{bmatrix} \begin{bmatrix} x'_c \\ y'_c \\ 1 \end{bmatrix} \end{aligned} \tag{2.37}$$

The derivatives of $\mathcal{K}_1$ and $\mathcal{K}_2$ are calculated as:

$$\begin{aligned} \frac{\partial \mathcal{K}_1(\mathbf{p}_c)}{\partial \mathbf{p}_c} &= \begin{bmatrix} \frac{1}{z_c} & 0 & \frac{-x_c}{z_c^2} \\ 0 & \frac{1}{z_c} & \frac{-y_c}{z_c^2} \end{bmatrix} \\ \frac{\partial \mathcal{K}_2(\mathbf{p}'_c)}{\partial \mathbf{p}'_c} &= \begin{bmatrix} f_u & 0 \\ 0 & f_v \end{bmatrix} \end{aligned} \tag{2.38}$$

Using these derivatives, the projection function derivative is:

$$\frac{\partial \mathcal{K}(\mathbf{p}_c)}{\partial \mathbf{p}_c} = \begin{bmatrix} \frac{f_u}{z_c} & 0 & \frac{-f_u x_c}{z_c^2} \\ 0 & \frac{f_v}{z_c} & \frac{-f_v y_c}{z_c^2} \end{bmatrix} \tag{2.39}$$

The estimation techniques presented in this section are used throughout the

remaining chapters of this thesis. However, while the estimation techniques are well-understood, the task of obtaining the underlying data associations remains a challenging task. This theme is explored in the following section.

2.3 Localisation and Mapping

This section provides a review of relevant concepts and state-of-the-art techniques in vision-only localisation and mapping.

2.3.1 Visual Odometry

Visual odometry (VO) is a technique for estimating the robot's motion through the world using a rigidly mounted camera. A solution to the visual odometry problem was presented by Matthies (1989); Nistér et al. (2004, 2006), with specific reference to ground vehicle applications. The term "visual odometry" was chosen because of its likeness to wheel odometry, which allows the robot to estimate its motion by counting wheel revolutions as it traverses the environment. However, unlike wheel odometry, visual odometry is able to estimate the full 6-DOF motion of the vehicle, and is not affected by wheel slip in difficult driving conditions. Visual odometry does not require a motion model for the vehicle's movement through the world, or knowledge of the environment being traversed. The following summarises the visual odometry implementation by Churchill (2012) which is used throughout this thesis. A more comprehensive overview of visual odometry techniques is presented by Scaramuzza and Fraundorfer (2011).

The goal of visual odometry is to determine the transformation $\mathbf{T}_{k,k-1}$ between consecutive images I_k and I_{k-1}, by observing landmarks across both frames. This is illustrated in Figure 2.5 where the transformation $\mathbf{T}_{3,2}$ is computed between images I_2 and I_3.

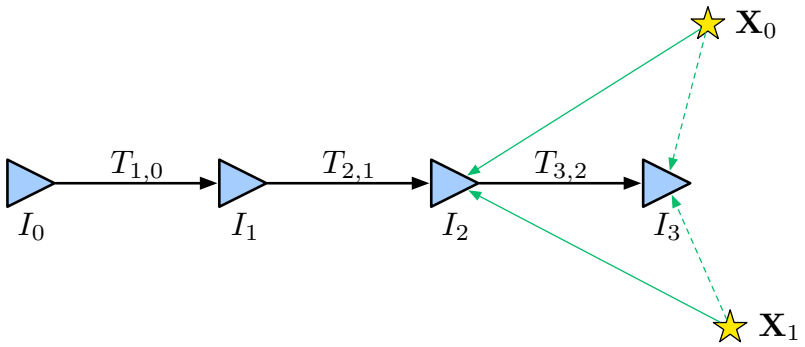


Figure 2.5: Visual odometry is used to calculate the relative motion between camera frames, $\mathbf{T}_{k,k-1}$. The figure shows how landmarks $\mathbf{X}_0$ and $\mathbf{X}_1$ are observed in the images I_2 and I_3 to determine the transformation $\mathbf{T}_{3,2}$ (only two landmarks are shown here for simplicity).

The transformation output from visual odometry is a rigid body transformation:

$$\mathbf{T}_{k,k-1} = \begin{bmatrix} \mathbf{R}_{k,k-1} & \mathbf{t}_{k,k-1} \\ 0 & 1 \end{bmatrix} \tag{2.40}$$

where $\mathbf{R}_{k,k-1} \in \mathbb{SO}(3)$ is the rotation matrix, and $\mathbf{t}_{k,k-1} \in \mathbb{R}^{3\text{x}1}$ is the translation vector. The pipeline for computing the transformation $\mathbf{T}_{k,k-1}$ is described below, and represented graphically in Figure 2.6:

1. **Input images:** An undistorted and rectified stereo image pair I_k is input into the pipeline, with known baseline.

2. **Feature detection:** Feature detection is the process of identifying salient keypoints in the image. A good keypoint is one which is distinctive, corresponds accurately to a landmark in the environment, is computationally efficient to extract from the image, and is observable across multiple images reliably. Our implementation detects FAST corners (Rosten and Drummond, 2006), an example of a point feature detector.

3. **Stereo matching:** A feature is a 2D observation of a 3D landmark in the environment. Stereo matching is the task of associating features in the left

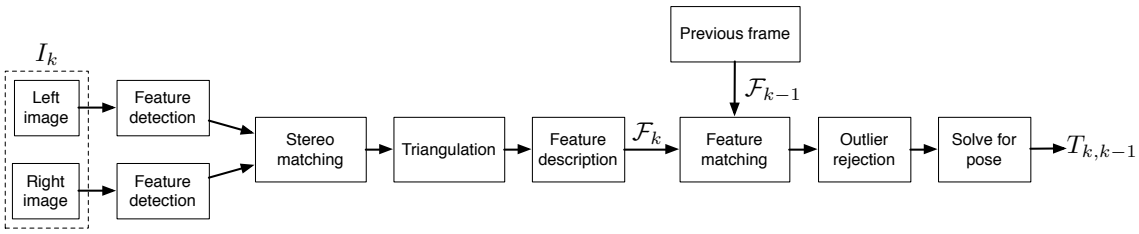


Figure 2.6: Block diagram illustrating the visual odometry pipeline. Features are extracted on a rectified input image I_k. Stereo matching between the left and right stereo images is performed, and the 3D positions of landmarks are determined by triangulation. These landmarks are associated with features extracted in the previous frame $\mathcal{F}_{k-1}$. RANSAC is used to perform robust outlier detection. A non-linear least squares optimisation for pose uses the remaining inlier set to determine the transformation $\mathbf{T}_{k,k-1}$.

image with corresponding features in the right image. Image rectification enforces parallel epipolar lines between the two images. This means that a 3D landmark will project into the same vertical pixel co-ordinate in both images, constraining the search along a single dimension. Given a reference keypoint in the left image, a search is performed along the corresponding row in the right image. For each FAST corner position found in that row, the Sum of Absolute Differences (SAD) is computed with respect to the reference keypoint. A data association is made between the reference keypoint in the left image, and the keypoint in right image with best SAD score. Figure 2.7 shows an example of stereo matching, where FAST corners (shown as red points) are matched between the left and right images (shown with horizontal lines).

4. **Triangulation:** The position of 3D landmarks can be triangulated since the disparity of the observations, stereo baseline, and intrinsic calibration of the cameras are known. The robot's ability to perceive the depth of 3D landmarks at far distances is a function of the stereo baseline, camera focal length, and the resolution of the images. Broadly, a smaller disparity results in a more uncertain estimate of depth.

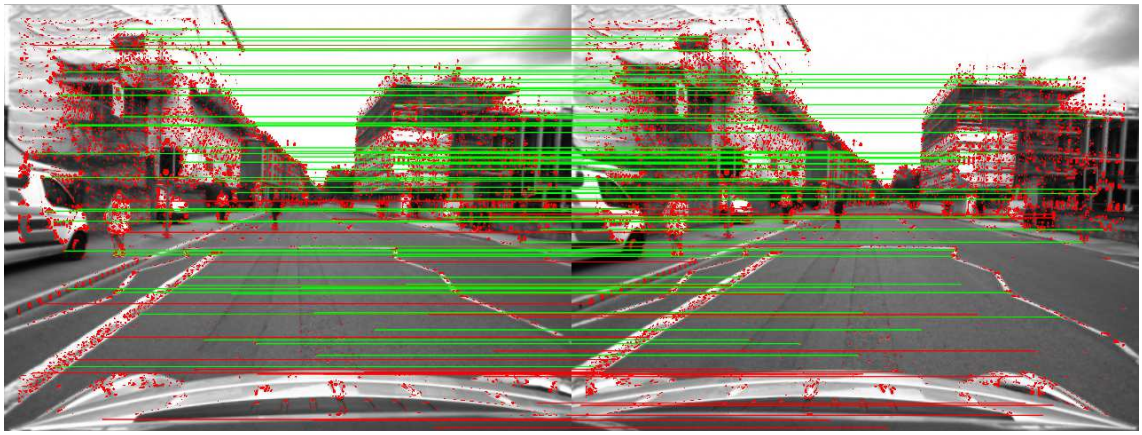


Figure 2.7: Figure illustrating how stereo matches (horizontal lines) are discovered between the left and right images of a stereo pair. FAST corners are detected and shown as red points. Matched features can be triangulated to obtain the 3D position of the landmark.

5. **Feature description:** Landmarks are described using feature descriptors. A feature descriptor is a compact representation of the landmark's appearance in image space. Feature descriptors are used to identify landmarks in subsequent images when the camera motion between the images is unknown. Feature descriptors should be robust to changes in translation, scale and orientation, although in our application of visual odometry the relative pose between consecutive images will be relatively small. Our implementation uses a binary feature descriptor, BRIEF (Calonder et al., 2010), to perform this task.

6. **Feature matching:** Feature matching is the task of associating 2D features observed in the live frame $\mathcal{F}_k$ with 3D landmarks stored in the previous frame $\mathcal{F}_{k-1}$. The matching is performed by comparing a reference feature descriptor against the feature descriptors corresponding to the set of 3D landmarks. A similarity metric is used to compare two feature descriptors. Floating point descriptors such as SURF (Bay et al., 2008) use the Euclidean distance, while binary descriptors such as BRIEF use the Hamming distance. Figure 2.8 illustrates feature matches between consecutive frames.

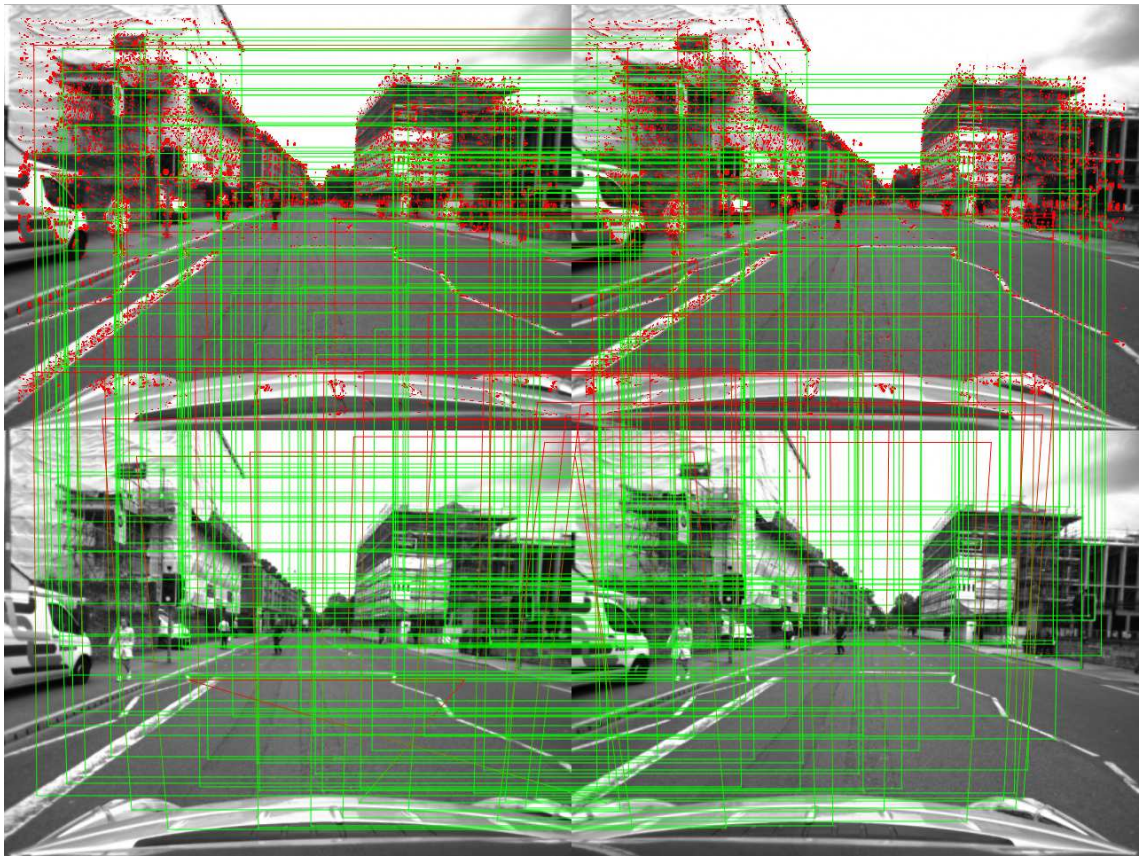


Figure 2.8: Figure showing how the live frame $\mathcal{F}_k$ (top) is matched against the previous frame $\mathcal{F}_{k-1}$ (bottom). Horizontal lines show stereo matching between left and right images, while vertical lines show feature matching between consecutive frames. Red lines indicate outliers, while green lines indicate inliers.

7. **Outlier rejection:** The transformation $\mathbf{T}_{k,k-1}$ can be calculated using the data associations which correspond 2D feature observations to 3D landmark positions. However, even a small number of incorrect data associations can result in large errors in the pose estimate. Incorrect data associations can be caused by image noise, blur, and occlusions, as well as changes in illumination, scale, translation and orientation for which the feature descriptor is not robust. Additionally, feature matches with landmarks corresponding to moving objects such as cars and pedestrians should be rejected, since the optimisation for pose assumes that all landmarks are stationary.

   RANSAC is a method used to perform robust model fitting in the presence of outliers. Within the context of visual odometry, the aim is to determine the transformation $\mathbf{T}_{k,k-1}$ which results in the greatest number of inlier feature observations. Nistér et al. (2004) noted that computing the camera's motion from 3D-2D correspondences resulted in better performance than that of 3D-3D correspondences, since error in the depth of landmarks has a minimal effect on the pose solution when the motion between camera frames is small.

8. **Solve for pose:** The output from RANSAC is a coarse estimate of the transformation $\mathbf{T}_{k,k-1}$. A non-linear least squares optimisation refines this estimate by minimising the re-projection error of all data associations marked as inliers, as discussed in the previous section.

Visual odometry is a fundamental competency in our vision-only localisation system. The following section describes our use of visual odometry to build topometric maps of the environment.

2.3.2 Topometric Maps

Pose estimation, or "metric" localisation, is the process of determining the robot's 6-DOF pose with respect to a prior map. A large body of work in localisation originates within the field of Simultaneous Localisation and Mapping (SLAM) (Dissanayake et al., 2001; Montemerlo et al., 2002; Durrant-Whyte and Bailey, 2006; Klein and Murray, 2007; Sibley et al., 2010b). The SLAM problem is one of mapping an initially unknown environment, while simultaneously localising within that environment. While the estimation problem in SLAM has been solved, real-world implementations face challenges in scaling over large environments (Bailey and Durrant-Whyte, 2006). While there has been much attention given to the problem of reducing the complexity of map updates, we assert that the concept of a globally accurate map is not required to perform autonomous navigation (Sibley et al., 2010a).

The notion of a global map may seem intuitive, since as humans we are accustomed to street maps and satellite imagery. However, these global maps are not required to navigate successfully through the world. For example, when travelling between two cities, a human driver does not require the exact co-ordinates of the route to be driven. Rather, the human driver follows a road which is known to connect the two places. In a similar way, path planning for mobile autonomy requires metric accuracy over small distances, but only topological connectivity over large ones.

This concept is demonstrated in *teach and repeat* autonomy systems. *Teach and repeat* consists of two phases. Firstly, a human operator manually drives the robot through the environment, "teaching" it which route to follow. In the second phase, the robot is able to autonomously repeat the taught route. A number of authors have used this technique to perform autonomous navigation (Furgale and Barfoot, 2010; McManus et al., 2012; Churchill et al., 2015; Paton et al., 2016b; Krusi, 2016). Typically, the map is represented as a series of submaps, where each submap models the

Figure 2.9: A simple map of the environment can be created by recording images as the robot traverses the environment. Visual odometry determines the motion between images, giving the map a topometric structure. At run-time, the localiser attempts to match the live image against one of the images in the map.

local environment and is metrically connected to neighbouring submaps such that the robot can transition between them. Importantly, these submaps are explicitly not aligned in a "global" co-ordinate frame.

In a similar manner, Figure 2.9 shows how a map of images can be used to model the environment, where images are spatially related to one another by local transformations. Section 4.2 describes how the map is represented as a graph structure, where nodes store images, and edges store transformations from visual odometry and loop closures. Since topometric mapping does not perform a global optimisation, augmenting the map is a constant time function. This is fundamental to our ability to localise and map robustly over vast scales.

2.3.3 Pose Estimation

A map can be thought of as a set of spatially indexed images, as demonstrated in Figure 2.9. The robot localises against this map by determining the transformation $\mathbf{T}_{k,q}$ between the live image I_k and an image from the map I_q.

Figure 2.10 illustrates the pipeline for estimating pose in a map of images. Note that during localisation, only a single image is required from the stereo pair. This is because the optimisation for pose minimises the 2D re-projection error of 3D landmark observations. This means that while a stereo camera is required to map the environment (in order to determine the 3D positions of landmarks), localisation could be performed using a monocular camera. In Figure 2.10, only the left camera

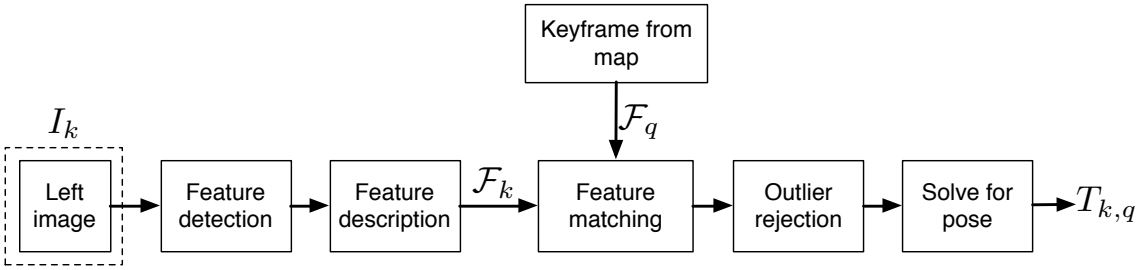


Figure 2.10: Diagram illustrating the pipeline for metric localisation in a map of images. Since a 3D-2D optimisation is performed for pose, only a single live image is required for localisation. Features are extracted from the live image I_k in a similar manner to visual odometry. However, instead of matching $\mathcal{F}_k$ to $\mathcal{F}_{k-1}$, the live frame $\mathcal{F}_k$ is matched to a frame stored in the map $\mathcal{F}_q$. Outliers are rejected and the transformation $\mathbf{T}_{k,q}$ is determined.

image from a stereo camera pair is used.

The remainder of the localisation pipeline is similar to that of visual odometry. To minimise computation, the same features that were extracted during visual odometry could be used for the purposes of localisation; however, we are not constrained in this way. Rather, we find that certain feature types and techniques are better suited to the task of robust localisation, as opposed to that of visual odometry. Chapters 6 and 7 discuss this idea in more depth, presenting new techniques for feature detection, modelling and data association.

As with visual odometry, RANSAC is used to reject outliers and provide a coarse seed for the transformation $\mathbf{T}_{k,q}$. A 3D-2D optimisation for pose refines this coarse estimate to determine the pose of the robot with respect to the map, as described in Section 2.2.

While the pipeline for pose estimation is well understood, the ability to perform robust localisation in outdoor environments remains a challenging problem. Changes in weather, lighting, season and scene structure can drastically alter the appearance of the environment. This means that the live image I_k may not visually resemble the map image I_q, causing the data association step to fail. The following section presents an overview of localisation techniques, with a focus on the problem of

appearance change in large, outdoor environments.

2.3.4 Robust Data Association

Previously, we described how data associations correspond observations in the live image with 3D landmarks stored in a map:

$$(u, v) \leftrightarrow (x, y, z)$$

where (u, v) are the pixel co-ordinates of an observation of a landmark in the world with position (x, y, z). A common approach to data association is the use of point features. Point features identify and describe low-level elements in the image, and usually consist of a feature detector and feature descriptor. The feature detector identifies distinctive keypoints in the environment, while the feature descriptor models the appearance of the keypoint. The descriptors facilitate matching keypoints between images.

A number of feature detectors and descriptors have been presented. Feature detectors can detect different types of low-level image features, for example corners (Rosten and Drummond, 2006; Harris and Stephens, 1988), blobs (Matas et al., 2002), and edges (Canny, 1986). SIFT (Lowe, 2004) and SURF (Bay et al., 2008) are examples of blob detectors which have been used widely due to their properties of scale and illumination invariance. FAST (Rosten and Drummond, 2006) is a corner detector which has also been used widely in applications which do not require scale invariance, due to its low computational cost.

Feature descriptors describe the underlying image patch, or keypoint, which has been detected by the feature detector. Feature descriptors are used to associate observations between images. Therefore, translation, illumination and scale invariance are desirable characteristics of feature descriptors. SIFT and SURF are examples

of floating point descriptors, while a number of binary descriptors have gained in popularity due to their low computation requirements and small data footprint, for example BRIEF (Calonder et al., 2010), ORB (Rublee et al., 2011), BRISK (Leutenegger et al., 2011) and FREAK (Ortiz, 2012).

However, while some feature types claim invariance to illumination, the level of appearance change considered is usually limited to mild indoor settings. Furgale and Barfoot (2010) used SURF features in their visual teach and repeat system, where they performed autonomous control of a rover over 32 km in a planetary analogue environment in the Canadian High Arctic. They noted the lack of localisation robustness to changes in the time of day. A recurring theme in the literature is that while point features such as SIFT and SURF may have some invariance to illumination, they are not able to withstand the extreme changes in appearance encountered in outdoor environments (Valgren and Lilienthal, 2010; Churchill and Newman, 2013; McManus et al., 2015; Paton et al., 2015a).

D-Nets were presented by von Hundelshausen and Sukthankar (2012) as an alternative to conventional keypoint-based approaches. They constructed networks of nodes and edges, where nodes corresponded to traditional keypoint locations and the appearance of the feature was a function of the image content along the edges. They noted increased robustness to translation, scale and reflection, however robustness to changing outdoor environments was not tested.

Churchill and Newman (2013) presented Experience-Based Navigation (EBN) as a technique for performing robust, outdoor localisation using a stereo camera. They introduced the concept of an "experience" as a representation of the world under particular environment conditions. The robot was allowed to build up a map of overlapping experiences to capture the visual change in the environment. By accumulating multiple experiences, the difficult data association problem was avoided since the live image was likely to resemble at least one of the stored experiences in

memory. Experiences were added according to a policy based on localisation failure. This ensured that the map only expanded as much as was required to model the changes in the environment. The authors demonstrated their approach on over 50 traverses of a 700 m loop, showing that localisation performance improved as more experiences were added to the map. This thesis embraces the experience paradigm. Chapter 4 presents a graph-based implementation of the experience framework, while Chapter 5 describes a probabilistic technique which allows the robot to incrementally learn which experiences are most likely to be relevant at a given time.

Mühlfellner et al. (2015) presented Summary Maps as a method for handling appearance change in the world. Unlike the experience-based approach, they merge different appearance conditions into a single, multi-session map. To prevent unbounded growth, the authors describe the process of "summarising" a map as one of selecting a subset of landmarks which represent the full content of the map concisely. Different selection policies are evaluated over a 30 km dataset, collected over a period of 16 months.

In recent work, Paton et al. (2016a) presented an implementation of the Experience-Based framework. Their work focused on the application of *teach and repeat*, where the localisation estimate needs to be referenced against a single, privileged experience which corresponds to a manually driven *teach* pass through the environment. The authors note the importance of "bridging experiences", where in order to reference two vastly different appearances of the world against one another, one needs a series of "bridging experiences" which capture the incremental appearance change. An offline analysis is performed on nine repeats of a 1 km route, and a closed-loop experiment is performed where a robot repeats a 250 m route once per hour, for ten hours.

Robustness to shadows and changing illumination is a common failure mode of vision-based localisation systems in outdoor environments. Shadows introduce harsh

gradients and regions of high contrast into the scene, causing feature methods to break down. Techniques in the field of colour constancy have been used to address this challenge. Colour constancy is the ability to determine the colour of an object independently of external illumination. Since shadows are simply changes in external illumination, a colour constant image is one which does not contain any shadows.

A commonly used transformation is presented by Ratnasingam and Collins (2010). The technique acts on the RGB responses from a camera to generate a single-channel chromaticity image which is invariant to changes in illumination. The technique requires that the spectral responses of an RGB camera are assumed to be infinitely narrow, and that the source of illumination (i.e. the sun in outdoor environments) is modelled as a black body illuminator. This technique is discussed in more detail at the end of this section, while Foster (2011) presents an overview of other colour constancy techniques.

A number of works have used the transformation presented by Ratnasingam and Collins (2010) to perform robust localisation. McManus et al. (2014) presented a localisation system which operated on two parallel image streams, one grayscale and the other illumination invariant. Localisation was performed independently in each stream using BRIEF features. They concluded that the combined system offered improved localisation robustness, for only a small cost in additional processing.

Maddern et al. (2014b) presented localisation performance over a 24-hour period using the illumination invariant transform. Localisation with illumination invariant images performed poorly at night due to the violated assumption of a black body illuminator. However, they noted that at night the illumination conditions are fairly constant, resulting in consistent performance using grayscale images.

Paton et al. (2015a) used two illumination invariant transformations in parallel, a technique the authors refer to as "multi-channel localisation". However, instead of using the calibration parameters for the spectral response from the camera's

datasheet, they tuned these parameters for performance in different environments. One illumination invariant image is tuned for performance in environments with trees and bushes, while another illumination invariant image is tuned for performance in rocky areas. They showed increased localisation robustness while performing 26 autonomous repeats of a 1 km route in an outdoor environment.

In environments where appearance change is viewpoint dependent, a multi-camera system may offer additional robustness. Paton et al. (2015b) used two cameras to perform visual teach and repeat, where the cameras faced in opposite directions. The authors note how this increases the system's effective field of view, increasing the probability of finding stable features in the environment for localisation. The system was used to perform autonomous navigation of a 250 m loop six times, with the environment's appearance subject to melting snow and changing lighting conditions.

In a recent field of work, semantic localisation attempts to detect semantically meaningful objects in the environment for localisation. For example, a traffic light detector could be trained offline using large databases of training samples under varying environmental conditions. If a map were then annotated with the locations of traffic lights, at run-time these objects could be detected to perform robust localisation. This approach also speaks to the use of higher-level visual features for localisation which are less prone to failure under changes in appearance.

Salas-Moreno et al. (2013) presented SLAM++, a localisation and mapping technique which used objects such as chairs and tables for localisation. However, the method required detailed 3D models of the objects used for localisation, which may not be available in our scenario. Atanasov et al. (2016) also performed localisation using semantically meaningful objects. The detections were modelled using random finite sets, which allowed them to incorporate missed detections and false positives into the sensor model. Semantic techniques demonstrate the value of incorporat-

ing higher-level information into the localisation problem, however one would likely need to detect a very large number of different objects to gain enough information for continuous localisation in outdoor environments.

In response to the short-comings of point feature approaches, McManus et al. (2015) presented Scene Signatures, a robust method for pose estimation. Using multiple traverses through the environment, the authors trained robust classifiers to detect landmarks at run-time for localisation. Their work was inspired by Doersch et al. (2012), whose method distinguished between images of Paris and London by learning distinctive mid-level classifiers for each city. McManus et al. (2015) presented localisation results in challenging outdoor environments, across different times of the day and night, and in a number of challenging weather conditions.

Kendall et al. (2015) presented PoseNet, a convolutional neural net which is trained to regress the camera's orientation and pose from images. The system is able to tolerate large translations in camera pose, and has a degree of invariance to illumination and motion blur. Pose estimates using this technique are obtained to within 2 m and 3 degrees of error. The authors note that there is a limit to the size of the physical area which a finite neural net can map, but do not explore this limit in this work.

Pose estimation under appearance change has presented an active topic of research for a number of years. Researchers have approached the problem from a wide range of perspectives, attesting to the fact that no one solution has been proven to work under all conditions and in all environments. While significant advances have been made, it remains a difficult and unsolved problem.

2.3.5 Topological Techniques

So far, this section has described pose estimation as a localisation task which outputs 6-DOF metric pose with respect to a particular co-ordinate frame. Here, we

introduce "place recognition" as another type of localisation. However, unlike pose estimation, it only estimates the robot's *topological location* within the map. Place recognition outputs the place in the map nearest to the robot – it says nothing about the position or orientation with respect to that place. This topological estimate is used to seed the pose estimation techniques described later in this thesis. The interface between these two subsystems is described in more detail in Section 4.3, while a brief overview of topological localisation techniques is presented below.

Cummins and Newman (2008) presented FAB-MAP, an appearance-based topological localiser. FAB-MAP describes the appearance of a scene using a bag of words representation (Sivic et al., 2003), where a set of "words" are sampled from a "visual vocabulary". The original FAB-MAP implementation used SURF features (Bay et al., 2008) to generate the visual vocabulary and build the bag of words representation. For each location in the map, FAB-MAP learns a generative model for the obervations of appearance words at that place. Importantly, the generative model captures how observations of words are correlated, and the approach was shown to significantly outperform a Naive Bayes assumption of independence. At run-time, given a live image from the camera for localiastion, the system generates the probability distribution $p(L_i|\mathcal{Z}^k)$ which describes the probability of the location of the live image k. FAB-MAP was tested on a 1000 km car journey (Cummins and Newman, 2010), demonstrating how the technique scales over large distances. In recent work, MacTavish et al. (2016) used the illumination invariant transformation within the FAB-MAP framework to improve robustness under varying lighting conditions.

Milford and Wyeth (2012) presented SeqSLAM, a topological localiser which uses a whole image descriptor to perform localisation. They noted that FAB-MAP performed poorly under extreme changes in appearance, likely due to its use of sparse SURF features. Instead of using sparse features, SeqSLAM uses a simple whole

image descriptor. By exploiting the sequence information in the camera imagery as a vehicle drives through the environment, the authors were able to demonstrate notable robustness to changes in lighting and weather, including between day and night.

Suenderhauf et al. (2015) presented a topological localiser using ConvNet features (Krizhevsky et al., 2012). They noted that place recognition techniques which utilise a whole-image descriptor (such as SeqSLAM) lacked robustness to translation and viewpoint change. While FAB-MAP has some robustness to viewpoint change, it is less robust to appearance change. The authors use ConvNet features due to their greater invariance to both viewpoint and appearance change.

2.3.6 Illumination Invariance

As described previously, the objective of the illumination invariant transformation is to generate an "illumination invariant" image. An image which is not a function of external illumination implicitly does not contain shadows. The previous section described how various localisation systems have used an illumination invariant transformation to improve robustness to shadows. This thesis also makes use of the illumination invariant transformation presented by Ratnasingam and Collins (2010), so the method is summarised here.

The following relationship defines the response of an image sensor R with spectral sensitivity $F(\lambda)$:

$$R^{x,E} = \mathbf{a}^x \cdot \mathbf{n}^x I \int S^x(\lambda) E(\lambda) F(\lambda) d\lambda \tag{2.41}$$

where I is the intensity of the illuminator, $E(\lambda)$ is the illuminator's emitted spectral power density, and $S^x(\lambda)$ is the reflectivity of the scene at point x. The dot product $\mathbf{a}^x \cdot \mathbf{n}^x$ models the geometry of the scene, where $\mathbf{a}^x$ is the unit vector representing the direction of the light source, and $\mathbf{n}^x$ is the unit normal of the surface. These

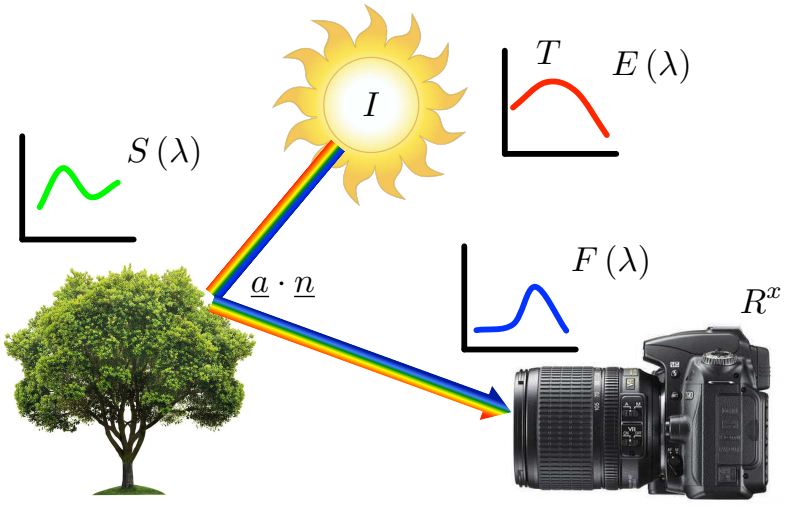


Figure 2.11: Diagram illustrating how an illuminated outdoor scene is observed by a camera. During daylight hours, the sun illuminates the scene with illumination I and spectral power density $E(\lambda)$. The illumination invariant transformation models the sun as a black body illuminator with unknown correlated colour temperature T. Objects in the scene have unknown reflectivity $S^x(\lambda)$, while the dot product $\mathbf{a}^x \cdot \mathbf{n}^x$ models the geometry of the scene at point x. The image sensor R^x has spectral sensitivity $F(\lambda)$. The illumination invariant transformation generates an image which is independent of external lighting conditions. This transformation is used to remove shadows from camera images. Image credit: Maddern et al. (2014a).

parameters are visualised graphically in Figure 2.11.

This expression can be simplified for a photodetector which is only sensitive to light at a particular wavelength λ_i, where the spectral sensitivity $F(\lambda)$ is assumed to be infinitely narrow, such that it can be modelled using a Dirac delta function:

$$R^{x,E} = \mathbf{a}^x \cdot \mathbf{n}^x I S^x(\lambda_i) E(\lambda_i) \tag{2.42}$$

Taking the log of both sides of this equation yields:

$$\log(R^{x,E}) = \log\{\mathbf{a}^x \cdot \mathbf{n}^x I\} + log\{S^x(\lambda_i)\} + \log\{E(\lambda_i)\} \tag{2.43}$$

Ratnasingam and Collins (2010) show that by assuming the source of light is from a black body illuminator with unknown correlated colour temperature T, this

reduces to:

$$\log(R_i) = \underbrace{\log(GI)}_{(a)} + \underbrace{\log(2hc^2\lambda_i^{-5}S_i)}_{(b)} - \underbrace{\frac{hc/k_B}{T\lambda_i}}_{(c)} \tag{2.44}$$

where h is Planck's constant, k_B is the Boltzmann constant, and c is the speed of light. The equation in this form illustrates the (a) wavelength-independent component, (b) the surface reflectivity component, and (c) a component that depends on the source of the illumination.

Components (a) and (c) can be removed to create an illumination invariant feature space F_1 by performing the log-difference between one detector's response, and the weighted sum of two other detectors:

$$F_1 = \log(R_2) - \{\alpha \log(R_1) + (1-\alpha)\log(R_3)\} \tag{2.45}$$

The feature space F_1 is invariant to illumination if:

$$\frac{1}{\lambda_2} = \frac{\alpha}{\lambda_1} + \frac{1-\alpha}{\lambda_3} \tag{2.46}$$

where λ_1, λ_2 and λ_3 correspond to the peak wavelength responses of the camera's Bayer filter. This information can be found in the camera's dataset, allowing us to calculate the parameter α which is unique to each camera. In our applications, the camera being used for navigation is always known, so the parameter α can simply be provided with the sensor calibration model.

Chapter 7 describes how the illumination invariant transformation is used within the broader localisation system to improve robustness to shadows.

2.4 Summary

This section has presented the elemental theory and concepts for camera-based localisation and mapping. Homogeneous co-ordinates in projective space $\mathbb{P}^2$ and $\mathbb{P}^3$ were discussed as an alternative representation to their Euclidean counterparts, $\mathbb{R}^2$ and $\mathbb{R}^3$, respectively. Rigid body transformations in $\mathbb{SE}(3)$ were presented as homogeneous 4×4 matrices which acted on homogeneous co-ordinates to transform them between right-handed, orthogonal reference frames. Additionally, a camera was modelled by a projective function $\mathbf{P}$ which operates on a homogeneous co-ordinate in $\mathbb{P}^3$ to project it into the sensor plane in $\mathbb{P}^2$.

Visual odometry was presented as a technique for incrementally determining the ego-motion of an agent. This introduced the concepts of feature detectors and feature descriptors, as well as methods for outlier rejection and pose estimation from point correspondences.

Finally, a review of state-of-the-art localisation and mapping techniques were presented. This included the use of a topometric mapping framework, where the map is permitted to exist in a relative space which does not globally align in a single "world" frame. Pose estimation techniques were presented, with the similarity to existing visual odometry techniques noted. The challenge of pose estimation was presented as that of determining data associations under the presence of extreme appearance change in outdoor environments. A review of the recent work in this field was presented, including the experience-based paradigm, illumination invariance and the notion of higher-level visual features for localisation. We note that while significant gains in vision-only pose estimation have been made in recent years, the problem remains unsolved. Importantly, we also note that to our knowledge, no previous work has attempted vision-only pose estimation over the large distances (1500 km, cross-validated) considered in this thesis.

The remaining chapters describe our contributions in the field of vision-only localisation, culminating in the presentation of a new, camera-only localisation system which exhibits significant robustness to appearance change.

Chapter 3

Vast-Scale Evaluation: Datasets & Metrics

3.1 1500 km of Outdoor Driving Data

This chapter describes the data used in the design, testing and validation of our localisation system. In total, we test on over 1500 km of data from five different datasets. The algorithms presented in this thesis are motivated by this vast-scale data collection effort, and the continual testing and validation on new data has provided a forcing function in the handling of corner cases and rare events. Table 3.1 presents a summary of the data collected, while Figure 3.1 shows images from the various datasets.

In all the datasets presented below, data was collected while the vehicle was manually driven. Drivers were instructed to follow a particular route, but were not constrained on how to position the vehicle in the lane, or which lane to select when multiple lanes were available. Since many vision-only localisation techniques are viewpoint dependent, this lateral deviation across the road makes the localisation problem significantly more difficult. However, since autonomous vehicles are ex-

	Central Oxford	Milton Keynes	Cornbury Off-Road	Bicycle	Begbroke
Route Length	10 km	10 km	10 km	2.2 km	0.7 km
Number of Repeats	100	20	20	25	107
Total Distance	1000 km	200 km	200 km	56 km	75 km
Sensor	Bumblebee XB3	Bumblebee XB3 & Widebaseline Pair	Bumblebee XB3	Bumblebee BB2	Bumblebee XB3

Table 3.1: Validation and testing is performed on over 1500 km of data from five challenging datasets. All of the five datasets are captured in large outdoor environments, with the Central Oxford dataset containing the most diverse set of environmental conditions including seasonal change, large-scale structural change, and harsh weather and lighting conditions.

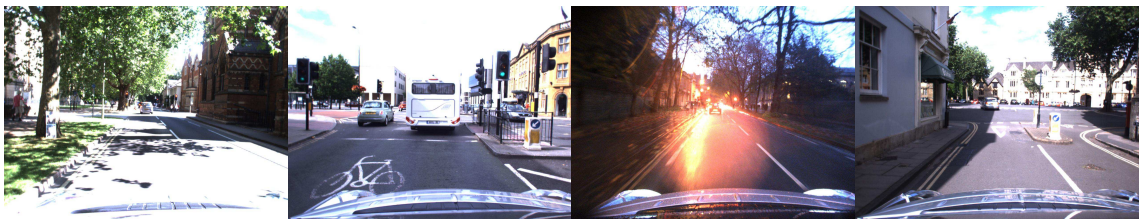
pected to identify and navigate around obstacles, we believe translation invariance to be an important characteristic of a localisation system.

We also note the effect that camera settings can have on the appearance of an image. While the same camera is used throughout data collection for a particular dataset, we have not kept the camera settings constant. In particular, the camera's auto exposure controller has changed several times over the period of data collection. The changing auto-exposure behaviour introduces another form of appearance change, since scenes may not be consistently exposed. However, we note that the magnitude of appearance change caused by this is relatively minor compared with the appearance change caused by weather and lighting.

The collection and management of large datasets has required a sustained commitment from a number of members of the Mobile Robotics Group. This thesis acknowledges the contributions from the Hardware, Platform, and Software Engineering teams for building and maintaining the systems, the Trial Support team for handling the logistics of public trials, and the students and researchers who assisted with systems integration and data collection.

3.1.1 Oxford Dataset

The Central Oxford dataset was collected over a period of 18 months using the Oxford RobotCar, a modified Nissan LEAF. The dataset consists of 100 repeats of a 10 km route through busy central Oxford, totalling over 1000 km of logged data.



(a) Central Oxford

(b) Milton Keynes

(c) Cornbury Off-Road

(d) Bicycle Dataset

(e) Begbroke

Figure 3.1: Sample images from the five datasets used in the development of the localisation system. Data was collected over three years, in all weather and lighting conditions, at all times of the day, in busy city centres and rural off-road environments.

Figure 3.2: Significant road works were encountered over the data collection period. Each of the above images is from the same place (note the building on the left is visible in all images). In this example, the road network was altered to change the flow of traffic. Our system automatically updates its map representation when it detects change in the environment, including structural change.

While other datasets may be of similar size or greater, this dataset contains by far the greatest number of repeats of a single route. A public release of this dataset was made earlier this year (Maddern et al., 2017).

The dataset is exposed to an extreme variety of environmental conditions. Previously, Figure 1.1 illustrated this diversity by showing one image from each of the 100 repeats of the 10 km loop, at exactly the same place. The figure shows the effects of seasonal change over the 18 month period, as well as how changes in lighting and weather can significantly alter the appearance of a scene. Additionally, the dataset was also affected by large-scale construction works in a number of areas of the map. Figure 3.2 illustrates one particular construction works where the road network was physically altered. This highlights the importance of systems which can adapt to their environment, since a static world assumption does not hold over long periods of time in outdoor environments.

3.1.2 Milton Keynes Dataset

The Milton Keynes dataset was collected in preparation for the LUTZ PathFinder Project, a project which aims to transport members of the public between popular destinations in Milton Keynes. The pods travel mainly on shared pedestrian and cycle paths over a 10 km route. The dataset presented here contains 20 repeats of

(a) 2016-04-22 **(b)** 2016-05-26 **(c)** 2016-06-01

Figure 3.3: The Cornbury Off-Road dataset takes place in a predominantly forested environment. This figure illustrates the dramatic effect of seasonal change on scene appearance as late winter turns to summer. Each image is captioned with the date the log was collected.

the route, captured over six months.

This dataset contains data captured using two Grasshopper cameras mounted in a rigid wide-baseline configuration (a baseline of 65 cm). The system cost of using two monocular cameras is significantly cheaper than using a single stereo camera, and the larger baseline allows the localisation system to triangulate for the position of landmarks further in the distance.

3.1.3 Cornbury Off-Road Dataset

The Cornbury Off-Road dataset is set in a private estate in North Oxfordshire. Unlike the other urban datasets presented here, this dataset consists of a 10 km route in an off-road, forested environment. This route does not contain man-made structure such as buildings and roads, meaning that the localisation system must use natural landmarks for localisation such as trees, bushes and rocks. These natural landmarks are significantly more susceptible to changes in environmental conditions, in particular seasonal change as shown in Figure 3.3.

3.1.4 Begbroke Dataset

The Begbroke dataset consists of 75 km of driving around a 0.7 km loop of the Begbroke Science Park in North Oxford. The route is traversed using the Oxford RobotCar at hourly intervals throughout the day, between 7am and 7pm, over a period of two months. This dataset was used prior to collecting the central Oxford dataset.

3.1.5 Keble Bicycle Dataset

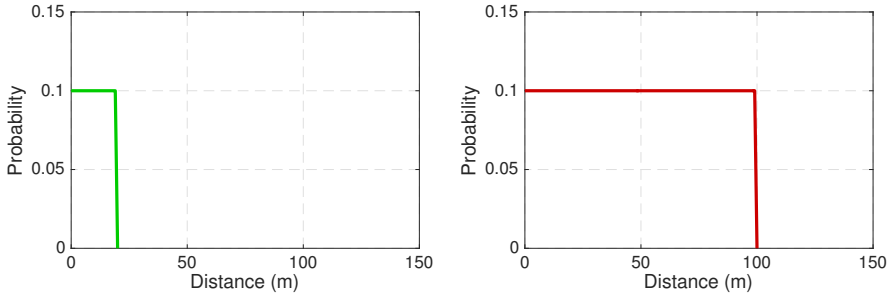
The Keble Bicycle dataset consists of 56 km of data captured using a Bumblebee BB2 stereo camera mounted to the handlebars of a bicycle. The route traverses bicycle paths and pavements through a busy urban area. The 2.2 km route was traversed regularly over a period of three months and contains harsh lighting conditions, changes in weather and occlusion of the camera by pedestrians and cars.

3.2 Performance Metrics

Performance metrics are an important concern when processing data over vast scales. In particular, we require an unsupervised metric which is able to (i) summarise localisation performance over thousands of kilometres, (ii) highlight problem areas where the system failed, and (iii) make predictions about future performance under real-world conditions. This section describes our approach to this problem.

3.2.1 Probability of Localisation Failure

A simple metric for localisation performance might be to calculate the percentage of successfully localised frames in a particular experiment. However, this metric does not take into account how the localisation system will be used on an autonomous



(a) Five 20 m failures **(b)** One 100 m failure

Figure 3.4: Example scenario demonstrating our key performance metric. Two vehicles travel 1 km. The first vehicle loses localisation five times, for a 20 m interval each time. The second vehicle experiences a single localisation failure for 100 m. Both vehicles have failed to localise for 10% of the total distance, however the first vehicle's performance is significantly better since it has relied on visual odometry for much shorter intervals than the second vehicle.

vehicle in closed loop. Recall that during localisation failure, the robot can integrate visual odometry to update its position and orientation in the map. For example, if the localisation system failed to localise on every second frame from the camera, the autonomous vehicle would continue to operate normally by integrating visual odometry during these periods of temporary failure. A metric which only considers the percentage of frames localised would reflect poorly on the localiser's performance (i.e. only 50% of frames localised), even though the autonomous vehicle operated successfully. Note that we do not use precision-recall curves for the same reason, since they only consider the percentage of frames localised as the measure of success. Additionally, precision-recall curves rely on adjusting some underlying tuning parameter or system threshold, neither of which exist in this localisation system.

However, visual odometry is only accurate over short distances, and the open loop estimate exhibits drift over large distances. Our performance metric captures this by measuring the probability of the vehicle having to drive a certain distance on visual odometry, without successful localisation.

For example, consider the two graphs in Figure 3.4. In this example, two vehicles

each travel 1 km. In Figure 3.4a, the vehicle lost localisation five times, travelling 20 m each time before re-localising in the graph. In Figure 3.4b, the vehicle suffered a single localisation failure of 100 m. Both vehicles failed to localise for 10% of the route, however the vehicle in the second scenario performs significantly worse since it must drive for a longer distance on visual odometry.

As discussed above, a localisation failure occurs when the vehicle is unable to determine its 6-DOF pose with respect to the map. We describe the localisation failure by the distance travelled d m before the vehicle is able to re-localise. The probability of travelling a distance greater than X m without successful localisation is calculated as the ratio of the sum of localisation failures greater than X m, to the total distance y travelled during testing:

$$p(x > X) = \frac{\sum_{d_i > X} (d_i)}{y}$$

where $D = \{d_0, d_1, ..., d_i\}$ is the set of distances for which localisation failed during testing. This naturally extends to multiple localisation runs over many thousands of kilometres, since each localisation run simply appends its set of localisation failures to D, and increases the total distance travelled y. If the testing data is representative of real-world conditions, this distribution can be interpreted as the probability of localisation failure over a certain distance.

Localisation failure is detected automatically when the optimisation for pose (i) fails to converge, (ii) converges with a high inlier RMS error, (iii) converges with a small number of inliers, or (iv) the sequential localisation estimates do not correspond to the relative motion estimates from visual odometry. We find that this is an intuitive and natural method for assessing localisation performance.

3.2.2 Ground Truth

The probability of localisation failure is a measure of robustness, but does not measure localisation accuracy. Localisation accuracy is an important, but difficult question to answer. Even a state-of-the-art GPS+INS system does not output consistently accurate pose, and has been observed to drift in the order of metres over time (Levinson and Thrun, 2010; Maddern et al., 2015). These GPS systems rely on an unobstructed signal path between the vehicle receiver and the constellation of satellites above. Tall buildings, tree cover, and tunnels can obstruct the signal path. Additionally, since a GPS signal is an electromagnetic wave, it is subject to reflection and refraction. This can result in an effect called multi-path error, where the signal reaches the receiver after being reflected off another surface. This results in poor localisation estimates from the GPS system. Figure 3.5 plots the localisation estimates of a high-end GPS+INS system on an overhead map. The vehicle drove the same 10 km route in all cases, but the GPS failed to reproduce the trajectory accurately.

In Chapter 7, we obtain accurate ground truth by performing global pose graph optimisation. This is a time-consuming process which requires manual inspection of loop closures and hand-tuning of constraints. We perform a ground truth analysis on 100 km of data from the Central Oxford dataset, but we note that our key performance metric remains the probability of localisation failure as described above, due to its ability to scale over large experiments.

3.2.3 Cross-Validation

In a number of our experiments, we perform cross-validation to better estimate the system's real-world performance. Cross-validation prevents bias towards a particular mapping or localisation sequence which yields favourable results. Additionally, by

(a) Success **(b)** Failure

(c) Failure **(d)** Failure

Figure 3.5: Figure illustrating the unreliable performance of a high-end GPS+INS system while repeating the same 10 km route through central Oxford. Figure (a) shows the correct trajectory, when GPS performed well. Figure (b) exhibits a subtle error on the right-hand side of the map, where the trajectory drifts by several metres. Figures (c) and (d) demonstrate gross localisation errors, in the order of kilometres.

using the data in different combinations and orderings we are able to discover more corner cases and rare events.

Cross-validation is performed as follows. Consider the scenario where N logs are available, where a log corresponds to a single traverse of the route. In an experiment with m folds, m independent maps are created. For each map:

1. Sample the set of N logs to draw e experiences (without replacement) which will be used during mapping.

2. The remaining $l = N - e$ logs are used to test localisation against that map.

In total, ml localisation runs will be performed. If the distance of the route is x m, the total distance travelled during localisation is mlx m. We note that this procedure for cross-validation differs slightly from the traditional k-fold cross-validation. This is because the number of experiences e used in mapping is a variable we want to control in our experiments, so we cannot simply divide the available logs into k groups in order to perform conventional k-fold cross-validation.

3.3 Summary

This section has presented five challenging datasets, totalling over 1500 km of collected data. Data has been collected over a period of three years under harsh sun, rain, snow, at all times of the day and night, in all seasons, and in urban and off-road environments. The largest of these datasets is the 1000 km Oxford dataset, consisting of 100 repeats of a 10 km route through central Oxford over a period of 18 months. The dataset is unique in the high number of repeats, as well as the diverse range of extreme appearance change conditions encountered. This dataset was made available for public download earlier this year.

Additionally, we have presented performance metrics which will be used to evaluate the localisation system. We described the importance of assessing localisation

performance within the context of how the system will be used, presenting our key performance metric as the probability of localisation failure over a particular distance. We discuss our approach to achieving ground truth localisation results, and demonstrate some of the challenges of obtaining ground truth from even high-end GPS+INS systems. Finally, we discuss our use of cross-validation as an essential tool to prevent bias towards a particular mapping or localisation sequence which produces optimal results. These techniques are used extensively throughout the remaining chapters as we begin to describe the algorithms, implementation and testing of the various system components.

Chapter 4

Experience-Based Navigation

4.1 Introduction

This thesis is about life-long, vast-scale localisation in challenging outdoor environments. Outdoor, camera-only localisation is difficult because of complex changes in the environment's appearance, such as those caused by changes in lighting, weather, season and scene structure. The underlying problem is that of data association: How should the robot associate landmarks in the map with observations in the live camera image, in spite of this extreme appearance change?

We approach this problem through the paradigm of "experiences". Pedagogically, an experience can be thought of as a visual memory. It is a single representation, or snapshot, of the world under particular conditions. The robot gathers many of these overlapping experiences as it navigates through the world, storing them in its map. In so doing, the robot is able to model the environment using multiple, independent representations under different conditions. For example, Figure 4.1 demonstrates that if the robot traverses the same environment three times under three different conditions, then the map may contain three distinct visual memories of how the world looked under those different conditions.

Figure 4.1: An experience is a stream of images recorded as the robot drives through the environment under particular conditions. Multiple experiences model the appearance of the world under different conditions. Three experiences are shown here, (i) a cloudy, winter's day, (ii) a sunny summer's day, and (iii) a rainy winter's evening. Note that the underlying cause of appearance change is not important, but only that the appearance of the environment has changed.

At run-time, the localiser need only localise in a single experience to remain localised in the map. Since the map is able to model a much wider spectrum of appearance change than a single-experience map, the localiser is more likely to find an experience which closely resembles the live image from the camera. In essence, we are explicitly avoiding the difficult data association problem. By accumulating multiple experiences in the map, localisation is robust to cyclic appearance change (diurnal lighting, seasonal changes, and extreme weather conditions) and is also able to adapt to slow structural change. A "policy" determines when new experiences are saved to the map, so that only the minimum number of experiences are stored. This policy is based on localisation failure; we assert that localisation failure occurs when the map does not sufficiently model the world's current appearance condition. When localisation failure occurs, a new experience is saved to the map to represent the new appearance conditions. Over time, localisation performance improves and the number of experiences stored in the map tends towards a constant.

This approach to robust localisation is referred to as Experience-Based Navigation and was originally presented by Churchill and Newman (2013). While this thesis makes extensive use of the experience paradigm, its implementation of the ex-

perience framework is different to that of Churchill and Newman (2013). A subtle, but important, example of this is in how the map is used for localisation. Churchill and Newman (2013) describe using N localisers in parallel, where each localiser operates on its own independent experience. In contrast, our implementation uses a single localiser which operates on a multi-experience graph structure, dynamically sampling from different experiences for localisation as required. This chapter describes the framework and interfaces of a robust, graph-based implementation of Experience-Based Navigation as it is used over the following chapters.

4.2 The Experience Graph

A core feature of our work is the use of multiple experiences to model change in the environment. Experiences stored in the map are independent in the sense that different experiences are not merged. We assert that merging experiences is undesirable. For example, merging two experiences from summer and winter would not produce an experience equivalent to autumn or spring, since the features observed across different seasons are fundamentally different. Attempting to merge multiple experiences would lead to questionable data associations, with little performance gain to be achieved. Rather, we maintain independent experiences but actively seek loop closures between different experiences when possible. This section describes how these multiple experiences are represented as a graph structure, which we refer to as the "experience graph". An illustration of the experience graph is shown in Figure 4.2.

4.2.1 Building an Experience

An experience is a representation of the environment under particular conditions. Consider a robot equipped with a stereo camera, capturing images $\mathbf{I} = \{I_0, I_1, ..., I_\alpha\}$

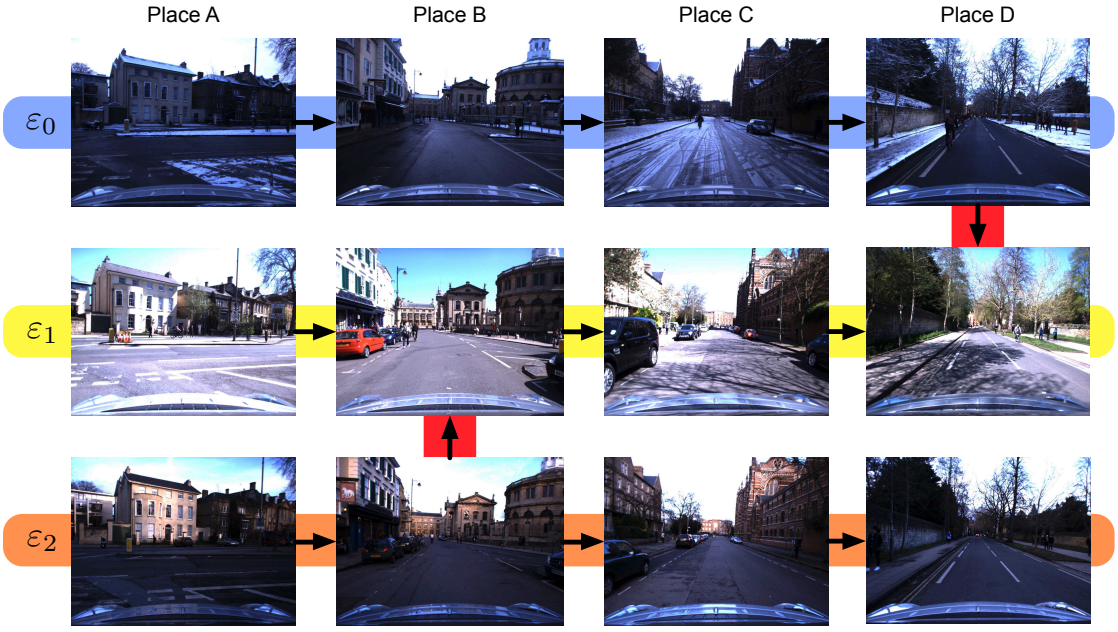
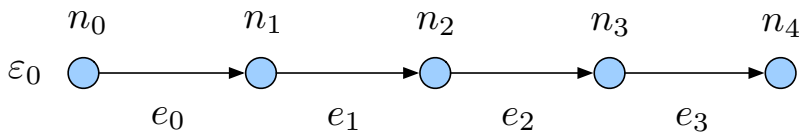


Figure 4.2: An experience graph is shown with three experiences, where an experience is a sequence of images modelling the world under particular conditions. The experiences ε_0, ε_1, and ε_2, model the world under snow, mid-day sun, and late afternoon sun, respectively. Loop closures (shown in red) relate images of different experiences.

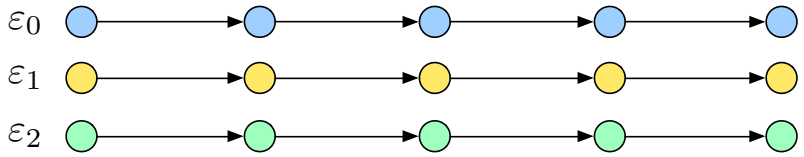
as it traverses through the world. Visual odometry was discussed in Chapter 2 as a system which operates on a sequence of stereo images to estimate the camera's motion through the world. The camera's motion is represented by a set of corresponding relative transformations $\mathbf{T} = \{\mathbf{T}_{1,0}, \mathbf{T}_{2,1}, ..., \mathbf{T}_{\alpha,\alpha-1}\}$.

The stream of images $\mathbf{I}$ and relative transformations $\mathbf{T}$ are used to create an experience. An experience ε_i is defined as a directed linear graph, consisting of a sequence of nodes $\varepsilon_i = \{n_0, n_1, ..., n_\alpha\}$ connected by $\alpha-1$ edges $\{(n_0, n_1), (n_1, n_2), ..., (n_{\alpha-1}, n_\alpha)\}$. This is shown graphically in Figure 4.3a. Nodes model the appearance of the world and are appended with the corresponding images in $\mathbf{I}$, as well as the 3D landmarks which will be used for localisation. Edges connect nodes and are appended with 6-DOF transformations from visual odometry, giving the graph a topometric structure.

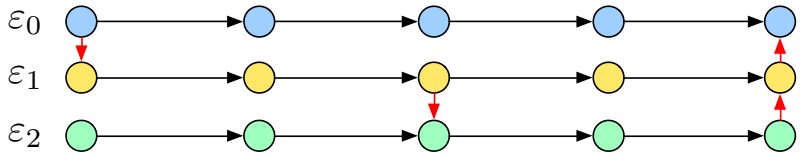
Note that since visual odometry provides the relative motion between images



(a) An experience is a linear subgraph, where nodes model the appearance of the world, and edges contain transformations from odometry.



(b) Multiple experiences are added to the experience graph $\mathcal{G}$, however these subgraphs are disconnected since no loop closures exist to associate nodes across different experiences.



(c) Loop closures (shown in red) increase the connectivity of the graph, allowing the localiser to query the graph $\mathcal{G}$ for all nodes in a particular place.

Figure 4.3: Illustration of an experience graph as a collection of multiple, connected experiences. Experiences allow the robot to model appearance change in the environment, while loop closures increase graph connectivity and facilitate graph searches.

I, these images can be downsampled on distance such that only one image exists for every x m travelled. This ensures an even distribution of data throughout the experience regardless of camera capture rate or vehicle speed, and prevents the map size from increasing when the vehicle is stationary.

Multiple linear experiences can be stored in a single graph structure, as shown in Figure 4.3b. We refer to the complete graph structure as the experience graph $\mathcal{G}$. A single experience ε_i is a linear subgraph of $\mathcal{G}$, and each experience models the world under different appearance conditions. The experience graph $\mathcal{G}$ forms the map against which localisation is performed. It is stored in a SQLite database[1], where a graph-based interface is used to query the data (Nelson et al., 2015b).

4.2.2 Adding Loop Closures

So far, we have described how multiple experiences are added to the experience graph $\mathcal{G}$. However, we have not discussed how these experiences are linked. We use the term "loop closure" to refer to an external relative pose constraint which links two nodes[2]. Loop closures can be added between nodes of the same experience, or nodes of different experiences, if an external localiser is able to localise one node against another. Loop closures indicate that two nodes are in the same place topologically, although we additionally require the 6-DOF relative transformation between the two nodes. Note that the problem of determining loop closures is a specialism of the more general localisation problem, where instead of localising the live image to a map image, localisation is performed between two map images.

A loop closure is represented as an edge between two nodes (n_s, n_d). In implementation, the edge is identical to the edges created using visual odometry constraints,

[1] https://www.sqlite.org/

[2] A more conventional use of this term would be to indicate the event that the robot has returned to the starting location, thus "closing the loop" after a mapping expedition. We assert that the loop closure event is simply the event that the robot visits a place it has already stored in memory, regardless of where this location occurs along the mapping trajectory.

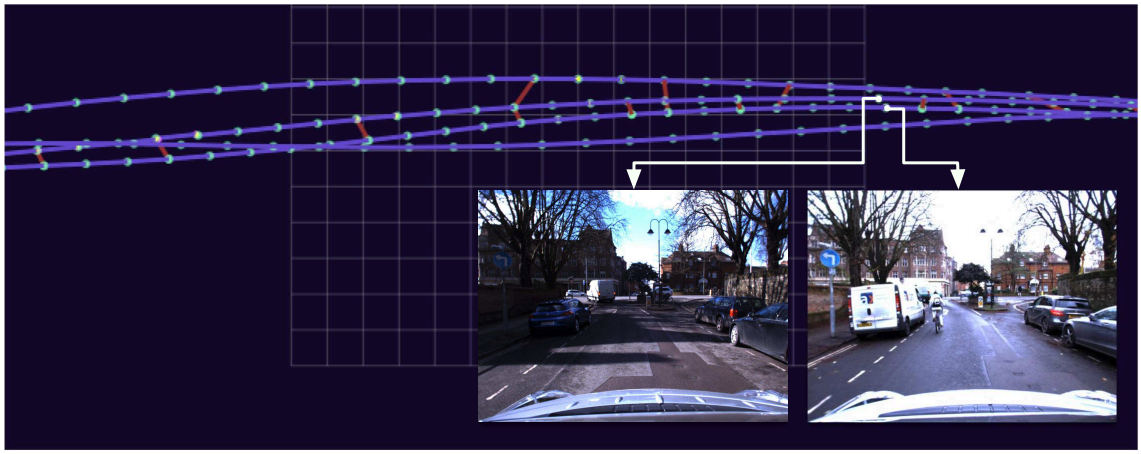


Figure 4.4: Snapshot of an experience graph $\mathcal{G}$ over an approximately 30 m stretch of road. Two images are shown which correspond to nodes in two of the four experiences contained in the map. Purple edges show visual odometry constraints, while red edges show loop closures between experiences. The position of nodes is determined by performing a projection from relative space to local Euclidean space using a breadth-first search, as discussed in Section 4.2.3.

except that the transformation $\mathbf{T}_{s,d}$ appended to the edge is obtained from localisation. Figure 4.3c shows how loop closures (in red) connect nodes across different experiences, where nodes contain images which represent the world under different appearance conditions.

Loop closures are added where possible to increase the connectivity of the graph structure. While it is helpful to think of experiences as independent traversals through the environment, at run-time the localiser operates on a single graph structure $\mathcal{G}$. Greater connectivity allows the localiser to access more experiences, resulting in more robust localisation. This behaviour is discussed in more detail in the following section. A snapshot of an experience graph over a 30 m stretch of road is shown in Figure 4.4. The figure shows how nodes contain images, and how loop closures (in red) tie different experiences together.

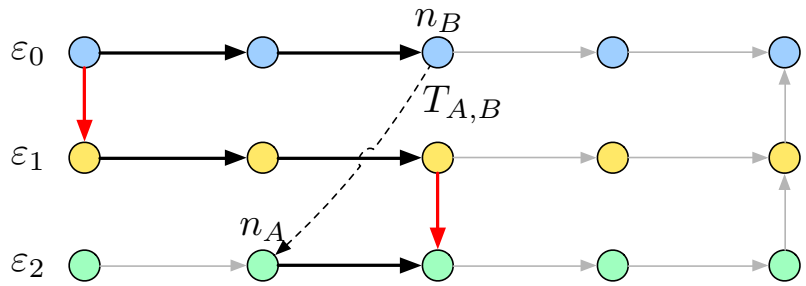


Figure 4.5: Relative pose can be calculated between nodes n_A and n_B by integrating the transformations along the edges of the shortest path between the two nodes.

4.2.3 Topometric Maps

The graph structure $\mathcal{G}$ is an example of a topometric map. Topometric maps were introduced in Chapter 2. Unlike a "global" map, which references all landmarks and robot pose in a single co-ordinate frame, a topometric map contains many reference frames connected by relative transformations (Brooks, 1985; Bosse et al., 2004). Because the formation of a topometric map does not require optimisation, augmenting an experience is a constant time operation.

As discussed previously, closed loop navigation of a robot requires local metric information over small distances, since the world is locally Euclidean. Over large distances, topological connectivity is sufficient to guide the robot towards its goal. A topometric map provides this information without the computational burden of large-scale optimisation. However, in the case that a particular application might require a global map (for example, visualisation of pose estimates in a global frame), the global optimisation could be performed in an offline process. Since map optimisation would only alter the transformations stored on the edges by a small amount, the global optimisation would be transparent to the localisation system.

An important competency in an experience-based framework is the ability to query the graph structure for relative pose estimates between nodes, across different experiences. In a topometric map, this operation must be performed as a graph

search since there is no external "world" reference frame. Figure 4.5 illustrates the scenario where a relative pose estimate $\mathbf{T}_{A,B}$ is obtained between nodes n_A and n_B, even though there is no loop closure directly connecting the two nodes.

Relative pose between two nodes can be determined as follows:

1. Perform a graph search to determine the shortest path between the two nodes. In Figure 4.5, the graph search begins at n_A in experience ε_2, traverses through intermediary experience ε_1, to find node n_B in experience ε_0.

2. Integrate the transformations stored on the edges of the shortest path, taking care to integrate transformations in the correct direction.

This graph search can be posed as a breadth-first search (BFS) to discover all nodes within radius r m in Euclidean space. For example, consider Figure 4.6. The breadth-first search start position is centred on the robot frame, labelled (a) and (b). A breadth-first search discovers nodes as it searches outwards, integrating transformations along each edge traversed such that the pose of a newly discovered node can be expressed with respect to the robot frame. This function can be thought of as a projection from relative space into a Euclidean frame centred on the robot. Because the projection is highly dependent on the position of the robot, the same map projects very differently into the robot frame as a function of the robot's location in the graph.

Importantly, the projection of the topometric graph structure into the robot frame is always metrically accurate nearby the robot. Yellow lines indicate when the pose of a node is ambiguous due to it being discovered from opposite ends of a closed loop, where different magnitudes of drift occur along each path. Note that these discontinuities in the graph always occur on the "opposite" end of the map to the robot's position, in terms of the total distance searched through the graph. This is due to the behaviour of the BFS, where nodes nearby the robot are discovered

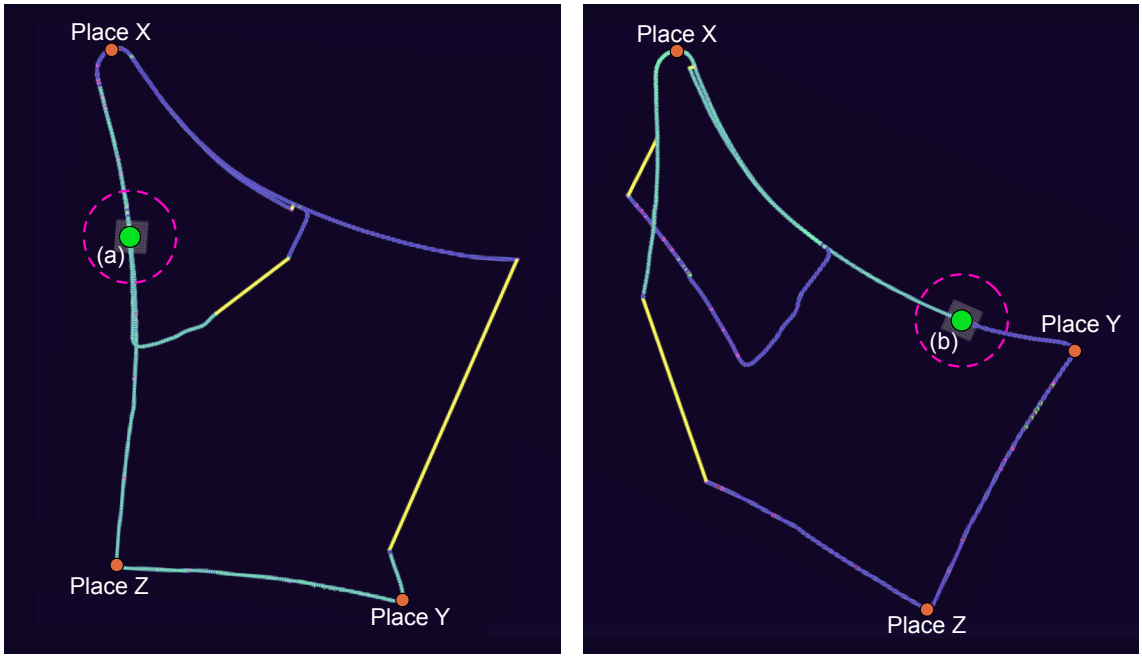


Figure 4.6: Diagram illustrating how a topometric map projects into the robot frame. Both maps here are identical, but appear different because they are projected into the robot frame when the robot is in different places, (a) and (b). The projection into robot frame is implemented as a breadth-first search (BFS), where the starting position coincides with the robot position. Yellow lines indicate ambiguity in the position of a node, due to that node being discovered via multiple search paths in a single loop, with different magnitudes of drift along each path. Note that these discontinuities in the graph always occur on the "opposite" side of the map to the robot, i.e. the furthest point from the robot in terms of graph search distance. The pink circles are of radius 20 m, and illustrate that the world is locally Euclidean about the robot.

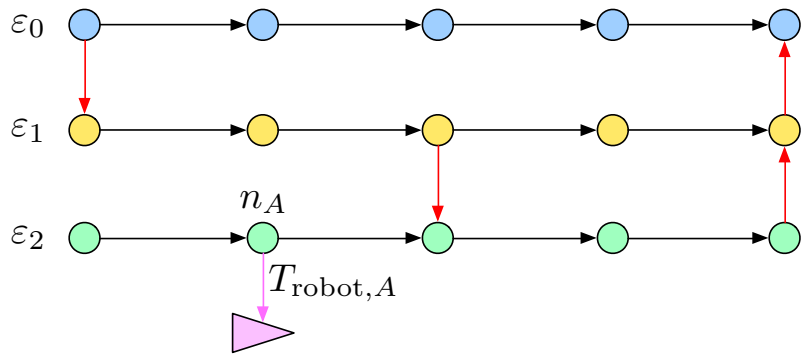


Figure 4.7: The localiser obtains a pose estimate with respect to one of the nodes in the graph. Here, localisation was obtained between the live image and node n_A, with 6-DOF transformation labelled $\mathbf{T}_{\text{robot},A}$.

first, and nodes far away from the robot are discovered last, accumulating more drift during the search.

A pink dashed circle of radius 20 m is drawn around each robot. This shows that over small distances, a metric assumption about relative pose holds, and that this metric frame is always centred on the robot (recall the close-up snapshot of the experience graph in Figure 4.4, projected into the robot frame). However, over large distances the metric assumption is no longer valid, and only topological connectivity can be assumed.

4.2.4 Specifying Camera Pose

In a traditional map, robot pose would be output in a single world reference frame. However, there is no concept of a world reference frame in a topometric map. Instead, each node in the experience graph $\mathcal{G}$ defines its own reference frame. At run-time, the localiser attempts localisation with at least one of the nodes in $\mathcal{G}$. If localisation is successful, the localiser can report pose with respect to that node, as shown in Figure 4.7. Alternatively, Figure 4.5 described how pose could be calculated with respect to neighbouring nodes in $\mathcal{G}$ by performing a graph search.

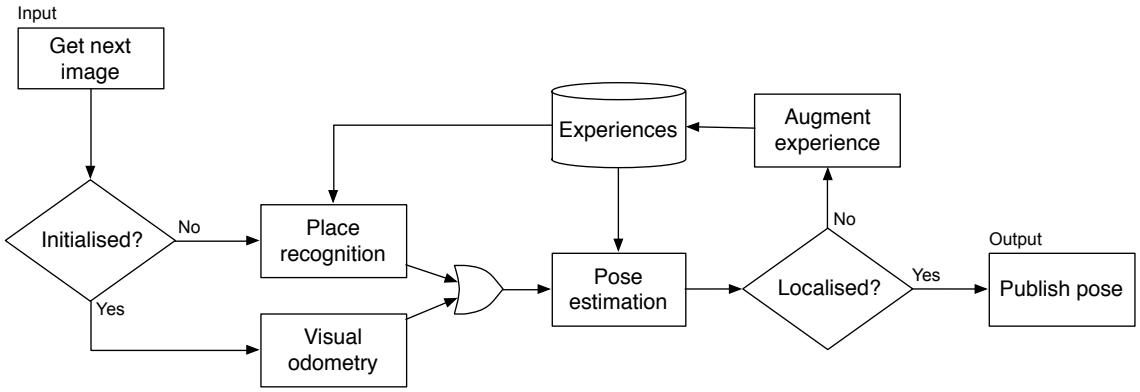


Figure 4.8: Diagram demonstrating the pipeline for Experience-Based Navigation, for a single pose estimation attempt. New images are received from the camera. If the localiser is not currently initialised, a place recognition tool is used to obtain the topological location of the robot. Otherwise, if the robot is currently localised, the robot's estimate in the map can be updated using visual odometry. Pose estimation operates in a local region of the map. If pose estimation results in a successful estimate, the pose is published to other subsystems downstream. If pose estimation fails, it means the robot is viewing the world under new appearance conditions, and a new experience can be created or augmented.

4.3 The Experience Framework

The experience paradigm refers to a mapping framework, and is agnostic to the localisation technique, or sensor-type, being used. This section describes the generic interfaces in our implementation of the experience framework, with reference to the block diagram in Figure 4.8.

4.3.1 Place Recognition in the Experience Graph

Place recognition is a localisation technique which outputs a topological estimate of the robot's position. A topological estimate is one which indicates the nearest node in the map; it does not contain any metric information. The topological estimate is used to initialise the localiser in the experience graph. Section 2.3.5 discussed this in more detail and presented a brief overview of related place recognition techniques.

Place recognition is a function defined as:

$$n_k = \text{PlaceRecognition}(I, \mathcal{G})$$

where n_k is the node nearest to the robot, I is the input image, and $\mathcal{G}$ is the experience graph. Because topological techniques do not make metric assumptions about the map structure, they can operate efficiently over large maps which have not been globally optimised. Topological techniques can be used to perform initialisation in the map, so that a human operator does not have to manually specify a starting point or other heuristics. It also enables the robot to perform loop closures in real-time.

Our implementation uses FAB-MAP (Cummins and Newman, 2008) to perform place recognition. FAB-MAP is an appearance-based probabilistic technique for recognizing places which the robot has previously visited. FAB-MAP maintains a topological map of the world, where each location in the map corresponds with a node in the Experience Graph. Since the Experience Graph explicitly supports the existence of multiple nodes in the same physical location, we extend the concept of multiple experiences to FAB-MAP's topological map as well. FAB-MAP uses SURF features to generate the bag of words representation which describes a place, so FAB-MAP localisation is prone to failure under appearance change in the same way that pose estimation is. While not investigated in detail in this thesis, we find that the concept of multiple experiences aids FAB-MAP topological localisation as well as pose estimation.

The complexity of FAB-MAP localisation is linear in the number of places in the map, so the use of multiple experiences within FAB-MAP increases the computational cost of localisation. However, we have not found this to be a limiting factor in our system so far. Rather, the additional computational cost is outweighed by

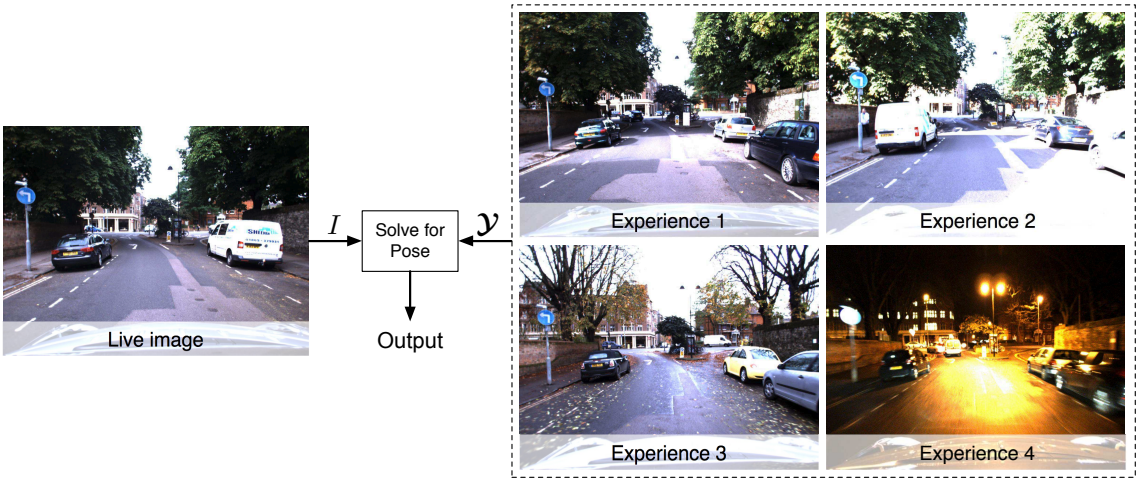


Figure 4.9: Illustration of the pose estimation task. An experience graph query returns a set of candidate nodes $\mathcal{Y}$. Pose estimation attempts to match the live image I to one of the images in $\mathcal{Y}$.

the improved localisation performance under appearance change.

4.3.2 Pose Estimation in the Experience Graph

While place recognition outputs a topological estimate of location, pose estimation outputs the metric 6-DOF pose of the robot with respect to a particular image, or node, in the experience graph. Pose estimation only operates on a small region of the graph at a time – usually within a local metric neighbourhood around a prior estimate of the robot's pose. This prior on the robot's pose can take the form of a topological estimate provided by the place recognition subsystem (in our implementation, FAB-MAP), for example when the robot is initialising in the graph. Alternatively, the prior can be obtained by integrating the relative motion estimate from visual odometry over the time period since the last successful pose estimation attempt. The pose estimation interface is defined as:

$$\{n_j, \mathbf{T}_{\text{robot},n_j}\} = \text{PoseEstimation}(I, \mathcal{G}, n_k, \mathbf{T}_{\text{robot},n_k})$$

where n_j is the node localised against, $\mathbf{T}_{\text{robot},n_j}$ is the 6-DOF transformation between the robot and n_j, $\mathbf{T}_{\text{robot},n_k}$ is the prior estimate of the robot's pose, and n_k is the node referenced in the prior. Note that n_j and n_k are not necessarily the same node, since the localiser may obtain a better localisation against a neighbouring node during the pose estimation task.

In our implementation, the pose estimation task operates on a set of candidates node $\boldsymbol{\mathcal{Y}}$. This set contains nodes nearby the robot which may result in successful localisation with the live image. An example of this is shown in Figure 4.9. This set is obtained by projecting the relative graph structure into the robot frame using a breadth-first search, as described previously. The robot frame is specified as a relative pose estimate, given by the prior $\mathbf{T}_{\text{robot},n_k}$.

Pose estimation is performed between the live image I_q and an image I_j from one of the candidate nodes n_j. Chapter 5 presents a probabilistic technique for determining which candidate nodes are most likely to result in successful localisation. While it would be possible to match the live image to several nodes simultaneously and perform a single optimisation for pose, we have not found it necessary. Rather, we perform multiple, independent pose attempts in parallel. We choose the "best" pose estimate as the one with the greatest number of inliers, although other heuristics could be used depending on the localisation technique employed. Chapters 6 and 7 discuss pose estimation in more detail.

We note that the localiser may jump between different experiences at run-time, depending on which nodes offer the best localisation performance. Section 4.2.3 described how a graph search can be used to transform the pose estimate between different reference frames in the graph structure as required.

4.3.3 Augmenting an Experience

Augmentation of the experience graph is the process of adding additional representations of the world to the stored set of experiences. To prevent unbounded growth of the experience graph, an experience creation policy determines when to add new representations to the map. This policy should attempt to minimise map size, while maximising the spectrum of appearance represented by the map, to support robust localisation.

A simple, but effective, policy is one based on localisation success. If the live image localises successfully in the experience graph, the experience graph must already contain an experience which closely matches the live image. In this situation, the live image is discarded and not stored in the experience graph. If localisation fails, this means that the experience graph does not contain a sufficiently similar representation of the environment to localise the live image. Under these circumstances, a new experience is saved to the experience graph.

Visual odometry provides relative motion between nodes as the new experience is created, and loop closures are added in real-time to increase connectivity between the new experience and the existing experience graph.

4.4 Summary

This chapter has described our approach to localisation in outdoor environments. We embrace the experience paradigm as a core principle in our work, using multiple, overlapping experiences to capture change in the environment. In outdoor areas, this change is commonly caused by changes in lighting, weather, season and scene structure. However, the system does not need to know the cause of the appearance change, but simply that appearance change has taken place. We describe how the implementation used in this thesis differs to previous work, where a single localiser

operates on a graph structure to perform pose estimation in real-time.

The concept of the experience graph was introduced as a framework for storing multiple experiences. The graph is an example of a topometric map, where global alignment in a single reference frame is not enforced. The graph-based framework facilitates queries and graph searches across multiple experiences. The framework is independent of the sensor or localisation technique being used, and contains generic interfaces for localisation and mapping functions. We utilise this as a base for our work over the following chapters.

Experience-Based Navigation is a powerful paradigm in large-scale localisation, however it introduces new challenges too. Chapter 5 describes and presents a solution to the problem of experience density, where as experience density increases, the probability of finding a successful match decreases given finite computational resources. Chapter 6 addresses the core localiser, presenting a new technique for obtaining robust data associations across extreme appearance change. This eases the requirement on the robot to have mapped every possible appearance condition before robust localisation is achieved, since the localiser is capable of localising across a wide range of appearance conditions. Finally, in Chapter 7 we present a new localisation system called Dub4 which brings together these methods to achieve a robust, real-time localiser.

Chapter 5

Prioritised Recollection of Experience

5.1 Introduction

This chapter is about intelligently managing a map of visual memories, and the prioritised recollection of map images to support time-constrained localisation. Chapter 4 described an experience-based framework for robust localisation in challenging outdoor environments. While this approach provides significant robustness to appearance change, it is computationally demanding. Figure 5.1 demonstrates the challenge of finding a successful match when the density of experiences is high. As experience density increases, the robot must do more work to obtain a successful localisation. This results in a navigation system which becomes less efficient over time. We find that resource-constrained machines cannot keep up with the additional work load, resulting in localisation performance which degrades as map size increases. This poses a new challenge: In a region with many stored representations, which one(s) should we try to localise against given finite computational resources?

Figure 5.2 illustrates the pipeline for localisation, showing the parameters which

Figure 5.1: This figure illustrates the problem of experience density. The localiser queries the experience graph $\mathcal{G}$ for the candidate nodes $\mathcal{Y}$ nearby the robot which may result in successful localisation. As more experiences are added to the map, the number of candidate nodes increases. Given that each experience is visually distinct, the live image I may only localise in a small subset of these nodes. This chapter presents a method for ranking the candidate nodes based on their probability of localising with the live image I.

control the number of localisation attempts that can be attempted. Localisation performance could be improved by increasing the number of CPU cores N_p, however this may not be feasible in many applications. Alternatively, the number of matching rounds N_r could be increased, however this would result in a reduction in online performance. Lastly, one could limit the experience density in the experience graph, however this would constrain the system's ability to adequately model the environment. Rather, we seek a method which pre-emptively caches a small subset of nodes nearby the robot which have a high probability of matching with the live image.

This chapter presents a probabilistic method for predicting which nodes are most likely to localise with the live image. We introduce the concept of "path memory" as a way of encoding the robot's localisation history into the map representation. Path memory consists of a collection of paths through the experience graph $\mathcal{G}$, where each path links nodes used by the robot during localisation on a particular outing.

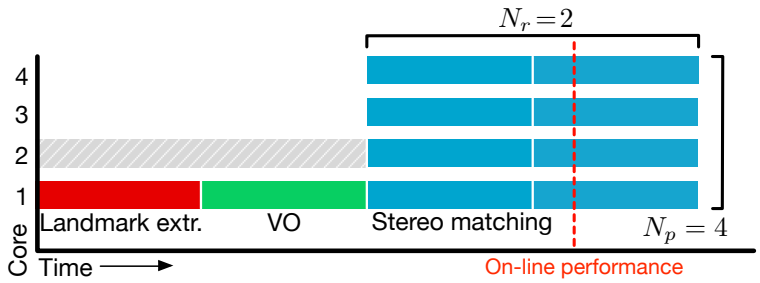


Figure 5.2: This figure illustrates a typical localisation attempt. A stereo image pair I_q is obtained from the camera. Visual odometry operates on I_q and I_{q-1} to estimate the robot's pose in the map. Attempts are made to match I_q with candidate nodes $\mathcal{Y}$ in the graph. This last step is performed in parallel, where we show the number of parallel processes $N_p = 4$ and the number of matching rounds $N_r = 2$. The requirement for online performance is shown as a dashed red line. As shown, with $N_r = 2$ the system would not be meeting its online performance criteria, and would have to reduce the number of localisation attempts by setting $N_r = 1$. This results in fewer localisation attempts.

These paths implicitly link relevant experiences together. For example, consider an experience graph containing sunny and rainy experiences. Without knowledge of the underlying causes of the appearance change (in this case weather), paths would link the sunny experiences together, and the rainy experiences together, since the two appearance modes are visually distinct. This information is used to predict which visual memories will be relevant in the next localisation attempt, conditioned on how the robot is currently localised in the graph.

The system is demonstrated over hundreds of kilometres, in a diverse range of lighting and weather conditions, scene clutter, camera occlusions, and permanent structural change in the environment.

5.2 Ranking Policies for Candidate Nodes

Chapter 4 presented an overview of the experience framework. When describing the pose estimation interface, we described how a set of candidate nodes $\mathcal{Y}$ are extracted from the experience graph using a breadth-first search. These candidate nodes are

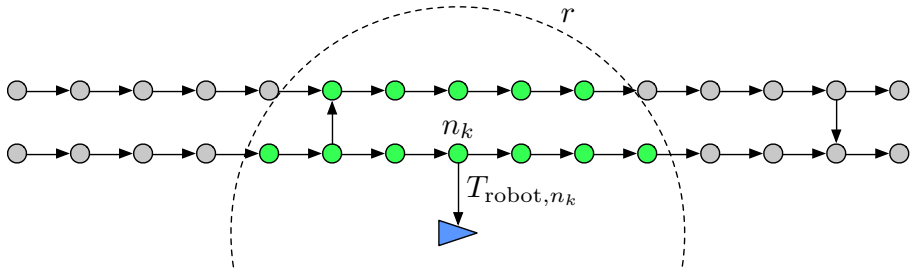


Figure 5.3: Diagram illustrating how a breadth-first search is used to discover candidate nodes $\mathcal{Y}$ (shown in green). $\mathbf{T}_{\text{robot},n_k}$ is the seed estimate of the robot's pose in the experience graph, referenced against node n_k. The transformation $\mathbf{T}_{\text{robot,node}}$ can be calculated by integrating transformations along the shortest path to each node. The search terminates when it exceeds a distance threshold in Euclidean space r, making the search independent of the spacing between nodes.

nodes nearby the robot which may result in successful localisation, due to their close proximity to the robot. Figure 5.3 illustrates how candidate nodes (green) are discovered using a breadth-first search. The pose estimation techniques operate on the set $\mathcal{Y}$, attempting to localise the live image I_q against images which correspond to the candidate nodes. We note that each localisation attempt with a node is independent and can be performed in parallel, resulting in potentially multiple successful matches in a single iteration. The vehicle needs at least one successful match to remain localised in the map. In the event that multiple localisation attempts are successful, a heuristic such as the number of inlier data associations is used to select the best estimate. A block diagram for pose estimation is shown in Figure 5.4.

However, some areas may require a high number of experiences to sufficiently model the appearance change in the environment, as shown in Figure 5.1. The number of candidate nodes $|\mathcal{Y}|$ increases with local experience density. Given finite computational resources, the robot may only be able to attempt localisation in a small number of these candidate nodes before being forced to abort in order to maintain constant processing time.

A ranking function Γ is introduced to order the candidate nodes by their probability of obtaining a successful match with the live image I_q. The localiser attempts

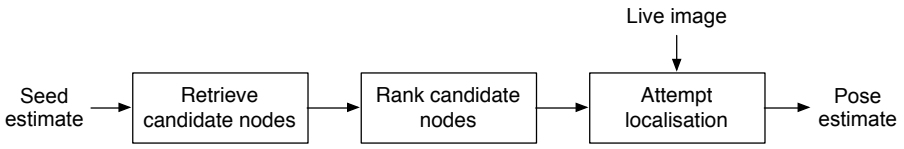


Figure 5.4: Block diagram illustrating the pipeline for pose estimation. A seed estimate provides a prior on the robot's pose in the experience graph. Candidate nodes are retrieved by performing a breadth-first search through the graph structure. The candidate nodes are ranked by a ranking function Γ based on their probability of obtaining a successful match with the live image. Localisation with the live image is attempted with the top ranking N nodes.

localisation with the top ranking N nodes, where N is the maximum number of localisation attempts allowed. This ensures that if computational resources are limited, only the candidate nodes with the greatest chance of successful localisation will be used.

Two ranking policies are considered. Firstly, a baseline ranking policy Γ_{distance} orders candidate nodes by their distance to the robot's position. Point-based localisation techniques have limited translation invariance, so nodes closer to the robot are expected to have a higher probability of obtaining a successful match. The probabilistic ranking policy presented in this chapter is referred to as Γ_{path}, and is discussed in more detail in the following sections.

5.3 Path Memory

"Path memory" is presented as a way to encode the robot's past use of the experience graph. Path memory refers to a collection of paths, where an individual path records the robot's trajectory through the experience graph on a particular outing. A path implicitly links nodes that represent the environment under similar conditions. Figure 5.5 demonstrates an example of this. Three experiences are shown, where the first experience (top) is captured under sunny conditions, and the remaining two experiences (centre, bottom) are captured under cloudy conditions.

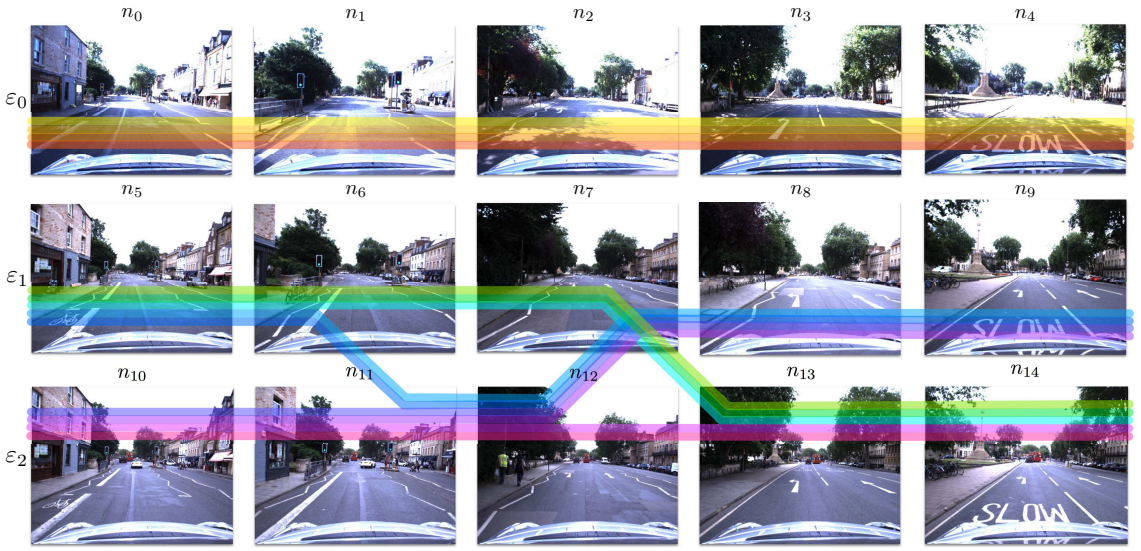


Figure 5.5: Path memory records the robot's use of the experience graph. Three experiences are shown here, one sunny (top) and two cloudy (centre, bottom). The robot drives through the environment 13 times, corresponding to the coloured horizontal lines. These lines connect images (i.e. nodes) which were used for localisation successfully at run-time. Trends emerge in the way the experience graph is used, and we exploit these trends at run-time to predict relevant experiences for localisation.

The robot traverses the area 13 times, each time recording the sequence of images used to perform localisation. We refer to these as "paths" through experience space. Very quickly, trends emerge in the way the experience graph was used. These trends are identified at run-time to make predictions about which image will result in successful localisation next, conditioned on how the robot is currently localised in the graph.

Figure 5.6 presents this in another way. Two paths are shown, where the red path might have been created on a sunny day and the blue path on a rainy day. The two paths link different nodes, since the weather conditions force the robot to localise in either sunny or rainy experiences. If the robot re-visits the area for a third time and starts to localise to nodes on the sunny path, we can infer that the robot will probably localise to other sunny nodes in the near future too. So, without knowing anything about what caused the appearance change in the environment, the robot can automatically learn which nodes are more likely to result in a successful

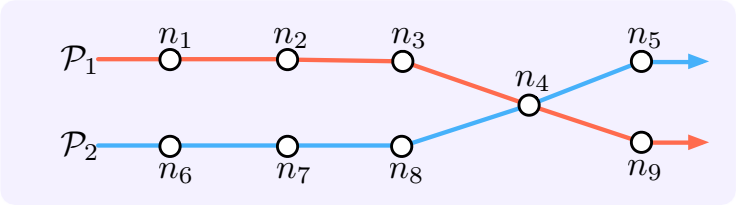


Figure 5.6: Paths connect nodes in the experience graph. Here, two paths are shown, $\mathcal{P}_1$ (red) and $\mathcal{P}_2$ (blue). From this we can infer that nodes $\{n_1, n_2, n_3, n_4, n_9\}$ represent the environment under similar conditions (e.g. afternoon sunshine), and $\{n_6, n_7, n_8, n_4, n_5\}$ represent the environment under a different set of conditions (e.g. rain). Note that the robot may localise in some nodes under both conditions, as shown here by n_4.

localisation.

In Chapter 4, the experience graph $\mathcal{G}$ was presented as a graph structure of nodes and edges, where edges contained 6-DOF transformations giving the graph a topometric structure. In this section, we introduce non-metric edges which do not contribute to the relative structure.

A path $\mathcal{P}_m$ of length k is defined as a sequence of non-metric edges:

$$\mathcal{P}_m = [e_0, e_1, ..., e_k]$$

where an edge connects two nodes, n_s and n_t in the experience graph $\mathcal{G}$.

Paths are created incrementally at run-time, after the localiser has finished processing. If the localiser returns a successful localisation match between the live frame and node n_k, which is different to the previously matched node n_{k-1}, the robot is considered to have moved in experience space from n_{k-1} to n_k. This triggers the creation of a new edge belonging to path $\mathcal{P}_m$, between n_{k-1} and n_k.

Path memory refers to the collection of paths stored in the database:

$$\boldsymbol{\mathcal{P}} = \left\{\mathcal{P}_1, ..., \mathcal{P}_{|\boldsymbol{\mathcal{P}}|}\right\}$$

where $|\boldsymbol{\mathcal{P}}|$ is the number of paths recorded, and a single path $\mathcal{P}_m$ represents the robot's trajectory through the experience graph on a particular outing.

5.4 Predicting Relevant Experience

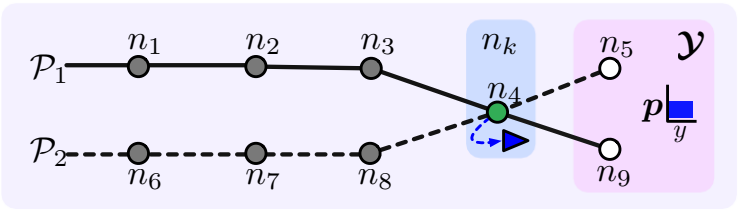
This section describes the probabilistic framework which enables the robot to predict candidate nodes that are likely to localise successfully. The algorithm generates a probability distribution over the set of candidate nodes $\boldsymbol{\mathcal{Y}}$ based on a set of conditionally independent observations. The system uses the robot's past localisation history as training data, and since this is stored implicitly in path memory $\boldsymbol{\mathcal{P}}$, the learning process is unsupervised.

In addition to recording path memory $\boldsymbol{\mathcal{P}}$, we also explicitly record a summary of the robot's recent localisation attempts on the live run. We record these nodes in $\boldsymbol{\mathcal{W}}$ and refer to the j^{th} node in the set as ${}^{j}\boldsymbol{\mathcal{W}}$. We define the term "recent" by the parameter T, where T is the number of preceding iterations of the localiser to remember (recall that the localiser can make several localisation attempts on one iteration). An example of these nodes is shown in Figure 5.7b, where $\boldsymbol{\mathcal{W}} = \{n_1, n_2, n_3, n_4, n_8\}$.

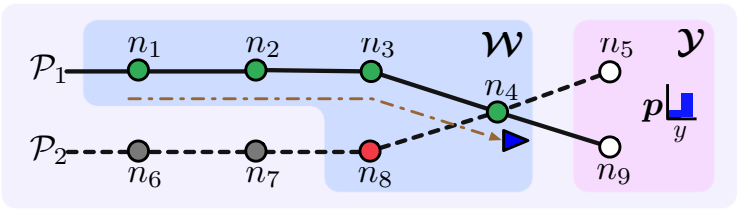
Each localisation attempt with nodes in $\boldsymbol{\mathcal{W}}$ will have resulted in either a successful or failed stereo match. In Figure 5.7b, successful localisation attempts are marked green, while failed localisation attempts are marked red. We record success or failure in the binary observation vector $\boldsymbol{\mathcal{Z}} = [z_1, ..., z_{|\boldsymbol{\mathcal{W}}|}]$. For each node in $\boldsymbol{\mathcal{W}}$, there exists a corresponding bit in $\boldsymbol{\mathcal{Z}}$ such that:

$$z_j = \begin{cases} 1 & \text{if localisation with node } {}^{j}\boldsymbol{\mathcal{W}} \text{ succeeded} \\ 0 & \text{if localisation with node } {}^{j}\boldsymbol{\mathcal{W}} \text{ failed} \end{cases}$$

In the example in Figure 5.7b, we see that $\boldsymbol{\mathcal{Z}} = [1, 1, 1, 1, 0]$.



(a) Prior



(b) Likelihood

Figure 5.7: This diagram illustrates how path memory is used to make predictions about which candidate nodes are most likely to result in successful localisation. Nodes in the experience graph are labelled $n_1, ..., n_9$ and two paths from path memory are shown, $\mathcal{P}_1$ and $\mathcal{P}_2$. In this example, the set of candidate nodes are $\boldsymbol{\mathcal{Y}} = \{n_5, n_9\}$. In Figure 5.7a, n_k is the node the robot is currently localised against ($n_k = n_4$). The prior distribution over the candidate nodes is calculated based on the number of paths connecting n_k and each node in $\boldsymbol{\mathcal{Y}}$. In Figure 5.7b, the set of observation nodes $\boldsymbol{\mathcal{W}} = \{n_1, n_2, n_3, n_4, n_8\}$ are the nodes which the robot has recently tried to localise against (the robot's trajectory is shown as a dot-dashed line). Green nodes denote successful localisation attempts, whereas red nodes denote failed localisation attempts. The corresponding observation vector is thus $\boldsymbol{\mathcal{Z}} = [1, 1, 1, 1, 0]$. A probability distribution over the candidate nodes is generated as shown, which is combined with the prior to make predictions about which candidate node is most likely to result in a successful localisation.

Lastly, the indicator variable $\boldsymbol{y} = [0, ..., 1, ..., 0]^T$ is defined as a vector of length $|\boldsymbol{\mathcal{Y}}|$. One element takes the value 1 and all remaining elements are 0. The variable y is defined as a discrete random variable describing which candidate node will localise successfully, under the assumption that only one candidate node will achieve successful localisation. The expression $p(y = i)$ is used to refer to the probability of successfully localising the live camera frame to the i^{th} candidate node ${}^{i}\boldsymbol{\mathcal{Y}}$.

Using Bayes Theorem, the probability distribution over $\boldsymbol{y}$ is calculated as:

$$p(\boldsymbol{y}|\boldsymbol{\mathcal{Z}}) = \frac{1}{\beta} \underbrace{p(\boldsymbol{\mathcal{Z}}|\boldsymbol{y})}_{\text{likelihood}} \underbrace{p(\boldsymbol{y})}_{\text{prior}} \tag{5.1}$$

where β is a normalisation constant. Since we only need to rank the candidate nodes, we do not explicitly calculate β. The likelihood and prior terms are discussed separately, before the system is presented as a whole again.

5.4.1 The Likelihood

Intuitively, we want to capture the following: in path memory, if many paths connect node ${}^{j}\boldsymbol{\mathcal{W}}$ and node ${}^{i}\boldsymbol{\mathcal{Y}}$, it means ${}^{j}\boldsymbol{\mathcal{W}}$ and ${}^{i}\boldsymbol{\mathcal{Y}}$ must represent the world under similar conditions (e.g. early morning sunshine). At run-time, if the robot localises in ${}^{i}\boldsymbol{\mathcal{W}}$, path memory would suggest that we are also likely to localise in ${}^{i}\boldsymbol{\mathcal{Y}}$. For example, in Figure 5.7b if we have localised to n_3, we also expect to localise to n_9 since it is connected by a path.

Recall that each binary element in $\boldsymbol{\mathcal{Z}}$ represents an observation, and that each observation corresponds to a node in $\boldsymbol{\mathcal{W}}$ which either succeeded ($z_i = 1$) or failed to localise ($z_i = 0$). We make the assumption that all localisation attempts in $\boldsymbol{\mathcal{Z}}$ are conditionally independent, given that we know which candidate node ${}^{i}\boldsymbol{\mathcal{Y}}$ will localise successfully. This is a simplification of reality, since the probability of localising to a node on a path is affected by the success or failure of all neighbouring localisation

attempts.

We introduce $\boldsymbol{\theta}$ as a $|\boldsymbol{\mathcal{Y}}| \times |\boldsymbol{\mathcal{W}}|$ matrix describing $p(\boldsymbol{\mathcal{Z}}|\boldsymbol{y})$, where a single element $\theta_{i,j}$ is defined as:

$$\theta_{i,j} = p(z_j|y = i) \tag{5.2}$$

We express the likelihood term for a particular candidate node ${}^{i}\boldsymbol{\mathcal{Y}}$ as:

$$p(\boldsymbol{\mathcal{Z}}|y = i) \propto \prod_{j=1}^{|\boldsymbol{\mathcal{W}}|} p(z_j|y = i) \tag{5.3}$$

We treat each observation in z_j as a single Bernoulli experiment described by $\theta_{i,j}$, where the number of trials in the experiment is the number of paths connected to ${}^{i}\boldsymbol{\mathcal{Y}}$. These parameters are learned from path memory:

$$\theta_{i,j} = \frac{Z_{i,j} + \alpha_j}{\sum_{x=1}^{|\boldsymbol{\mathcal{W}}|} (Z_{i,x} + \alpha_x)} \tag{5.4}$$

where $Z_{i,j}$ is the number of times a path links ${}^{j}\boldsymbol{\mathcal{W}}$ and ${}^{i}\boldsymbol{\mathcal{Y}}$ in path memory, and the parameter α_j specifies a prior Beta distribution. We set $\alpha_j = 1$ to represent the probability that in the absence of path memory, all observations are equally likely. This can be thought of as adding "pseudocounts" to the results from the binomial experiment in order to prevent the "zero count problem" which can occur in sparse training sets (Murphy, 2012).

The likelihood term can be thought of as the probability of the robot's live trajectory generating the observation vector $\boldsymbol{\mathcal{Z}}$, given that a particular candidate node localises successfully. In the example in Figure 5.7b, we would say the observation vector $\boldsymbol{\mathcal{Z}} = [1, 1, 1, 1, 0]$ is unlikely if n_5 were to localise successfully. However, the observation vector $\boldsymbol{\mathcal{Z}}$ would be likely if n_9 localised successfully, since this corresponds with the knowledge in path memory. Thus, we calculate the likelihood term

as:

$$p(\boldsymbol{\mathcal{Z}}|y=i) \propto \prod_{j=1}^{n} \theta_{i,j}^{\mathbb{I}(z_j=1)} (1-\theta_{i,j})^{\mathbb{I}(z_j=0)} \tag{5.5}$$

where $\mathbb{I}(x=a)$ is an indicator function, such that:

$$\mathbb{I}(x=a) = \begin{cases} 1 & \text{if } x=a \\ 0 & \text{otherwise} \end{cases}$$

5.4.2 The Prior

The prior models our initial belief in the probability distribution over $\boldsymbol{\mathcal{Y}}$, in the absence of the observation vector $\boldsymbol{\mathcal{Z}}$. The prior is calculated by querying path memory $\boldsymbol{\mathcal{P}}$ for the number of paths connecting the node n_k currently localised in, to each candidate node in $\boldsymbol{\mathcal{Y}}$. We bias the prior towards candidate nodes with many paths connecting n_k and ${}^i\boldsymbol{\mathcal{Y}}$. An example of this is shown in Figure 5.7a, where paths connect n_k and the candidate nodes $\boldsymbol{\mathcal{Y}} = \{n_5, n_9\}$.

The parameter vector $\boldsymbol{\pi}$ is used to model the probability distribution over $\boldsymbol{\mathcal{Y}}$, where π_i is the probability that candidate node ${}^i\boldsymbol{\mathcal{Y}}$ will localise successfully.

The elements in $\boldsymbol{\pi}$ are calculated using the Dirichlet expectation for the corresponding i^{th} candidate node:

$$\pi_i = \frac{N_{i,k} + \gamma_i}{\sum_{x=1}^{|\boldsymbol{\mathcal{Y}}|} (N_{x,k} + \gamma_x)} \tag{5.6}$$

where $N_{i,k}$ is the number of paths in path memory connecting n_k and ${}^i\boldsymbol{\mathcal{Y}}$. We set $\gamma_i = 1$ to represent a uniform distribution over the candidate nodes in the absence of path memory.

5.4.3 Implementation

We began by introducing our system's output as a probability distribution over the set of candidate nodes $\boldsymbol{\mathcal{Y}}$ in Equation 5.1. We have shown that this can be achieved by simple event counting (Equations 5.4 to 5.6), where events are stored implicitly in path memory. This enables the robot to learn from its past localisation history and make robust yet computationally inexpensive predictions.

The probability distribution over the candidate nodes $\boldsymbol{\mathcal{Y}}$ is calculated using the equations for the likelihood (Equation 5.5) and prior (Equation 5.6):

$$p(y = i|\boldsymbol{\mathcal{Z}}) \propto \pi_i \prod_{j=1}^{n} \theta_{i,j}^{\mathbb{I}(z_j=1)} (1 - \theta_{i,j})^{\mathbb{I}(z_j=0)} \tag{5.7}$$

Finally, the set of candidate nodes $\boldsymbol{\mathcal{Y}}$ is ranked by the probability distribution over $\boldsymbol{\mathcal{Y}}$ such that relevant nodes are prioritised over nodes unlikely to obtain successful localisation.

5.5 Results

This section tests the system's ability to perform life-long navigation in large outdoor environments. In particular, we are concerned with how localisation performs as the computational resources available to the localiser are decreased. The proposed ranking policy Γ_{path} is compared against a baseline ranking policy Γ_{distance} over three challenging outdoor datasets, totalling 205 km of data. These include:

1. The Keble dataset, consisting of 56 km of data around a 2.2 km route in a busy urban environment.

2. The Begbroke dataset, consisting of 75 km of data around a 0.7 km route in a private business park. Data was collected over a 12 hour period between 7 am and 7 pm.

3. A subset of the Central Oxford dataset, consisting of 76 km of data around a 7.6 km route. This dataset was collected prior to the 1000 km Central Oxford dataset, but traverses much of the same route.

These datasets contain challenging outdoor appearance change including sun, rain, camera occlusion, and changing illumination conditions. Section 3.1 presents these datasets in more detail.

So far, we have presented two ways that the robot can learn from its surroundings. Firstly, the robot automatically augments its map with new experiences as it encounters change in the environment. Secondly, the robot learns to predict which experiences will be used next, based on the robot's prior use of the experience graph. This means that the ordering of logs input into the system affects the overall localisation performance. We use cross-validation to prevent bias toward a particular ordering of logs which yields optimal localisation performance.

Our implementation of cross-validation is described in Section 3.2.3, and is briefly summarised here. In each dataset, there are N logs available. The logs are divided into five groups. For each experiment, four of the groups are used to create the experience graph and record path memory. The hold-out group is used to perform localisation, where the map is not modified during localisation. In terms of the parameters described in Section 3.2.3, the number of logs used during mapping is $e = \frac{4}{5}N$ and during localisation $l = \frac{1}{5}N$. Experiments and localisation results are presented below.

5.5.1 Increasing Experience Density

This first experiment can be considered a motivating case for an intelligent ranking policy. Figure 5.8 plots localisation performance as a function of map size, where localisation performance is measured as the probability of driving more than 5 m

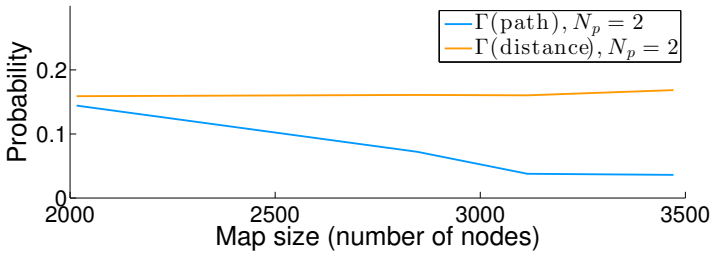


Figure 5.8: The probability of travelling more than 10m without a successful localisation is shown as a function of map size (the number of nodes in the experience graph), where the length of the route is kept constant. Without an informed ranking policy, localisation failure increases with map size, whereas our proposed ranking policy Γ_{path} results in an improvement by a factor of 4 over time.

without a successful localisation attempt. The figure shows that when using the baseline ranking policy Γ_{distance} on a robot with limited CPU cores, localisation performance *degrades* over time as nodes are added to the experience graph. This is expected, since if the robot can only localise against N nodes in a single iteration, increasing the number of candidate nodes $|\boldsymbol{\mathcal{Y}}|$ dilutes the chances of a naive selection policy choosing the nodes which will result in successful localisation. Figure 5.9 shows that only a small subset of the candidate nodes have the potential of matching successfully with the live image. This is also expected, since by definition new experiences are only added when they are sufficiently distinct from those already stored in the experience graph.

However, when using the probabilistic ranking policy Γ_{path}, Figure 5.8 shows that localisation performance *improves* as more experiences are added to the map. Specifically, the probability of localisation failure is reduced by a factor of 4 as nodes are added to the map. This is because the additional nodes provide experience diversity (the map models the environment over a wider spectrum of appearance change), which the ranking policy Γ_{path} exploits by selecting experiences which are more likely to localise the live camera image successfully. This is a significant result for robots with limited computational resources, as it demonstrates that Γ_{path} enables long-

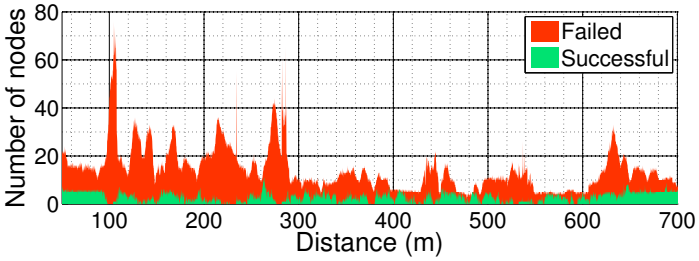


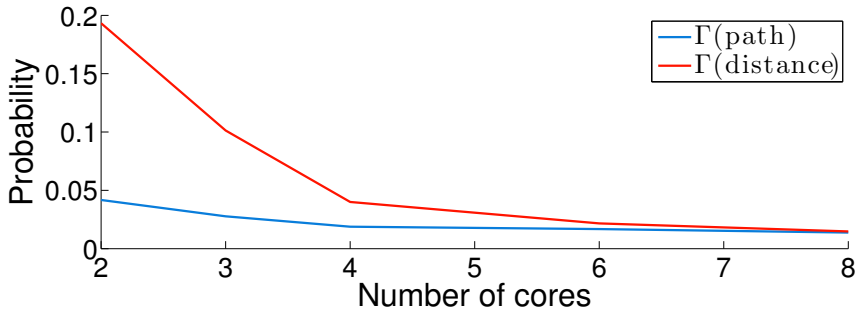
Figure 5.9: The size of the candidate node set $|\mathcal{Y}|$ is plotted as a function of distance around one loop of the Begbroke Science Park. The red and green areas are the number of candidate nodes that would have resulted in failed and successful localisation attempts respectively, had the robot attempted to localise against every candidate node in $\mathcal{Y}$. This graph was calculated offline for demonstration purposes - at run-time, the robot can make only a finite number of localisation attempts and needs to predict which nodes are likely to result in localisation success.

term autonomy and life-long learning in spite of the resource constraints imposed on the system.

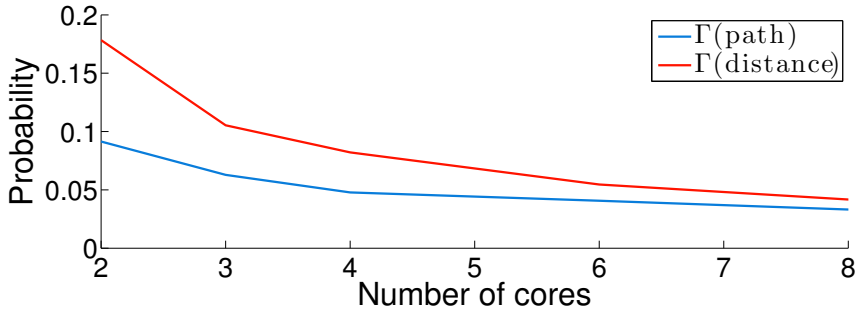
5.5.2 Effect on Localisation Performance

This experiment tests how the ranking policies Γ_{path} and Γ_{distance} affect localisation performance, as a function of the number of CPU cores made available during localisation. Reducing the CPU cores available reduces the number of localisation attempts the localiser can make while maintaining run-time performance, making localisation more difficult. Figure 5.10 presents the results of this experiment, where localisation performance is measured as the probability of travelling further than 5 m without a successful localisation in the experience graph. This is an important measure, since during periods of localisation failure the robot must proceed in open loop using visual odometry.

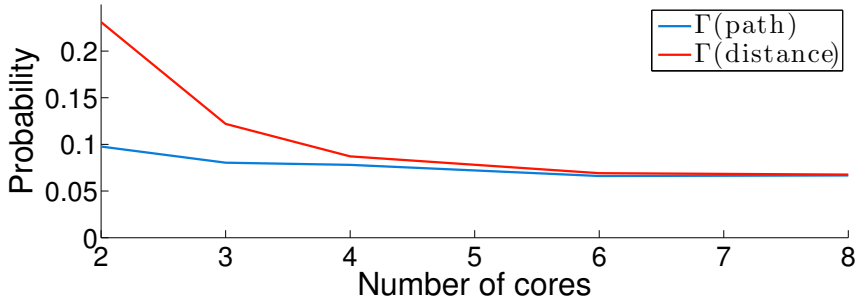
The figure shows that for both ranking policies, localisation failures reduce as the number of cores are increased, converging on the best possible localisation performance given infinite resources. However, Γ_{path} clearly outperforms Γ_{distance} when the number of CPU cores is fixed to a small number. From the perspective of effi-



(a) Begbroke Dataset



(b) Keble Dataset



(c) Oxford 10 km Dataset

Figure 5.10: The probability of travelling more than 5 m without a successful localisation in the experience graph, as a function of the number of cores available during localisation. The figures show that in all three datasets, the ranking policy Γ_{path} is less likely to result in localisation failure compared with the baseline policy Γ_{distance}.

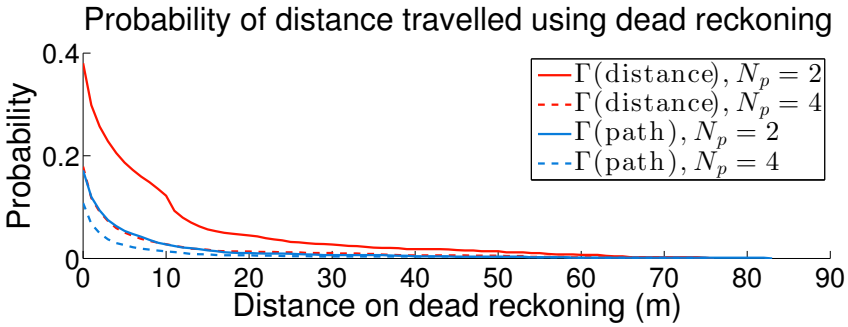


Figure 5.11: Graph showing the probability of travelling a certain distance without successfully localising in the experience graph, using the Begbroke dataset. During this period, the robot proceeds in open loop using visual odometry. The graph shows that using Γ_{path} maintains performance using using half the number of cores when comparing Γ_{path} with $N_p = 2$ and Γ_{path} with $N_p = 4$.

ciency, Γ_{path} results in approximately the same localisation performance as Γ_{distance}, but uses only half the number of CPU cores.

Of the three datasets, the Begbroke dataset (Figure 5.10a) showed the biggest improvement using Γ_{path}, reducing localisation performance by a factor of 5 for number of cores $N_p = 2$. This can be attributed to the large number of repeat traversals of the Begbroke loop compared with the other datasets.

Conversely, the Oxford 10 km dataset's performance with Γ_{path} and Γ_{distance} converges at approximately 6 CPU cores, fewer than the Begbroke and Keble datasets. This is because the Central Oxford dataset does not contain as many repeat traversals through the same area as the Begbroke and Keble datasets, resulting in reduced experience density. Over time, experience density would certainly increase and require more CPU cores to obtain the optimal performance with Γ_{distance}.

Figure 5.11 presents a more complete view of localisation performance, plotting the probability of localisation failure over a particular distance. The graph shows that Γ_{path} and number of CPU cores $N_p = 2$ provides nearly identical performance to Γ_{distance} and $N_p = 4$, showing that the same performance is obtained while performing half the computation work.

The computational cost of implementing Γ_{path} was measured on a 2.3 GHz Intel

Core i7 processor, where the maximum processing time was 0.5 ms. Given that the localiser operates at 16 Hz (the frame rate of a Bumblebee XB3 camera), this represents a negligible portion of the total processing time required to perform a single localisation attempt.

We note that while Γ_{path} outperforms Γ_{distance} in every test, Γ_{distance} still performs reasonably well considering it operates on very limited information. This is because the point features used during localisation have limited invariance to translation, so nodes that are closer to the robot are more likely to result in successful feature correspondences and consequently in successful localisation. However, this approach scales poorly with experience density, requiring more CPU cores to process greater numbers of candidate nodes to ensure successful localisation.

5.5.3 The Effect of Changing T

The parameter T controls the number of preceding localisation iterations to remember when generating the summary of recent localisation attempts $\boldsymbol{\mathcal{W}}$ and corresponding observation vector $\boldsymbol{\mathcal{Z}}$. Figure 5.12 shows that between $T = 5$ and $T = 200$, localisation failure increases very slightly with T, a result of observations close to the robot being more relevant than those further away. However, this effect is minimal, and for $5 > T > 200$ the localisation performance is not sensitive to the parameter T. Note that for $T = 0$ the performance decreases significantly. This is because the likelihood term (Section 5.4.1) is not used when $T = 0$ and the candidate nodes are predicted solely using the prior distribution (Section 5.4.2). This justifies the use of the likelihood term in Section 5.4.1.

The parameter T is a parameter which is in part a function of how quickly we expect the appearance of the world to change. For example, if the appearance of the world were always to change drastically from frame to frame, maintaining a history of recent localisation attempts ($T > 0$) would not yield an improvement

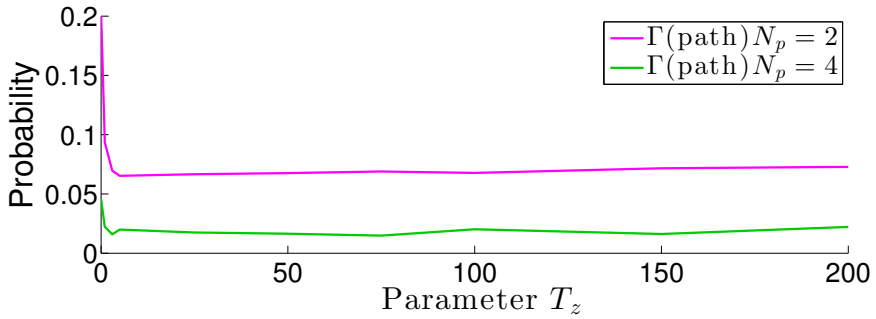


Figure 5.12: Graph showing that localisation performance is not sensitive to T, for $T > 5$. This parameter controls the number of preceding localisation iterations to remember when generating the set of observation nodes $\mathcal{W}$.

in performance. Of course, in reality outdoor appearance change occurs relatively slowly (weather, lighting and season change relatively slowly compared with the frame rate of a camera), which is why the techniques presented in this chapter are effective in predicting relevant nodes for localisation.

A more realistic scenario would be a single, abrupt change in appearance conditions, for example if the sun suddenly came out from behind the clouds. This step change in appearance would initially mean that the predicted nodes would fail to localise, since there is some inertia encoded by the parameter T. However, since the localisation history records localisation failures as well as successes, as more "cloudy" nodes failed to localise the penalty on them would grow. Over several iterations, the ranking algorithm would start to prioritise other nodes in the vicinity for localisation (nodes which are not cloudy in appearance). The success or failure of these localisation attempts would again be captured in the localisation history, and further contribute to the performance of the ranking algorithm. When eventually a node was found which matched the live image, the ranking algorithm would recover and successfully begin to predict nodes for localisation.

5.6 Summary

Experience-Based Navigation is a simple, but effective, technique for achieving robust localisation in spite of challenging appearance change in outdoor environments. However, as more experiences are added to the map, the localiser must do more work to ensure successful localisation. In many applications, the additional computational resources may not be available.

We have presented a technique for intelligent memory management. As the robot traverses the environment, we record the sequence of nodes that were used during localisation. This information is stored in Path Memory, which consists of a sequence of non-metric edges in the experience graph. At run-time, this information is used to predict which experiences will be relevant to the localiser, conditioned on how the robot is currently localised in the experience graph.

We have demonstrated how the prioritised recollection of relevant experiences is essential in ensuring real-time, robust localisation performance for resource-constrained robots. We evaluate our system on three different datasets totalling 206 km of outdoor travel. We show that an informed ranking policy that exploits knowledge of the robot's past use of the experience graph reduces localisation failure by as much as a factor of five for robots with a limit on the number of CPU cores and processing time for localisation. Even in the case of sparse training data, the system still outperforms the baseline ranking policy based on distance. From an efficiency perspective, we are able to maintain localisation performance while using half the number of CPU cores as previously.

The system presented so far is capable of a high degree of localisation performance. However, the robot requires many repeat visits to an area to build up a sufficient number of experiences for robust localisation. This may not be possible in many scenarios. The underlying cause of this problem is that the localiser is

brittle to appearance change. The following chapter addresses the localiser subsystem, presenting a new technique for obtaining data associations in spite of extreme appearance change.

Chapter 6

Place-Dependent Landmark Detectors

6.1 Introduction

So far, this thesis has approached the problem of appearance change from the perspective of experience-based mapping. By incrementally building up a map of overlapping experiences, localisation is robust to changes in lighting, weather and season. While this technique is effective, some areas may exhibit a wide spectrum of appearance change. These environments would require many repeat visits, under different environmental conditions, before robustness is achieved. This may not be possible when autonomy is required within a short space of time.

The requirement for many experiences is a limitation of the localiser. A brittle localiser is only able to localise across a narrow band in the spectrum of appearance change, meaning that many experiences are required to generate a complete model of a changing world. This lack of robustness can be attributed to the use of point features such as SIFT, SURF and BRIEF for localisation. 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, or when there has been large-scale structural change in the environment. Secondly, *the feature descriptors are not invariant* to the level of appearance change encountered in outdoor environments. Point features describe low level elements in the scene, which while providing accurate landmarks for pose estimation, tend not to be robust to changes in lighting and weather.

In this chapter, we present an alternative to traditional point feature techniques. An unsupervised training algorithm is presented which uses data captured from a single visit to an environment to extract distinctive landmarks for localisation. For every landmark identified, a bespoke linear SVM classifier is trained using Aggregated Channel Features (ACF) as the underlying feature type. At run-time, these detectors are used to identify known landmarks in the environment for the purposes of localisation. We find that these landmarks are significantly more robust to appearance change than traditional point feature approaches, enabling the localiser to localise across a much wider range of appearance change.

Our work is similar in spirit to McManus et al. (2015), where linear SVM classifiers were trained to detect mid-level, distinctive regions in the image. The authors presented large-scale localisation results across different times of the day (including between day and night), and in a number of challenging weather conditions. However, the approach we present in this thesis differs in a number of key areas:

1. The method proposed in this chapter operates on a single pass through the environment, whereas McManus et al. (2015) required many repeat visits to the same place in order to train robust classifiers.

2. The training method in McManus et al. (2015) used GPS to manually align the datasets. If GPS were not available in an area (for example, due to tall

Figure 6.1: Robust landmarks in this scene might include the signpost (yellow), the tree in the intersection (green), or the building's roof (pink). These landmarks are used to perform robust localisation across challenging appearance change in the environment.

buildings or indoors), a human operator would have to align the datasets manually.

3. The training method presented here relies on inexpensive geometric tests, as opposed to the method described by Doersch et al. (2012). This results in a significantly faster training algorithm. As a result, much larger maps can be trained, with fewer computational resources.

4. Our method exploits the RGB colour information in images by using ACF features (Dollár et al., 2014) as the underlying feature representation, rather than HOG (Dalal and Triggs, 2005).

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 main 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%.

Figure 6.2: 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.

6.2 Environment Model

The training method is an unsupervised, offline mining procedure which identifies and models distinctive landmarks in the environment for localisation. A distinctive landmark might be a postbox alongside the road, a marking on 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. Figures 6.1 and 6.2 illustrate examples of the landmarks which the training method seeks to identify and describe.

Rather than training generic object detectors, we train banks of unique landmark detectors on a place-dependent basis. This means that for every landmark, we train

a bespoke detector to detect that landmark. These detectors only have meaning in the local environment around the landmark. The notion of place-dependence is a powerful one, as it allows us to over-fit models to the specific environment we are operating in. Given a coarse prior on the robot's location at run-time, only the detectors which are in close proximity to the robot are tested.

Linear SVMs are used because they provide a powerful way to describe the appearance of landmarks. Traditional point feature techniques can only operate on a single image, i.e. a single observation of a landmark. However, SVMs can be trained using multiple observations of the landmark. Firstly, this allows us to explicitly train translation invariance into the appearance model by observing the landmark multiple times from different perspectives. Secondly, we can generate synthetic observations of the landmark to simulate varying illumination conditions. For example, a simple image transformation to darken the image can approximate the landmark's appearance under poor lighting conditions. These real and synthetic landmark observations are used to train robust linear SVM classifiers. We show that using a single pass through the environment, we are able to achieve sufficient generality to localise across the extreme appearance change shown in Figure 6.5. Unlike point features, these patch detectors are able to *generalise* beyond the data on which they have been trained.

6.2.1 Definition of a Landmark

We use the term "landmark" to refer to a distinctive element in the environment which can be used for localisation. Landmarks in traditional point-feature localisation systems usually correspond to 3D points in the world. For example, keypoints may correspond to the corner of a windowsill or signpost. If these 3D landmarks are observed at run-time, the data associations can be used to determine the robot's pose.

However, the landmarks discussed in this chapter 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 landmarks to "behave" in the way that a conventional 3D point-feature landmark would. This allows us to exploit existing point feature pose optimisation techniques, in that we only change the manner in which data associations are obtained. To this end, we require that as the robot moves through the environment, the appearance of the landmark should project into the camera frame consistently. This behaviour is defined in Chapter 2 by the camera's projective function, where an observation $\mathbf{z}$ of a 3D landmark $\mathbf{X}_{\text{origin}}$ is described by $\mathbf{z} = \mathbf{K}[\,\mathbf{I}\,|\,\mathbf{0}\,]\mathbf{T}_{\text{camera,origin}}\mathbf{X}_{\text{origin}}$. For each place k in the map, the training method outputs a bank of landmarks:

$$\mathcal{B}_k = \{l_0, l_1, l_2....\}$$

where the i^{th} landmark is represented as:

$$l_i = \{\mathbf{d_i}, \mathbf{X_i}\}$$

where $\mathbf{d_i}$ is the linear SVM detector trained to detect the landmark (i.e. the landmark's appearance model), and $\mathbf{X_i}$ is the homogeneous co-ordinates of the landmark relative to the co-ordinate frame k.

6.2.2 Extracting Distinctive Landmarks

This section describes the algorithm for generating a bank of landmark detectors $\mathcal{B}_k$. The training algorithm iteratively applies simple geometric tests on each candidate image patch to determine whether the patch is a robust landmark for localisation. It is an unsupervised, camera-only technique that does not rely on GPS or manual

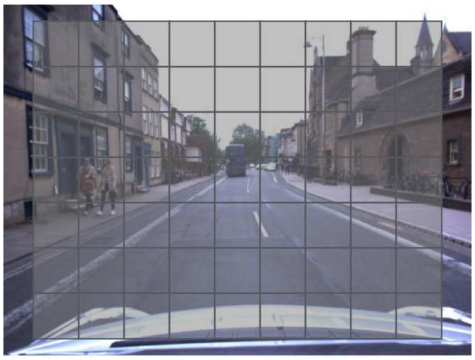
alignment of training images.

The intuition behind the training algorithm is a simple geometric check, described as follows. The training method seeks to identify visually distinctive elements in the environment for localisation. Additionally, the appearance of the landmark is required to project into the camera frame in the way that a 3D point would. The technique presented here exploits the relationship between these two requirements – that a landmark detector which has been trained to detect a nondescript landmark will fire inconsistently across the image.

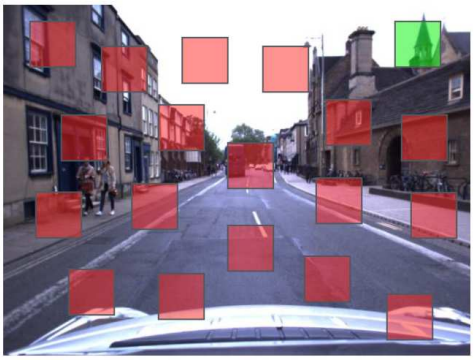
For example, a landmark detector trained to detect a small patch of road would not produce a good landmark for localisation, since that patch of road is unlikely to be visually distinct. By repeatedly observing the landmark as the robot drives through the environment, the system can attempt to solve for the 3D position of the patch of road. This solve can fail for two reasons, (i) the landmark cannot be modelled as a 3D point landmark, or (ii) the detector has not fired consistently because the landmark is not visually distinct. Either of these reasons is cause to reject the landmark.

The training procedure is described below, while Figure 6.3 describes the training procedure graphically.

1. **Train seed detectors**. Select a single image I_0 from place k. Slide a window of size $\mathbf{s}$ over image I_0. Train detector $\mathbf{d_i}$ for each new position of the window. In our implementation, we use Aggregated Channel Features (ACF) as the underlying feature representation. Figure 6.3b demonstrates how the linear SVM is trained by obtaining negative samples from the remainder of the image (Section 6.2.3 describes how the linear SVM classifiers are trained in more detail). These detectors constitute the set of seed detectors and are stored 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



(a) A grid is placed over the input training image. Initially, all cells are candidates for landmarks. The training method iteratively removes patches which do not correspond to unique landmarks.



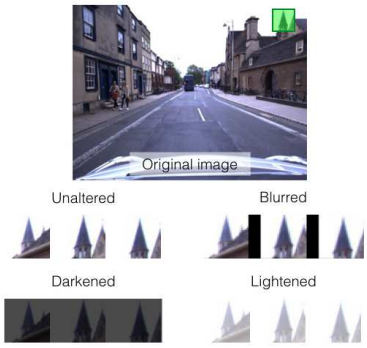
(b) For each candidate landmark patch (green), a linear SVM classifier is trained. The negative class consists of randomly sampled patches from the remainder of the image (red). Aggregated Channel Features (ACF) are used as the underlying feature type.

(c) Test the classifier on neighbouring images in the image stream. Attempt to triangulate for the 3D position of the classifier.

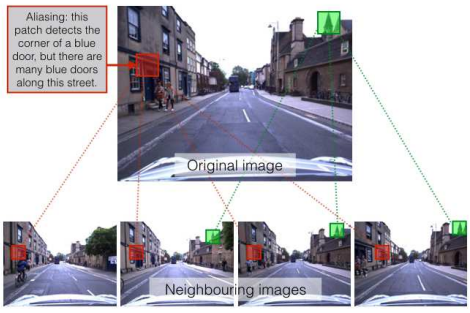
(d) If the classifier fires inconsistently (red), the optimisation for landmark position will contain a high RMS error. These patches are rejected.

Figure 6.3: Training method for identifying and modelling distinctive landmarks in the scene (figure continues on next page).

(e) The remaining classifiers all have a corresponding 3D position and can be used for localisation.



(f) The classifiers are re-trained using multiple observations of the landmark from neighbouring images. Synthetic observations are generated by darkening, lightening and blurring the input images.



(g) A test for aliasing is performed. For example, there are many blue doors along this street – patches which fire inconsistently over a local window around the original training image are rejected.

(h) This procedure is repeated for different patch sizes. Different classifier sizes result in different combinations of scene elements being used for localisation.

Figure 6.3: Training method for identifying and modelling distinctive landmarks in the scene (figure starts on previous page).

remain.

2. **Test detectors in nearby images**. Query the graph structure for images close to image I_0 and store them in set 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 I_1, as shown in Figures 6.3c and 6.3d using the method described in Section 6.2.3. We require a landmark to be visible in all images in 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. This function is described in Section 6.2.4. 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 6.2 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 each landmark has been observed in each image in I_1, retrain the linear SVM classifiers in $\mathcal{B}_k$ using these multiple observations. In other words, we use the successful detections from the previous two steps to make data associations between landmarks across different images. This makes the linear SVM detectors more robust. Figure 6.3f illustrates this re-training step. 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 blue door. 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 blue doors from other houses along the street. This is a problem for localisation, as we want to avoid the data association problem of knowing which blue door has been detected. This example is shown in Figure 6.3g. Additionally, the orange patches in Figure 6.2 show other examples of scene elements susceptible to aliasing.

   To this end, retrieve a set of test images 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 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 I_2. If a detector has fired incorrectly, remove it from $\mathcal{B}_k$.

The bank of detectors $\mathcal{B}_k$ now only contains a set of robust landmark detectors. This process is repeated for all places in the map, and optionally for different patch sizes. 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. Additional implementation details are included below.

6.2.3 Describing the Appearance of Landmarks

We use Support Vector Machines (SVM) (Vapnik, 1995) to detect landmarks in the scene. A function is defined to train a linear SVM from observations of the landmark, using Liblinear (Fan et al., 2008). The function accepts a vector of training images $\mathbf{I}$ and corresponding vector of landmark locations in the image $\mathbf{u}$ as input. Image patches corresponding to the landmark locations in $\mathbf{u}$ are extracted and labelled as members of the positive class. The SVM attempts to fit a decision boundary between the positive and negative classes in feature space. This decision boundary models the appearance of the landmark as a function of the surrounding environment. Since we only require the landmark to be locally distinct, we only supply negative training samples from the local environment around the landmark by randomly sampling the remainder of the image, as shown in Figure 6.3b. To improve the robustness of the classifier, we augment the training data using synthetic observations of the landmark (Krizhevsky et al., 2012). In our implementation, the images in $\mathbf{I}$ are darkened, lightened and blurred, as shown in Figure 6.3f.

Linear SVM classifiers require a feature representation on which to act. Histograms of Orientated Gradient (HoG) features (Dalal and Triggs, 2005) are commonly used for robust object detection in images, in particular within the field of pedestrian detection. The method transforms an input image into a feature representation whereby the occurrence of particular gradient orientations within local regions of the image define the feature response. McManus et al. (2015) use HoG features in their work.

The techniques described in this section are agnostic to the underlying feature type, however we found that Aggregated Channel Features (ACF) (Dollár et al., 2014) performed better than HoG features. Figure 6.4 illustrates how a 3-channel RGB image is converted to a 10-channel ACF image. The 10 channels in the ACF image correspond to:

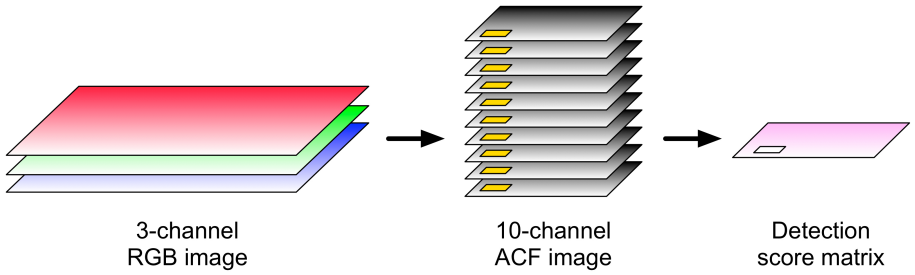


Figure 6.4: Diagram showing how a landmark is detected on a test image. The test image consists of a 3-channel RGB image. The landmark is represented as a 10-channel filter (shown in yellow), made up of the weights from the linear SVM training process. The test image is converted to a 10-channel ACF image format, which also reduces the dimensions of the image by a factor of 4. The filter is convolved with the ACF image to generate the matrix of detection scores.

- Normalised gradient magnitude (one channel)
- Histogram of oriented gradients (six channels)
- LUV colour channels (three channels)

Furthermore, an RGB image patch of size $x \times y \times 3$ in pixels corresponds to an ACF patch of size $\frac{x}{4} \times \frac{y}{4} \times 10$.

ACF features contain gradient information like HoG, but incorporate colour information as well. In our application, we train place-specific detectors, so incorporating colour into the feature representation can make detecting the object significantly easier. For example, a landmark detector trained to detect a red post box will gain a lot of information by modelling the colour, as well as the structure, of the post box. Usually, modelling the colour of objects is brittle due to changes in illumination, however we account for this variation by generating synthetic images of the landmark, for example by darkening and lightening the image. The intuition behind this is that under most outdoor lighting conditions, the colour of the post box will change between different shades of red (and not suddenly green) – this spectrum of possible colour can be learnt by the SVM classifier.

Landmark detections are performed using convolution. The process is illustrated

in Figure 6.4. The query image is converted from a 3-channel RGB image to a 10-channel ACF image (Dollár et al., 2014). The linear SVM detector is stored as a 2D filter with 10 channels, so that it corresponds to the feature space used by the ACF image. This filter is convolved with the ACF image to generate an output matrix of detection scores.

Since the training method requires landmarks to be locally unique, the detector returns only the maximum detection score and corresponding image co-ordinates (i.e. multiple detections of a single landmark in the same image are not allowed). If detection scores are unanimously low across the image, the landmark is declared not visible and no data association is made. A relative lowly detection score threshold (less than zero) is maintained, since false positives (incorrect data associations) will be rejected downstream by the pose optimisation.

6.2.4 Estimating Landmark Positions

The position $\mathbf{X_i}$ of a landmark l_i is modelled as a homogeneous co-ordinate vector $\mathbf{X_i} = (x, y, z, w)^T$ with corresponding Cartesian co-ordinates $(\frac{x}{w}, \frac{y}{w}, \frac{z}{w})^T$. Landmarks can be modelled at infinity by setting $w = 0$. The robot's ability to estimate the depth of landmarks far away is a function of the baseline of the stereo camera. During localisation, landmarks which are modelled at infinity constrain the vehicle's orientation, but not its position.

Chapter 2 described how the position of a landmark observed by a stereo pair can be determined by triangulation. Due to the use of large patches of the image, rather than accurate point features, we cannot simply use triangulation as before. Rather, the system observes the landmarks over a number of consecutive frames where the relative motion of the vehicle is known from visual odometry. The position of the landmark is determined by minimising the re-projection error of the observations

using a non-linear least squares solve with error function $\arg\min_{\mathbf{X}_i} \sum_i \|\mathbf{z}_i - \mathbf{z'}_i(\mathbf{X}_i)\|^2$, where $\mathbf{z}$ is the observation of the landmark in image I_i, and $\mathbf{z'}_i(\mathbf{X}_i)$ is the landmark i projected into image I_i as a funtion of $\mathbf{X}_i$. If the optimisation converges successfully with no observations marked as outliers, and a low RMS re-projection error, the position of the homogeneous co-ordinate $\mathbf{X_i}$ is returned. We note that these rejection criteria enforce the behaviour of a 3D landmark. Importantly, the training algorithm is not sensitive to these parameters. A conservative threshold is set so that only the grossly incorrect landmarks are rejected in order to reduce unnecessary processing during localisation. Landmarks which pass the checks, but which are modelled poorly, are rejected by RANSAC during localisation.

6.3 Localisation Using Patches

This section describes how localisation is performed, using the landmarks extracted in the previous section. Recall that a landmark is defined as $l_i = \{\mathbf{d_i}, \mathbf{X_i}\}$, where $\mathbf{d_i}$ is the weight vector output from the linear SVM classifier training, and $\mathbf{X_i}$ is the 3D position of the landmark. The detector $\mathbf{d_i}$ models the appearance of the landmark and is used to recognise the landmarks in the live image. The previous section described how the detector is simply a linear filter which is convolved with the live image to generate a matrix of detection scores. The location of the maximum detection score is taken as the location of the landmark, since the training algorithm ensured that each landmark was locally unique (multiple instances of the same landmark are not permitted). Since each landmark is associated with a landmark position $\mathbf{X_i}$, this implicitly performs the data association between image co-ordinates in the live image and 3D landmark positions in the map.

The pipeline for localisation 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, 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 above. 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.

3. **Do pose optimisation**. Optimise for pose in a similar manner to traditional point-feature localisers, as discussed in Chapter 2.

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 Churchill and Newman (2013); McManus et al. (2015). This prevents poor localisation estimates propagating through the system.

In the following section, we test this localisation method under a number of challenging conditions.

6.4 Results

Localisation performance of the proposed method is compared to a traditional point-based alternative. The experiment is performed over 205 km of data, consisting of 22 traverses of the Central Oxford dataset. Data was collected under a number of challenging appearance conditions, including bright sun, cloud, snow, and at all

Figure 6.5: Sample images from 205 km of data from the Oxford 10 km dataset. Each image is taken from one of the 22 logs used in this paper, illustrating the challenging weather and lighting conditions present in the dataset.

times of the day and night. Figure 6.5 shows the extreme appearance change being considered and Table 6.1 categorises the logs by weather and time of day.

6.4.1 N-vs-N experiment

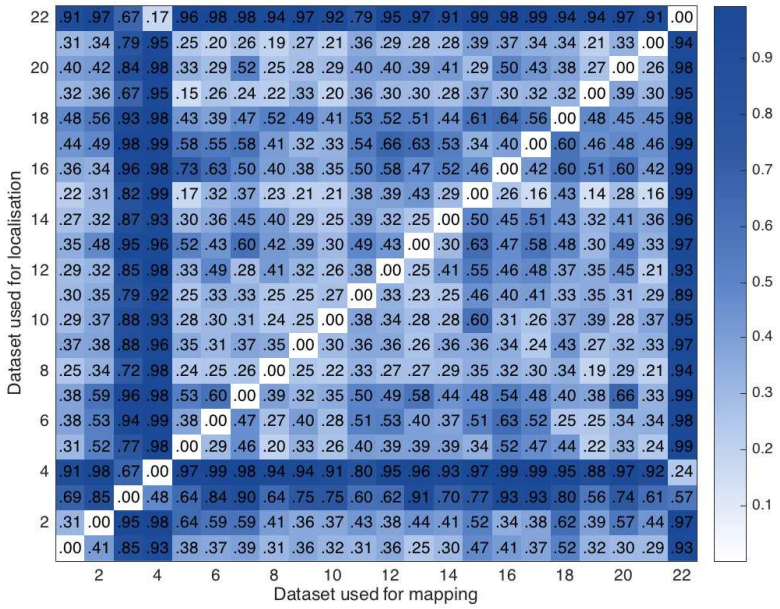
Our main experiment consists of an exhaustive N-vs-N comparison of the proposed method to a traditional point-based localisation system. As in the previous chapter, the point-based localisation system uses BRIEF features to obtain data associations for localisation. This experiment tests the robustness of the localiser under a range of appearance change conditions in an urban environment. The experiment is performed as follows. 22 maps are prepared using each of the 22 logs from the Central Oxford dataset. For each log, an independent map is created according to Section 6.2. These maps are topometric, where landmarks are stored on a place-dependent basis, and transformations link places locally. Within the context of experience-based mapping (Chapter 4), these maps can be thought of as each

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

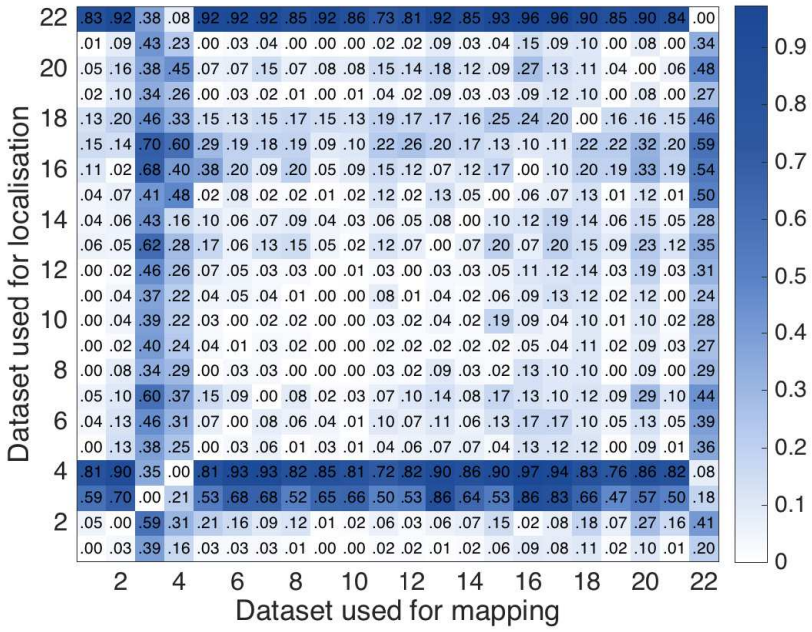
Table 6.1: Table categorising the 22 logs used for evaluation by their weather conditions and time of day. Sample images are shown in Figure 6.5.

containing a single experience.

For each map, localisation is performed against that map using all 22 traverses of the route (we include the case where the same log is used for mapping and localisation). As in the previous chapter, localisation success is determined by comparing the sequential localisation estimates with respect to the estimates from visual odometry. For each localisation experiment, the measure of performance is the probability of travelling further than 20 m in open loop. As an approximate estimate, we expect the visual odometry estimate to drift 1% per distance travelled in open loop. This would correspond to a 20 cm error over a 20 m localisation failure.



(a) Traditional point-feature localiser

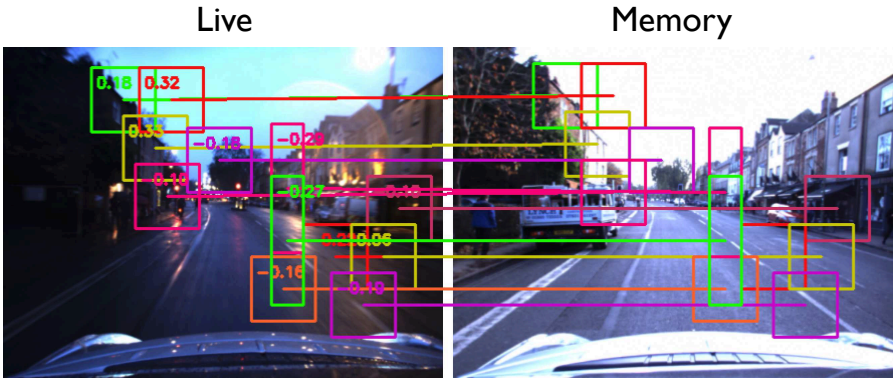


(b) Proposed system

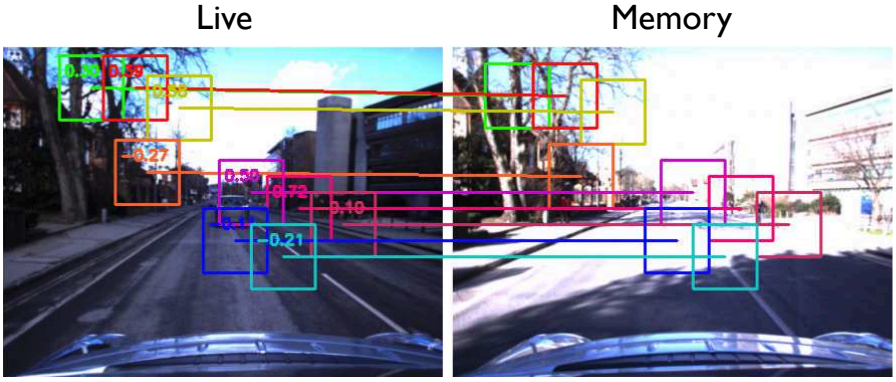
Figure 6.6: Using 22 traverses of the Oxford 10 km route, 22 independent maps (listed along the x-axis) are built. For each map, localisation is performed using all 22 traverses in that map (shown on the x-axis). The value in each cell is the portion of the localisation run in which localisation failed 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.

Figure 6.6 presents the results of this experiment. Figure 6.6a shows the performance of the baseline system, a traditional point-feature localiser, and Figure 6.6b shows the results of the 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 Log 1 was used to create a map, and Log 8 was used to localise against that map. The diagonal $x = y$ corresponds to the event where the same log is used for mapping and localisation. The figure shows that localisation performance is consistently more robust using our proposed method. This is an important result, and it is one of the key factors which makes our localisation system robust. This is discussed in more detail in the following chapter. Snapshots of the feature matches are shown in Figure 6.7, illustrating the extreme conditions under which the localisation system is performing.

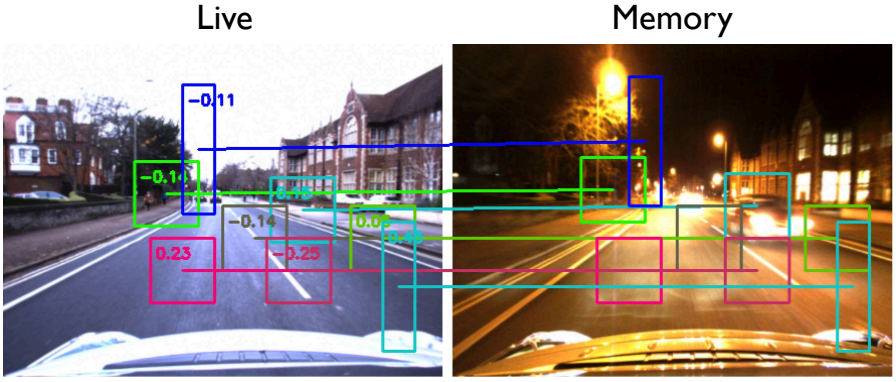
In Figure 6.6, it is clear that there is structure in the matrices. This implies that certain logs performed consistently better (or worse) than others during localisation and mapping. We group the logs in various combinations according to Table 6.1 and record the corresponding mean performance in Table 6.2. The mean is calculated by aggregating the respective matrices from the N-vs-N experiment in Figure 6.6 (except that the diagonal elements are excluded from the calculation since these represent the case where the same log is used for mapping and localisation). These results are explored in more detail below.



(a) Day and early evening



(b) Harsh lighting conditions



(c) Day and night

Figure 6.7: Examples of successful localisation between the live image (left) and an image from memory (right). These examples were chosen to demonstrate the extreme levels of appearance change the method is able to handle. To make the visualisation clearer, only a subset of data associations are shown.

Datasets	Baseline	Proposed	Improvement Factor
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 6.2: 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 6.6, except that we exclude the diagonal $x = y$ where the same log is used for mapping and localisation.

6.4.2 Localisation During the Day

This section analyses localisation performance during the day. Table 6.2 shows that when only considering logs captured during the day, 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 6.7b shows successful localisation under harsh lighting conditions during the day.

However, our system does not only show improvement when severe appearance change is present. Table 6.2 shows the average localisation performance over a set of logs captured in cloudy conditions. Cloudy conditions provide the most favourable conditions for the traditional point-feature localiser since they are the most visually similar (no shadows or direct sun). In spite of 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 Figure 6.6b, a number of the localisation runs completed with zero localisation failures.

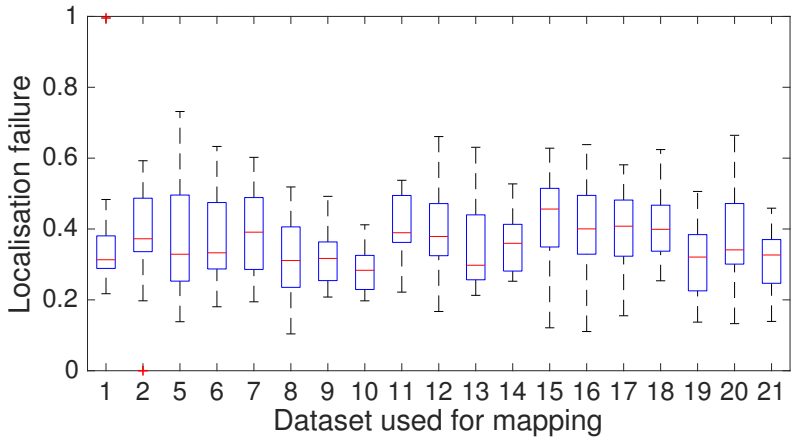
6.4.3 Localisation Between Night and Day

Consider the matrices in Figure 6.6. The matrix of the traditional localiser is roughly symmetric: Mapping with Log A and localising with Log B results in similar performance to the converse of mapping with Log B and localising with Log 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 logs are Logs 3, 4, and 22.

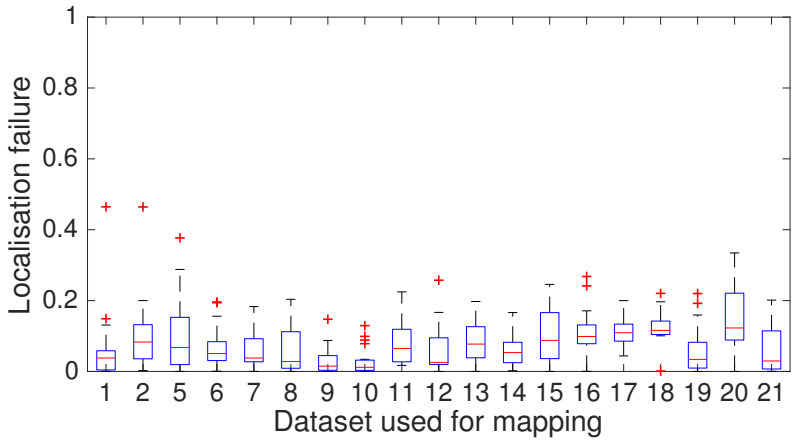
However, the matrix corresponding to the proposed localiser is not symmetric. When the map is created during the day, and localisation is performed at night, localisation performs poorly in the same way as the baseline system. Interestingly however, mapping during the night and localising during the day results in a reduction in the percentage of localisation failure from approximately 90% to 40%. Figure 6.7c shows an example of successful data associations between day and night. While localisation still fails for a large portion of the route, it remains a significant result. 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.

6.4.4 Which Logs Are Best for Mapping?

Figure 6.8 plots the distribution of the localisation results on a per-map basis for logs during the day. We assert that the average localisation performance on a given map is an indicator of underlying map quality. From Figure 6.8, cloudy datasets (Logs 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



(a) Traditional point-feature localiser



(b) Proposed system

Figure 6.8: Boxplot showing the distribution of the localisation results on a per-map basis for daytime logs. The median results are marked with horizontal red lines, the first and third quartiles are shown in blue, and the maxima and minima are shown in black. Outliers are marked with red crosses. The plot shows how better localisation performance is observed when the map is created under cloudy conditions (Logs 1, 9, 10, 19, and 21). This is likely due to the absence of harsh lighting conditions, allowing a camera with limited dynamic range to expose and capture the scene completely.

cannot capture. Table 6.2 shows the mean localisation performance when creating a map with a cloudy log, and localising in that map using all of the daytime logs (i.e. including the logs with sun and snow). Under these conditions, the average localisation failure of our proposed system is reduced from 8.5% (when any daytime log is used for mapping) to 5.5% (when only cloudy logs are used for mapping).

6.4.5 Lateral Translation

We have assumed in this chapter that localisation failure is caused mainly by changes in weather, lighting and season. However, another cause of localisation failure is the lateral translation of the vehicle across the road. Point feature techniques are viewpoint dependent, so movement across the road introduces additional appearance change.

While we have not specifically investigated this as a variable in localisation performance, we suspect it is a contributing factor in the stark difference in localisation performance between the traditional and proposed localisation systems. Point features have a limited invariance to translation, which is further reduced under harsh appearance change. Conversely, we have found that the proposed landmark detectors are more robust to translation. This is likely due to the training algorithm, where multiple observations of the landmark are used to train the linear SVM classifiers. In our implementation, we use landmark observations from both cameras in the stereo pair, as well as from multiple camera frames as the vehicle moves through the environment. In so doing, we are able to train translation invariance into the landmark model.

Figure 6.9: The output from a single localisation run, where a sunny log was used to create a map and a cloudy log 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.

6.4.6 Applicability to the Experience Framework

Previous chapters described the use of experiences to model change in the environment. A traditional point-feature localiser was used which was brittle to appearance change. This meant that the system required a high number of experiences to map the environment. From another perspective, this meant that 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 see strong evidence that the framework of experiences is beneficial even with a more robust localiser. Figure 6.9 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 log for localisation, that localisation failures occur in *different*

areas of the map. This means that an approach using multiple experiences would likely result in improved localisation performance. We explore this in more detail in the following chapter.

6.5 Summary

This chapter has presented a new technique for performing robust localisation under the presence of extreme appearance change in the environment. Previous chapters have motivated for the use of the experience paradigm in adapting to appearance change in the environment. However, some applications may require autonomy in a short space of time, before the vehicle has been able to survey the environment under a sufficiently large number of appearance conditions. This need for many experiences is a problem caused by the localiser – a brittle localiser requires many experiences to model the full range of appearance conditions.

This chapter describes an algorithm for extracting and modelling distinctive mid-level landmarks from a single experience using a richer, more robust representation. These landmarks are extracted by applying inexpensive geometric checks for consistency in landmark position, and are described using linear SVM classifiers with training data from a single pass through the environment. These landmarks are used at run-time to perform robust, metric localisation across extreme appearance change. This enables robust localisation with fewer experiences, and increases graph connectivity by the addition of more loop closures between experiences.

This 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. Importantly, we see strong evidence that the experience-based mapping framework would further

improve localisation performance. As a result of this, the following chapter presents a new vision-only localisation system called Dub4. The system draws on many of the techniques and algorithms discussed so far, including the experience-based paradigm and the higher-level visual features described in this chapter. We show how these techniques play a pivotal role in the development of a robust, vast-scale localisation system.

Chapter 7

Dub4: A Vision-Only Localiser

7.1 Introduction

This chapter presents a complete vision-only localisation system called Dub4. The goal of this system is to perform robust, real-time localisation, *"wherever, whenever, whatever the weather"*. This is a particularly challenging problem in outdoor environments, where sudden and unpredictable changes in weather, lighting and season cause traditional localisation systems to fail. The problem of appearance change has meant that many existing autonomy solutions in the literature have avoided the use of cameras for localisation, in spite of their low cost and widespread availability. In this chapter, we describe and validate a new localisation system, showing that it is able to perform reliable, camera-only pose estimation over vast scales and a diverse range of environmental conditions.

Dub4 draws on a suite of complementary algorithms to produce a single, robust localisation system. It embraces the experience-based paradigm described in Chapters 4 and 5, allowing the robot to perform ad hoc map updates as it traverses the environment. However, while multiple experiences are permitted, significantly fewer experiences are required to perform robust localisation due to a number of key

improvements in the core localisation subsystem. The localiser draws together the place-specific patch detectors described in Chapter 6, multiple feature types (BRIEF and SURF), and an illumination invariant image transformation which acts to diminish the effect of shadows in the scene. In concert, these subsystems work together to deliver a vision-only outdoor localiser which is robust in the face of changing weather, lighting, season, and scene structure.

To support the design and testing of this system, over 1000 km of data has been collected from a 10 km route around Oxford, over a period of 18 months. As discussed in Section 3.1, this dataset is unique in that it contains 100 repeats of an approximately 10 km route in a busy city, performed at regular intervals throughout the year. Data was collected at all times of day and night, under snow, rain, sun, cloud and fog, and through areas subject to significant construction and road works. This challenging dataset has played a crucial role in the development of this system, exposing corner cases and rare events which are simply not encountered on smaller datasets.

As well as the 1000 km Central Oxford dataset, the system is validated on a further 60 km dataset from Milton Keynes, and a 200 km off-road dataset from Cornbury Park. The latter takes place in a forested area where only natural landmarks such as bushes and trees are available for localisation.

In addition to the extensive offline testing presented here, Dub4 has served as the primary localisation source for a number of autonomous vehicles and projects. These vehicles are pictured in Figure 7.1 and include:

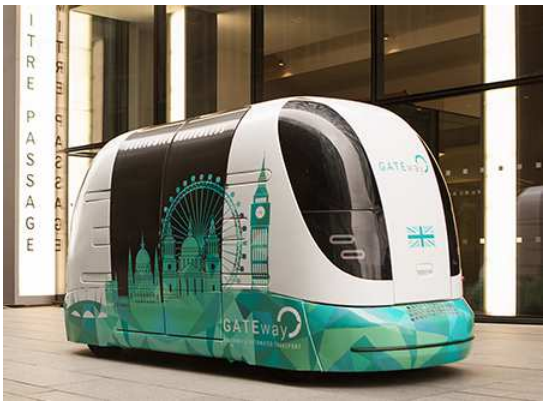
1. The Oxford RobotCar, a modified Nissan LEAF. Successful trials have been conducted regularly since late-2015.

2. The first autonomous vehicle entry in the 2016 Shell Eco Marathon in London. This was a track-based event, where the road was only created the day before

(a) Oxford RobotCar

(b) Shell Eco-Marathon

(c) LUTZ PathFinder Pod



(d) GATEway Pod

Figure 7.1: This chapter describes a new vision-only localiser called Dub4. It currently acts as the primary localisation source on the autonomous vehicles pictured here. These projects include the Oxford RobotCar, the Shell Eco-Marathon, the LUTZ PathFinder Project, and the GATEway Project.

the first public demonstration of the system. Not having the environment available beforehand placed a significant demand on the system's ability to act as a turnkey autonomy solution.

3. The Greenwich GATEway project, a major public demonstration of autonomous technology in London. Eight pods will operate autonomously for 40 hours per week over a period of months, transporting members of the public between various destinations.

4. The LUTZ PathFinder project, a public demonstration of autonomous technology in Milton Keynes.

The remainder of this chapter describes, tests and evaluates this system in more depth.

7.2 System Overview

This chapter draws together a number of techniques and algorithms, some of which have been described in previous chapters. A summary of the various components is presented below, with reference to the diagram in Figure 7.2.

Visual Odometry

Visual odometry operates on a pair of stereo images to estimate the robot's motion in open loop, as described in Chapter 2. Visual odometry is used for a number of different purposes. Firstly, it is used during mapping to create an experience, providing the relative transformations between consecutive images in the experience. Secondly, visual odometry is used to update the robot's pose estimate in the map during periods of localisation failure. Lastly, a reliable source of ego-motion is used to reject inaccurate pose estimates. Robust visual odometry is a key component in our localisation system.

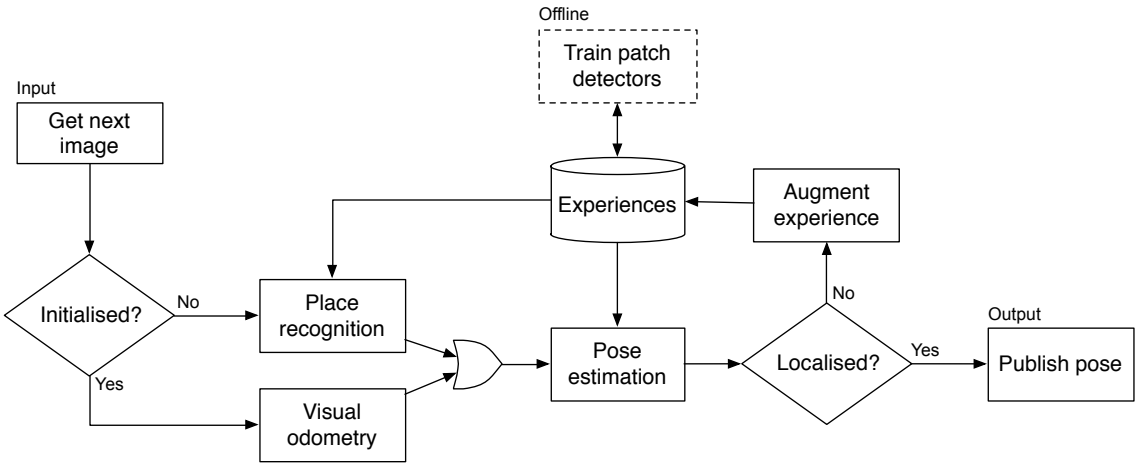


Figure 7.2: Diagram showing a high-level overview of the various components in Dub4, for a single pose estimation attempt.

The Experience Graph

Dub4 is founded on the principles of experience-based mapping, where multiple representations of the world are accumulated to make localisation more robust. Chapters 4 and 5 demonstrated that a high degree of localisation performance could be achieved using this technique. However, a drawback of this approach was that many experiences were required to generate a complete model of the environment. The localisation system described here includes a number of significant improvements to the localiser, meaning that fewer experiences are required to perform robust localisation.

We note that even if the core localiser were "perfect", in the sense of being truly invariant to lighting, weather and seasonal changes, the experience paradigm would still perform the valuable function of automatically updating the map when faced with large-scale structural change in the environment. While a significant portion of this thesis is dedicated to stretching the performance capabilities of the core localiser, we regard the use of multiple experiences as a crucial factor in the long-term performance of the system.

Place Recognition

Place recognition is a type of localisation which outputs a topological (rather than metric) estimate of the robot's location within the experience graph. As discussed previously, place recognition is used to seed the pose estimation techniques described below. We use FAB-MAP (Cummins and Newman, 2008) to perform place recognition. FAB-MAP is an appearance-based probabilistic technique for recognizing places which the robot has previously visited. It is robust to visual aliasing in the environment, and is able to detect when the robot visits a new area that is not already part of the map. FAB-MAP is used to perform initialisation in the experience graph, as well as re-localisation if the system becomes lost.

Pose Estimation

The goal of pose estimation is to output metric 6-DOF pose with respect to a node in the experience graph. A core contribution of this chapter is a localiser which draws on a suite of complementary algorithms to perform robust data associations across challenging environmental conditions. We use multiple feature types to perform data associations, an illumination invariant image transformation to reduce the effect of shadows, and mid-level patch detectors to provide coarse data associations across extreme appearance change. These techniques are described fully in the following section.

7.3 Robust Localisation

This section describes the techniques used to perform robust pose estimation in an experience graph. We describe a set of complementary algorithms which each contribute to the system's robustness under different conditions. Figure 7.3 presents an overview of the system. These multiple techniques operate in parallel to obtain data associations between observations in the live image and 3D landmarks stored

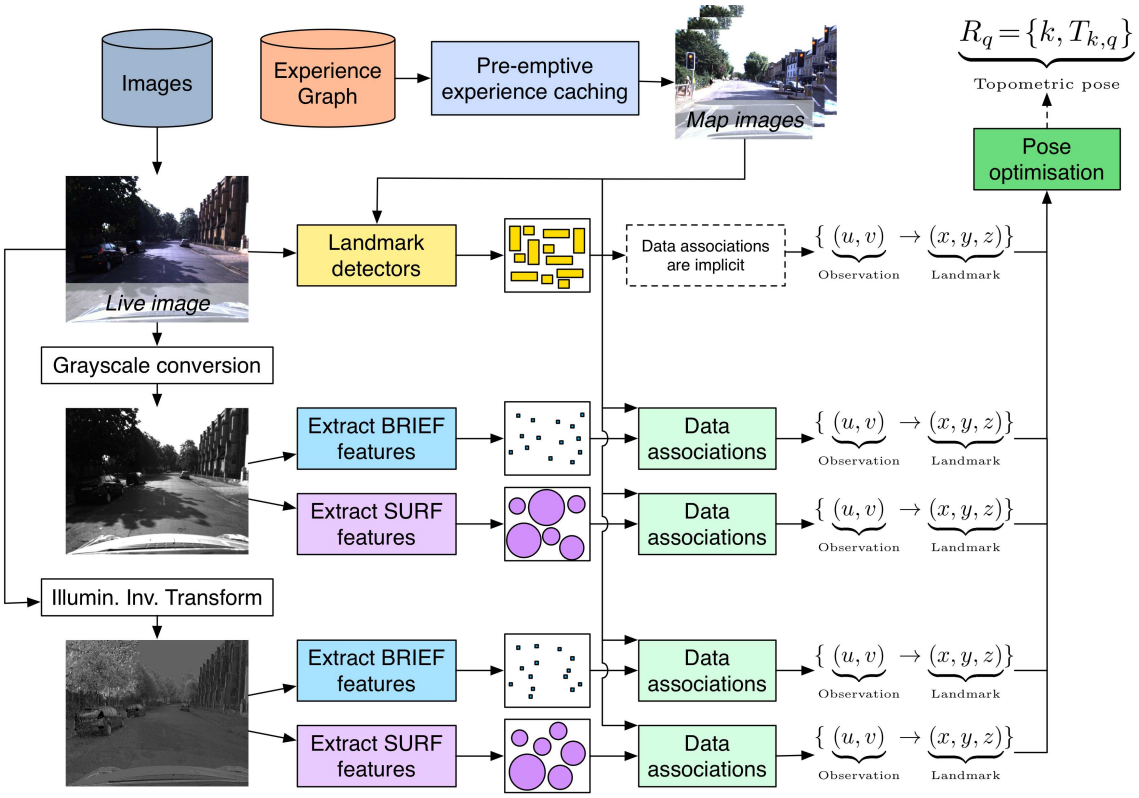


Figure 7.3: Block diagram showing the pipeline for pose estimation. The live RGB image is converted to grayscale and an illumination invariant representation. Features are extracted from both images and associated with features stored in the experience graph. In parallel, a bank of landmark detectors is loaded from the same experience and fired on the live RGB image. The data associations consist of a 2D observation in the camera frame and corresponding 3D landmark in the map. These are input to the robust pose optimisation which determines the 6-DOF pose of the robot with respect to a node in the experience graph.

in the map. These data associations are output to a single optimisation for pose, where the non-linear least squares optimisation described in Chapter 2 is used to solve for the camera pose. The techniques used to obtain these data associations are explored in more detail below.

7.3.1 Pre-emptive Experience Caching

A core feature of the localisation system is the ability to localise in a map containing multiple experiences (i.e. the experience graph). However, as more experiences are

added to the experience graph, the computational cost of localisation at run-time increases too. This is because the localiser may have to attempt localisation in a high number of experiences before achieving a successful match. We use the probabilistic method presented in Chapter 5 to pre-emptively cache nodes in the experience graph which have a high probability of resulting in successful localisation. The technique is unsupervised and uses the localisation history of the robot to make predictions about future localisation attempts. Figure 7.3 shows this subsystem querying the experience graph for training data, outputting the best N map images for localisation. This subsystem ensures that the localiser continues to perform efficiently as the experience density increases.

7.3.2 Point Features

A traditional approach to pose estimation uses point features to associate observations in the live image with landmarks stored in the map. Chapter 2 described how a feature detector is used to identify distinctive keypoints in an image, while a feature descriptor is used to describe those features. For example, FAST (Rosten and Drummond, 2006) is a feature detector, while BRIEF (Calonder et al., 2010) is a feature descriptor. Point features are brittle to appearance change in outdoor environments. However, the difficult data association problem is avoided by modelling the world using multiple experiences. At run-time, the localiser only attempts localisation in experiences which closely match the appearance of the live image. Chapters 4 and 5 demonstrated that this technique can achieve a notable level of performance using only simple point feature techniques.

In our work, we have found that the relative performance gains of different point features is marginal, and that there is no single "best" feature type. Rather, we have found that some features are better suited to particular environments than others. We have found that FAST / BRIEF works well in urban environments, which

can be attributed to the presence of buildings and urban structures which contain many distinctive corners. Conversely, we find that SURF performs better in off-road environments, likely due to it being an example of a blob detector rather than a corner detector.

However, many environments do not fall into this binary classification of urban and rural environments. For example, large trees and bushes often line the streets in urban environments too. Rather than attempting to choose the "best" feature type, the system uses multiple feature types in parallel. Recall that we use features to provide data associations between the live image and 3D landmarks in the map. A data association is simply the correspondence:

$$(u, v) \leftrightarrow (x, y, z)$$

which relates the pixel co-ordinates (u, v) of observations in the camera image, to the position (x, y, z) of a 3D landmark in the map. Importantly, the data association is agnostic to the feature type or method used to determine the correspondence. Figure 7.3 shows how multiple data association tasks are run in parallel, outputting to a single pose optimisation. We find that this significantly increases the range of conditions over which we are able to localise.

7.3.3 Illumination Invariance

Robustness to shadows is particularly difficult to achieve (Churchill and Newman, 2013; McManus et al., 2014; Maddern et al., 2014b; Paton et al., 2015a). Shadows introduce harsh gradients and regions of high contrast into the image, for which feature descriptors are not invariant. Additionally, feature detectors are also prone to failure under these conditions, as keypoints are detected on the shadows rather than on the structure in the environment. The effect of this can be thought of as intro-

(a) Day

(b) Day

(c) Night

Figure 7.4: Examples showing how the illumination invariant transform converts an RGB image (left) to an illumination invariant image (right). In Figures (a) and (b), the shadows in the RGB image are successfully removed from the illumination invariant image. This enables data associations to be made in regions of the image which were previously obscured by shadow. This transformation assumes that the source of light is a black body illuminator, an assumption which is not valid at night. An example of the illumination invariant transformation at night is shown in Figure (c).

ducing "phantom structure" into the environment. An experience-based approach (Chapter 4) models the world using a set of overlapping, discrete experiences. However, shadows change continually with the angle of the sun. In some situations, it is simply not possible to capture the full spectrum of appearance change under these circumstances. However, even if it were possible, the high number of experiences required would be an inefficient way of modelling the world. This presents a problem for long-term autonomy.

The field of colour constancy assists in this challenging scenario, where colour constancy techniques aim to determine the colour of objects independently of external illumination. Since shadows are simply variations in external illumination, a colour constant image is one that does not contain shadows. Section 2.3.6 described how a RGB image could be transformed into a single-channel illumination invariant image. Examples of the resulting illumination invariant images are shown in Figure 7.4.

In addition to the point feature data associations obtained from grayscale images, the system also mines data associations from the illumination invariant images. The block diagram in Figure 7.3 shows how BRIEF and SURF features are extracted from the illumination invariant images and used to perform data associations between the live image and an image from the map. These point feature techniques run in parallel to those which act on grayscale images. Together, the data associations from both the grayscale and illumination invariant images are output into the pose optimisation.

Importantly, we find that the features detected in grayscale images are *different* to those detected on illumination invariant images. Figure 7.5 shows that the features detected in the illumination image occur in different regions to the grayscale image. This is often as a result of the illumination invariant transformation revealing distinctive features on the underlying scene structure which would otherwise have

been hidden by harsh shadows. This is important, as a spread of features over the frame improves the accuracy of the pose optimisation.

7.3.4 Patch Detectors

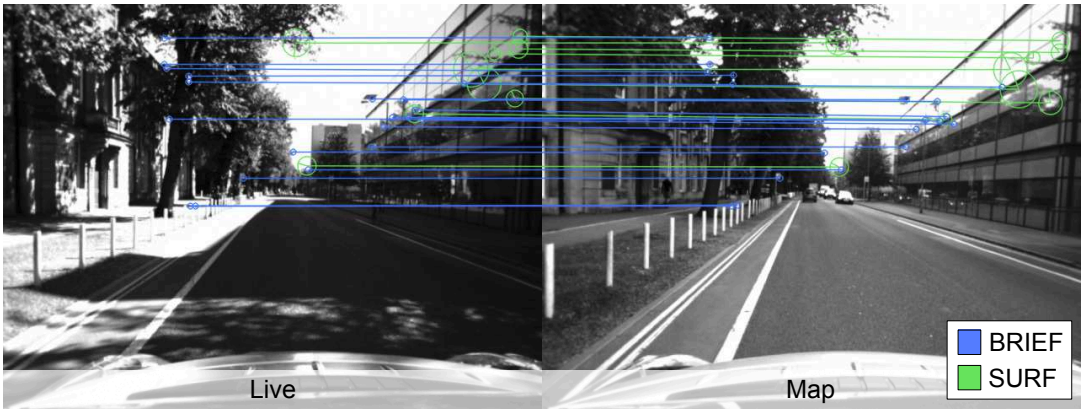
In spite of the performance gains described above, point feature approaches remain brittle to appearance change. Robustness is obtained by storing a high number of experiences in the map, but this may not always be feasible. The underlying cause of this problem is that point features have a limited ability to generalise beyond the data they have been shown. For example, no matter how many times the vehicle has traversed an environment in rainy conditions, localisation will likely fail on the first visit to the environment under sunny conditions.

Chapter 6 presented place-dependent patch detectors as an alternative to traditional point features. While point features describe low-level elements in the scene which are susceptible to changes in appearance, patch detectors use larger, distinctive landmarks in the scene for localisation. The appearance of the image regions are modelled using linear SVM classifiers, such that each landmark is associated with a bespoke, place-dependent detector.

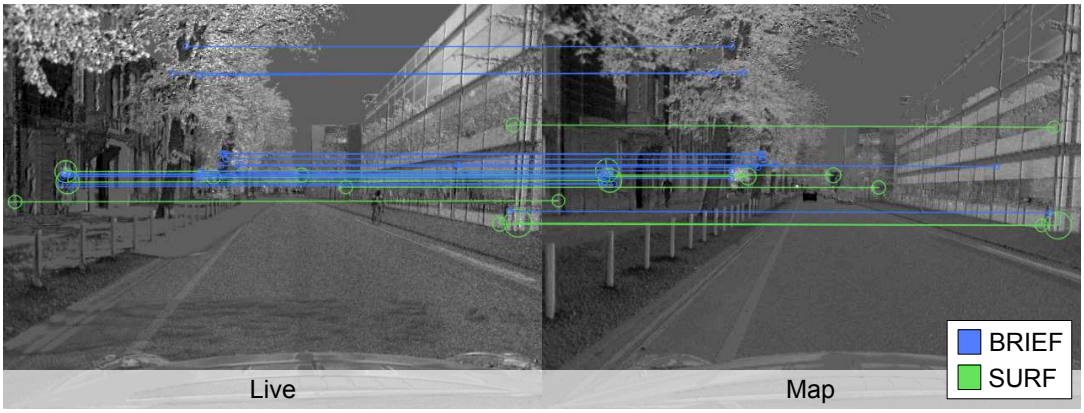
The training algorithm for patch detectors was described in Section 6.2. The method used a stream of stereo images from a single experience to extract and model 3D landmarks, using relative motion constraints between images from visual odometry. These images and relative motion constraints are already stored in the experience graph, so there is little additional overhead with regard to data structures and management tools. The system diagram in Figure 7.2 shows that the training method is performed as an offline process which queries the experience graph for training data.

Figure 7.3 shows that in parallel with the point feature techniques described above, the localiser loads a bank of patch detectors and tests them against the live

(a) Landmark detectors on RGB images



(b) Point features on grayscale images



(c) Point features on illumination invariant images

Figure 7.5: A single matching attempt between the live image (left) and image from the map (right), corresponding to T=13 s in Figure 7.6. Figure (a) shows the landmark detectors performing data associations using large, distinctive elements in the scene. Figure (b) shows BRIEF and SURF feature matches – note how BRIEF features correspond with corners, and SURF features correspond with blobs. In Figure (c), the illumination invariant transform removes shadows, allowing features to be detected in new regions of the image. All of these detections are input into a single pose optimization, as shown in Figure 7.3.

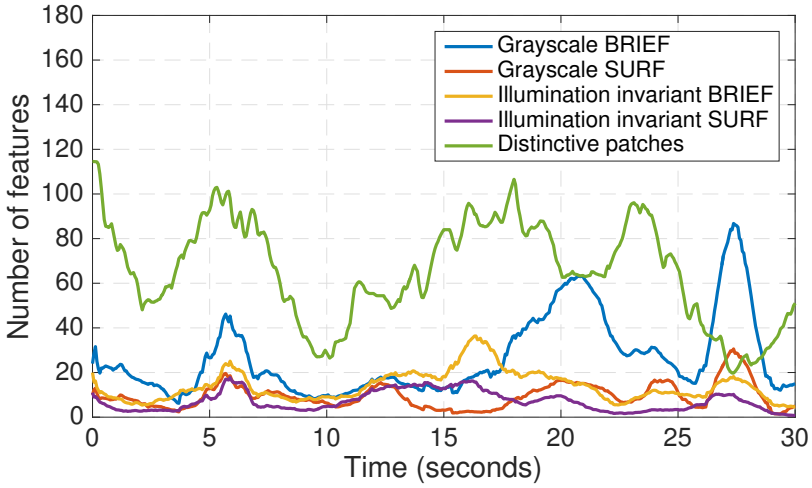


Figure 7.6: Graph showing the number of inliers detected against time, for a short segment. The graph shows how the relative performance of the different feature detectors changes over time as the robot drives through different environments. Rather than select a single "best" feature detector, this graph motivates the use of multiple feature detectors running in parallel. Note that where point-features fail completely, ours is a system which naturally falls back to the more robust, mid-level distinctive patches.

image. Since each detector is trained to detect a single landmark in the environment, the top scoring detection is treated as the observation of that landmark. Each patch detector corresponds to a 3D landmark in the map, so the data association between landmark observation (u, v) and landmark position (x, y, z) is implicit (there is no additional data association step to be performed). As above, these data associations are fed to the single optimisation for pose.

While the place-dependent detectors are more robust than the point features, there is a trade-off in accuracy. The patch detectors are required to "behave" in the way a 3D landmark would (in terms of the way they project into the camera frame as the robot moves through the environment), however in reality the patches often correspond to more complex image regions. For example, a patch might contain multiple planes at varying depths in the image. The assumption of a 3D landmark allows us to use existing pose optimisation techniques. However, this is at a slight

trade-off in accuracy. In the non-linear least squares optimisation, we have not adjusted the relative covariances of patch detectors and point feature detectors, but have allowed for different loss functions which apply a stricter penalty on point features over the patch detectors.

Our system is a hierarchical one. When there is little appearance change present in the scene, the brittle point feature techniques provide data associations with sub-pixel accuracy. As more appearance change is introduced, the point feature techniques begin to break down and provide fewer good data associations. Under these conditions, the robust patch detectors continue to output data associations, although the accuracy of the patch detector data associations are reduced due to the $\frac{1}{4}$ scaling introduced by the RGB to ACF conversion (Section 6.2.3).

A subtle, but powerful, interaction between point features and patch detectors occurs when the point features detect only a small percentage of inliers. With point features alone, it would be difficult to discern the minority of inliers from the vast majority of outliers. However, the patch detectors are robust under appearance change, and output a significantly higher proportion of inliers – albeit with slightly reduced accuracy due to the nature of the patches. The high proportion of patch detector inliers helps the pose optimisation to reject poor point feature data associations. This improves the pose estimate accuracy, since point feature data associations are more accurate than the patch detectors. This allows the system to output accurate pose estimates in spite of appearance change, and enables significantly better performance than if two independent pose optimisations, one for point features and one for patches, were performed.

Figure 7.7: Sample images from the Central Oxford dataset, demonstrating the extreme changes in appearance that took place over the 1000 km of data collection.

7.4 Results

This section presents, by far, the largest evaluation of vision-only pose estimation we have found in the literature. Localisation results are presented on over 1200 km of data, collected over a period of two years from three different environments. While the system is actively used on various autonomous platforms, the analysis here focuses on vast-scale offline evaluation due to our ability to isolate the performance of the localisation system independently of the perception, planning, control and other systems required to operate an autonomous vehicle safely. The following provides a detailed analysis of the behaviour and impact of our design decisions to support our current architecture.

7.4.1 Experimental Data

Three challenging datasets are used to validate our system:

1. **1000 km Central Oxford:** The Oxford RobotCar dataset consists of 100 repeats of a 10 km route, recorded over 18 months across a huge variety of

lighting, weather and seasonal conditions. The route traverses through the centre of a busy city, often in heavy traffic. Images are recorded from a forwards-facing Bumblebee XB3 stereo camera. This dataset is unique in the high number of repeated visits to the same place, under a wide variety of conditions. Figure 7.7 demonstrates the variation in appearance encountered.

2. **60 km Milton Keynes:** The Milton Keynes dataset is smaller in size, consisting of six repeats of a 10 km route. The data was collected over three months, predominantly from wide pedestrian or cycle paths (the vehicle used to collect data is approximately the size of a golf cart). The sensors used differ in that two monocular Grasshopper cameras, fixed in a wide-baseline configuration, are used rather than a stereo camera.

3. **200 km Cornbury Off-Road:** The Cornbury Off-Road dataset consists of 20 repeats of a 10 km route in a primarily forested environment in North Oxfordshire. The dataset is subject to significant seasonal change as winter changes to summer, and the overhanging trees draw harsh shadows across the scene. Images are captured using a forwards-facing Bumblebee XB3 stereo camera.

Together, these datasets represent more than 1200 km of driving and present a significant challenge for outdoor, vision-only localisation. These datasets were presented in more detail in Chapter 3.

7.4.2 Vast-Scale Localisation Performance

This experiment tests localisation performance across 1200 km of outdoor driving, under the presence of significant appearance change. As before, we use the cross-validation techniques presented in Section 3.2.3. For each dataset, we create m maps, using e experiences per map. Each experience corresponds to a single traversal, or

log, from the dataset. For each of the maps created, we perform localisation using the remaining l logs (i.e. $e + l$ is the total number of logs in the dataset), such that ml independent localisation experiments are performed.

Since the three datasets are of considerably different size, the following experiments are performed:

1. **1000 km Central Oxford:** 3 maps are created, with 15 experiences per map. Localisation is performed using the remaining 85 logs, for each of the three maps.

2. **60 km Milton Keynes:** 5 maps are created, with 5 experiences per map. A single hold out log is kept for localisation in each map.

3. **200 km Cornbury Off-Road:** 3 maps are created, with 2 experiences per map. The remaining 18 datasets are used for localisation in each map.

An example of successful localisation in the Central Oxford dataset is shown in Figure 7.8. The robot's pose in the experience graph is shown on the left, illustrated by the co-ordinate frame. Each node in the experience graph contains an image, and the two tracks running through the graph correspond to the two experiences in that area. A single thumbnail from each experience is shown at the bottom. The set of data associations that were obtained during localisation is shown on the right, where the live image and map image are shown alongside one another. Patch detectors (yellow) operate on RGB images (top), while BRIEF (blue) and SURF (pink) features are extracted on grayscale (middle) and illumination invariant images (bottom). These data associations are fed to a single optimisation for pose.

Figure 7.9 presents the localisation results of these large-scale experiments. As a result of the cross-validation approach above, this experiment required processing an equivalent of over 3000 km of data. This requires a significant processing effort, and

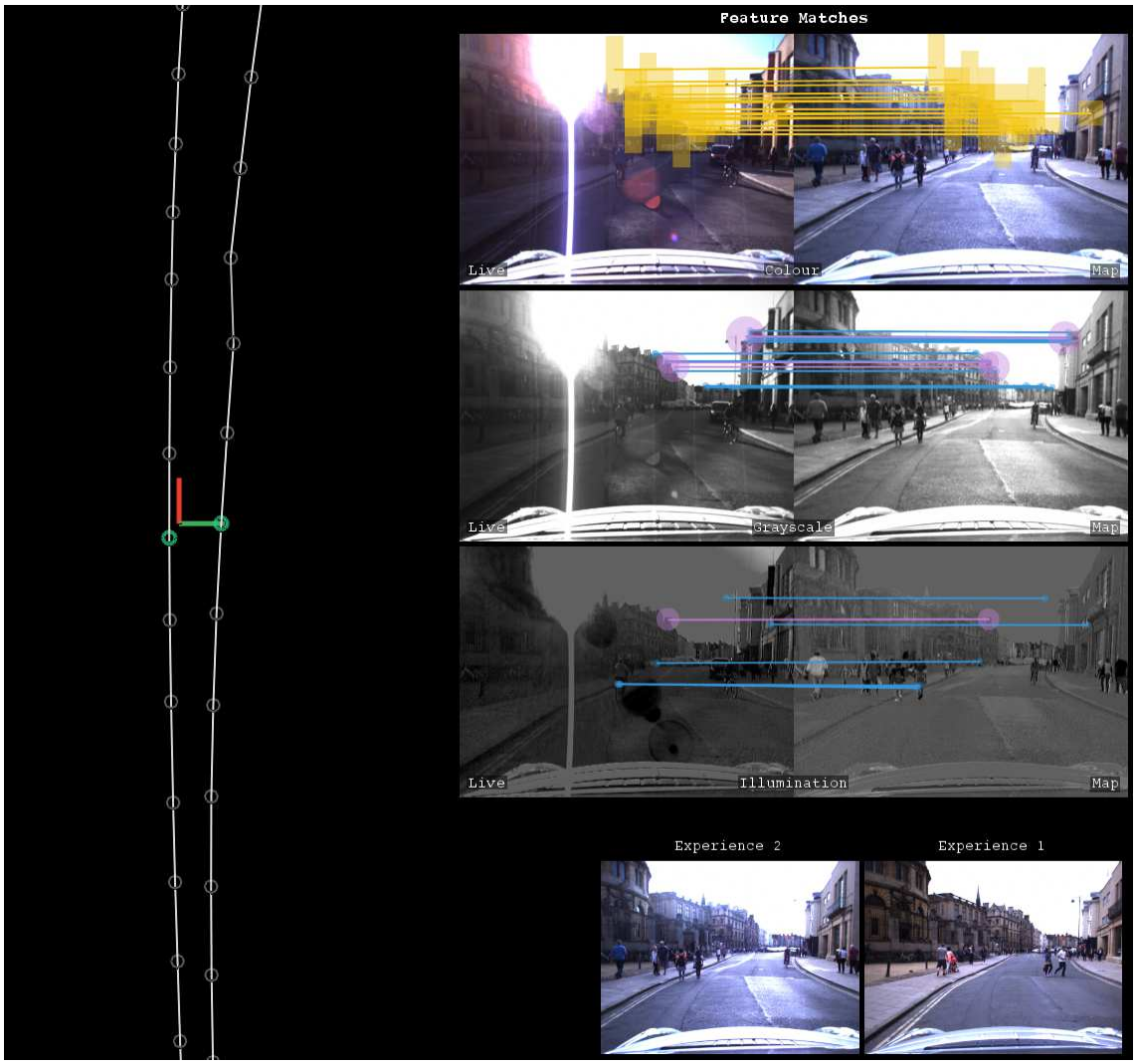


Figure 7.8: Example of successful localisation in the Central Oxford dataset. A top-down view of the experience graph is shown on the left, where the robot's pose is indicated by the co-ordinate frame. The localiser obtained successful localisation with the nodes highlighted in green (two independent matching attempts were made, and were successful). The inlier data associations are shown on the right, where the live image and map image are shown side-by-side. Patch detectors (yellow) operate on RGB images (top), while BRIEF (blue) and SURF (pink) features are extracted on grayscale (middle) and illumination invariant images (bottom). These data associations are output to a single optimisation for pose.

while a small number of topological localisers have demonstrated results over these scales, the experiment performed here is more than an order of magnitude greater in size than any other pose estimation experiment performed in the literature.

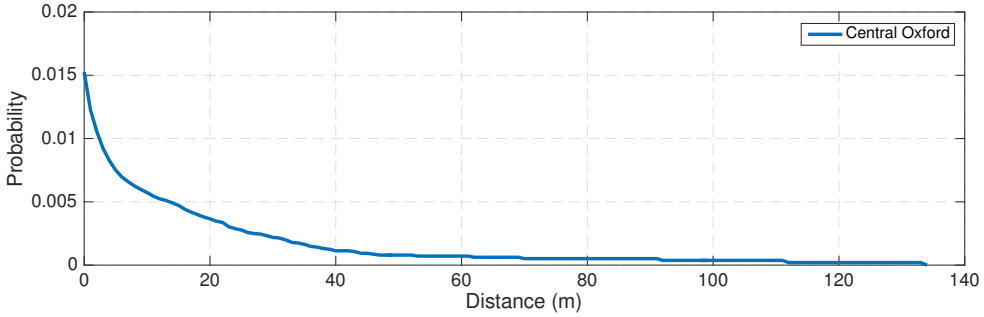
The key metric for localisation performance is the probability of localisation failure over a particular distance. As discussed previously, this is an important metric since during periods of localisation failure, the robot must update its estimate in the map in open loop using visual odometry, which is prone to drift as discussed in Section 3.2. We strive to minimise the distance travelled in open loop. Key results are as follows:

1. **1000 km Central Oxford:** Figure 7.9a shows that in the 1000 km Central Oxford dataset, the probability of failing to localise for longer than 20 m and 100 m is 0.00366 and 0.00037, respectively. The longest failure was 133 m.

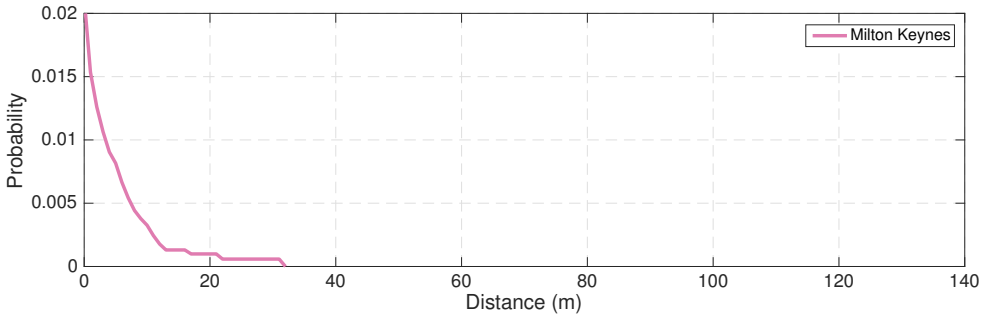
2. **60 km Milton Keynes:** Figure 7.9b shows that in the 60 km Milton Keynes dataset, the probability of failing to localise for longer than 20 m was 0.000992, and that the longest failure was 31 m.

3. **200 km Cornbury Off-Road:** Figure 7.9c shows that over the 200 km Cornbury Off-Road dataset, the probability of failing to localise for longer than 20 m was 0.0009, and that the longest failure event was 29 m.

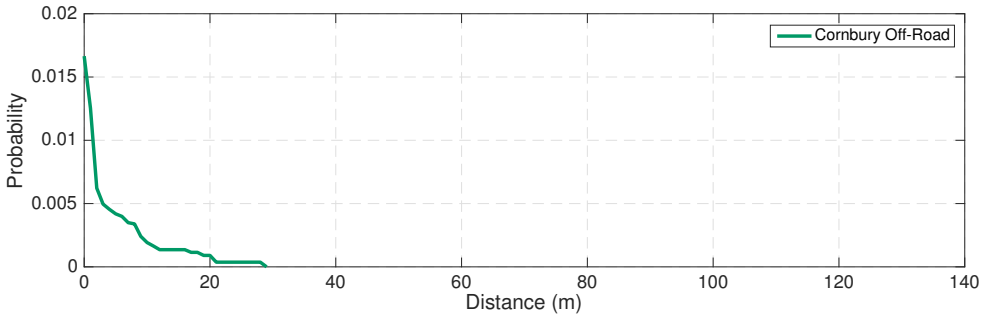
These are significant results given the size and diversity of these datasets. We note that better results were obtained from the Milton Keynes and Cornbury Off-Road datasets, compared with the Central Oxford dataset. This is likely due to the smaller size of these datasets, which are consequently exposed to less diversity in appearance in comparison to the Central Oxford dataset. This highlights the importance of vast-scale testing and validation, since corner cases and rare events may not be encountered in smaller datasets.



(a) Central Oxford

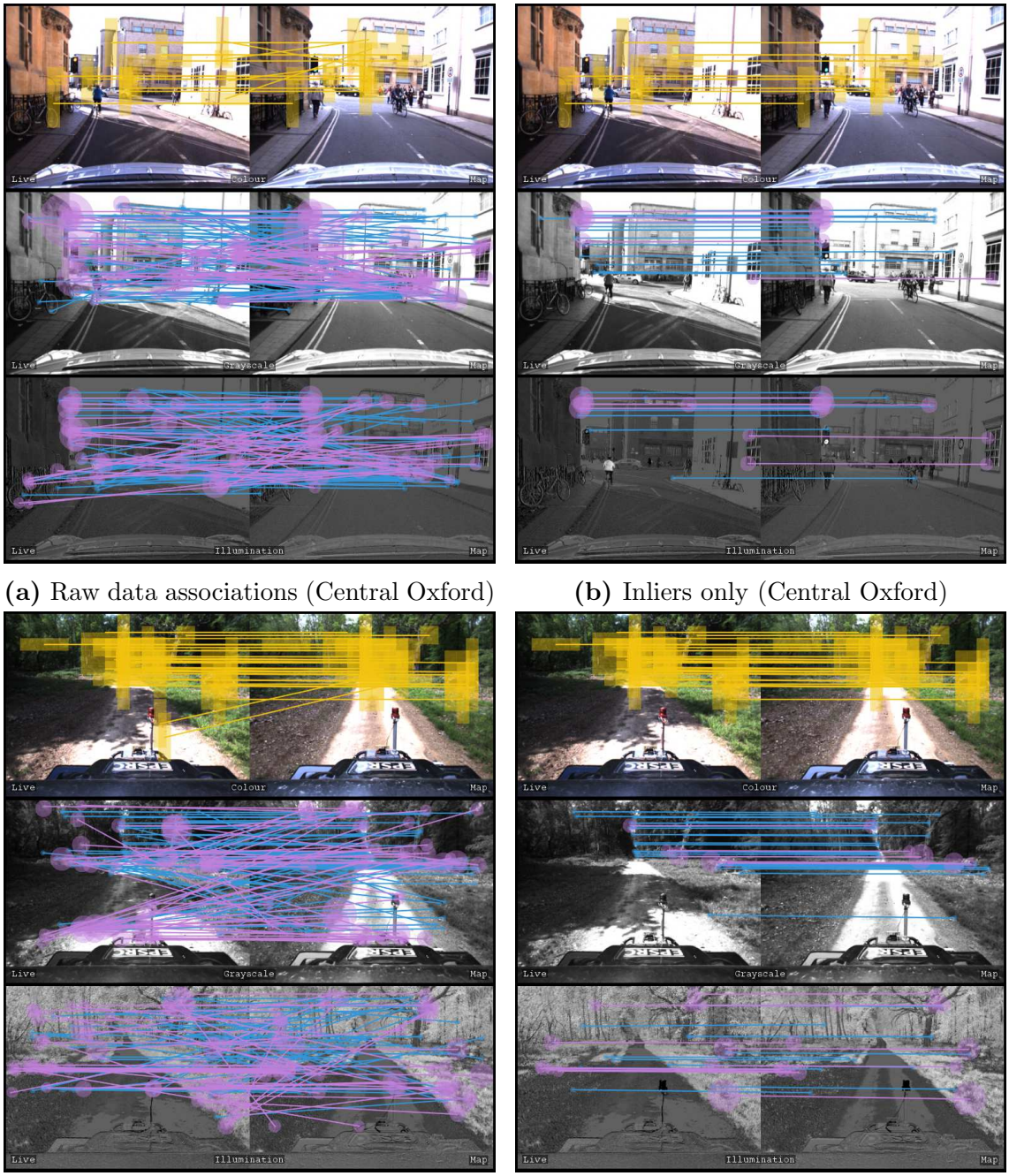


(b) Milton Keynes



(c) Cornbury Off-Road

Figure 7.9: The probability of localisation failure over a particular distance on three different datasets. The best results are observed on the Milton Keynes and Cornbury Off-Road datasets, however this is likely due to the significantly larger size of the Central Oxford dataset. This highlights the importance of testing over large scales to discover rare failure events.



(a) Raw data associations (Central Oxford)

(b) Inliers only (Central Oxford)

(c) Raw data associations (Cornbury)

(d) Inliers only (Cornbury)

Figure 7.10: Localisation performed on the Central Oxford and Cornbury Off-Road datasets. Data associations are shown for the patch detectors (yellow), SURF features (pink), and BRIEF features (blue). SURF and BRIEF operate on grayscale and illumination invariant images in parallel. The raw data associations are shown on the left, while the set of recovered inliers are shown on the right. Note that it would be difficult to extract the small set of inliers from the point correspondences alone, due to the high proportion of outliers. However, since patch detectors contain a high proportion of inliers, this facilitates robust outlier rejection on the point correspondences by providing a coarse estimate of pose, which is refined by the remaining inlier point correspondences.

A notable result of this experiment is the high degree of performance in the Cornbury Off-Road dataset. The only landmarks for localisation in this dataset are trees and bushes, rather than the static man made objects found in urban environments such as buildings and roads. Figure 7.9 shows that there is no loss in performance in these challenging off-road environments. Additionally, only two experiences were used in the off-road experiments, broadly corresponding to one in winter and one in summer. These two experiences captured the seasonal change in the environment, meaning that the localiser was largely invariant to illumination and weather changes.

Earlier in this chapter, we described Dub4 as a localisation system which employed a suite of techniques to perform robust localisation. Figure 7.10a and Figure 7.10c illustrate the raw data associations made on images from the Central Oxford and Cornbury Off-road datasets, respectively. Note that the point feature techniques (BRIEF in blue, SURF in pink) contain many outliers. Outlier rejection is difficult when the proportion of outliers is large. However, we observe that the patch detectors (yellow) contain very few outliers. While we could perform localisation purely using patch detectors (as in Chapter 6), the localisation estimate would be less accurate than that of point features. Rather, we use the patch detectors and point features in a single pose optimisation. The patch detectors provide a coarse estimate of the robot's pose, which at the same time assists with determining the set of inlier point feature correspondences. The small number of point feature correspondences in turn improves the accuracy of the pose estimate. Figure 7.10b and Figure 7.10d show the inlier data associations which are recovered and used to estimate the robot's pose.

This experiment has shown how the localiser is robust to a wide range of environmental change in the environment, and shows how the techniques described in this thesis are able to scale over large distances. The following section tests the

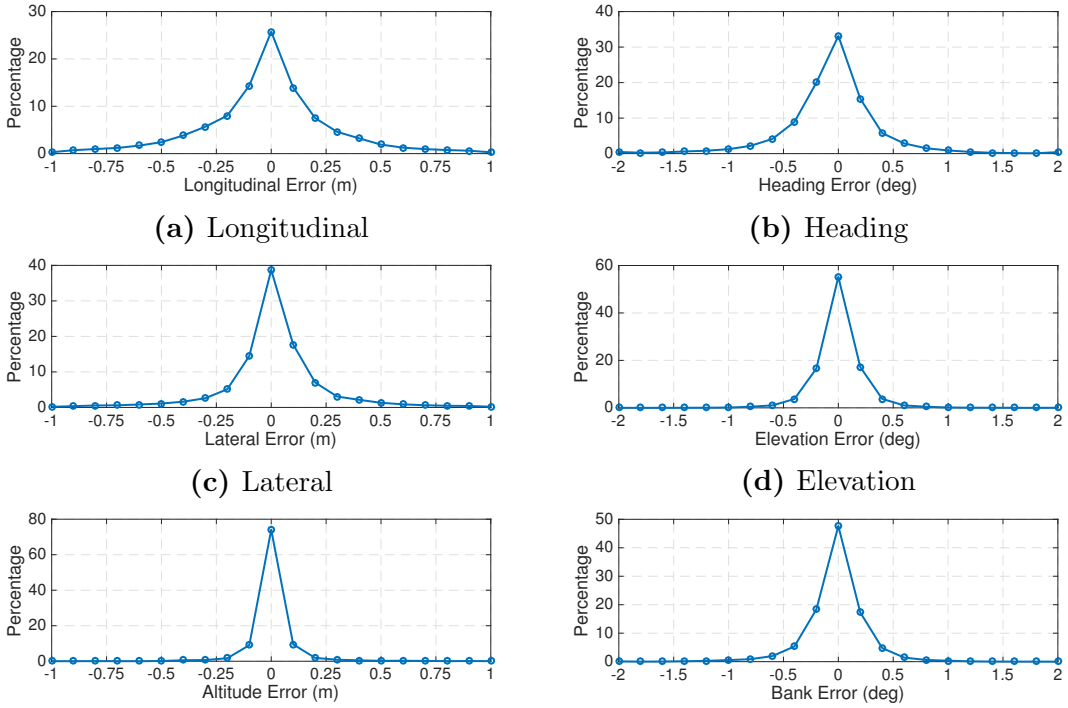
accuracy of the localiser against a ground truth estimate of the robot's pose.

7.4.3 Ground Truth

In this experiment, we compare the pose output from the localisation system against the known trajectory taken by the robot. However, determining this ground truth trajectory over large scales is a challenging problem in itself. As discussed in Chapter 3, even high-end GPS systems often fail near tall buildings and under tree cover, where the signal path between GPS satellites and the vehicle receiver is obstructed.

Rather, we manually align 10 logs of the Central Oxford route by performing a global pose graph relaxation. The optimisation uses relative constraints from visual odometry, and GPS is used to seed loop closures between logs. This is a time-consuming process and requires manual verification of loop closures, but ensures the best alignment given the available data. Maddern et al. (2015) use a similar technique to obtain ground truth estimates for the evaluation of their localisation system. They show that the obtained estimates consistently outperform the accuracy of a high-end GPS system. We acknowledge the limitations of this experiment, in that we are not able to fully characterise the accuracy of the estimate referred to as "ground truth". Obtaining accurate ground truth estimates over vast scales would be a valuable avenue for future research.

We perform the localisation experiment as follows. A map is generated using 15 experiences from the Central Oxford dataset (one of the maps used in the previous experiment). One of the experiences in this map is in the set of 10 logs which are globally optimised for ground truth. The remaining 9 logs are used for localisation. For each pose estimate obtained during the localisation runs, a graph search is performed between the localised node in the graph, and a node contained in the ground truth log. This allows us to report pose with respect to the single globally aligned log, and mimics how the localiser might be used in a closed-loop teach and



(a) Longitudinal **(b)** Heading

(c) Lateral **(d)** Elevation

(e) Altitude **(f)** Bank

Figure 7.11: Distribution of pose error with respect to the ground truth trajectories, where histograms are plotted with bin widths of 0.1 m for translation and 0.2 deg for rotation. The longitudinal, lateral and heading RMS errors are 0.27 m, 0.19 m, and 0.33 deg, respectively.

repeat experiment.

The results of this experiment are presented in Figure 7.11, where localisation errors are reported with respect to the globally optimised log. The longitudinal, lateral, and heading RMS errors are 0.27 m, 0.19 m, and 0.33 deg, respectively. These errors are inclusive of the following:

- Temporary localisation failures where the robot proceeds in open loop using visual odometry.
- Errors caused from incorrect data associations.
- Loss of accuracy using mid-level size patch detectors.
- Drift when performing a graph search through the experience graph to reach the single experience against which pose is output.

This performance is significantly greater than the high-end GPS+INS systems evaluated by Maddern et al. (2015). While the GPS+INS system was able to report consistent heading estimates, the translation estimates varied in the order of meters from ground truth. In their evaluation of laser localisation, Maddern et al. (2015) reported longitudinal, lateral and heading errors of 0.38 m, 0.07 m, and 0.43 deg, respectively. While we have not performed a direct comparison of these two localisers, we note that both systems perform to a comparable degree of accuracy. Future work could investigate this further.

7.4.4 Multiple Experiences

The use of multiple experiences is a core feature of this localisation system. Chapter 4 described how localisation performance improves as more experiences are added to the map, since the map is better able to model a complex and changing environment.

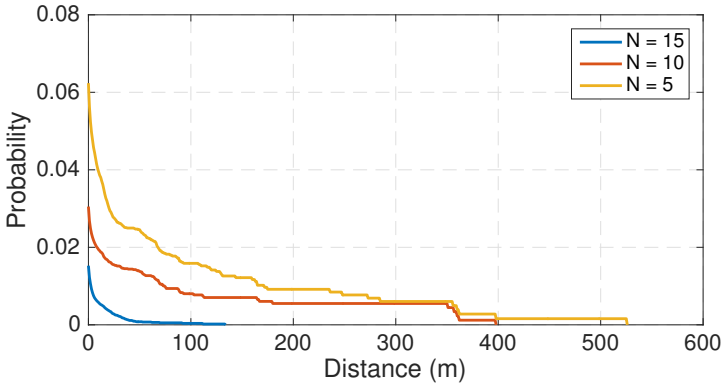


Figure 7.12: Graph showing that as more experiences are added to the map, the probability of localisation failure decreases dramatically. Here, N is the number of experiences from the Oxford 10 km route used in mapping.

In our main experiment, 15 experiences were used for mapping in the 1000 km Central Oxford dataset – significantly more than in the Milton Keynes or Cornbury Off-Road datasets. The Central Oxford experiment requires a higher number of experiences because of the large-scale construction works and accompanying road closures and detours that took place over the 18 months of data collection. This means that some areas of the map would have had significantly fewer experiences than others, in particular areas where a new route had to be learned. This problem would have affected any localisation system which relies on local sensing of the environment, and we see this ability to re-map areas automatically as an essential component of a life-long navigation system.

This experiment repeats the main experiment performed in Section 7.4.2 using the Oxford 10 km dataset. However, the number of experiences used to create the map is varied, and the effect on localisation performance observed. Figure 7.12 presents the results of this experiment. The graph shows that as more experiences are added to the map, the probability of localisation failure reduces.

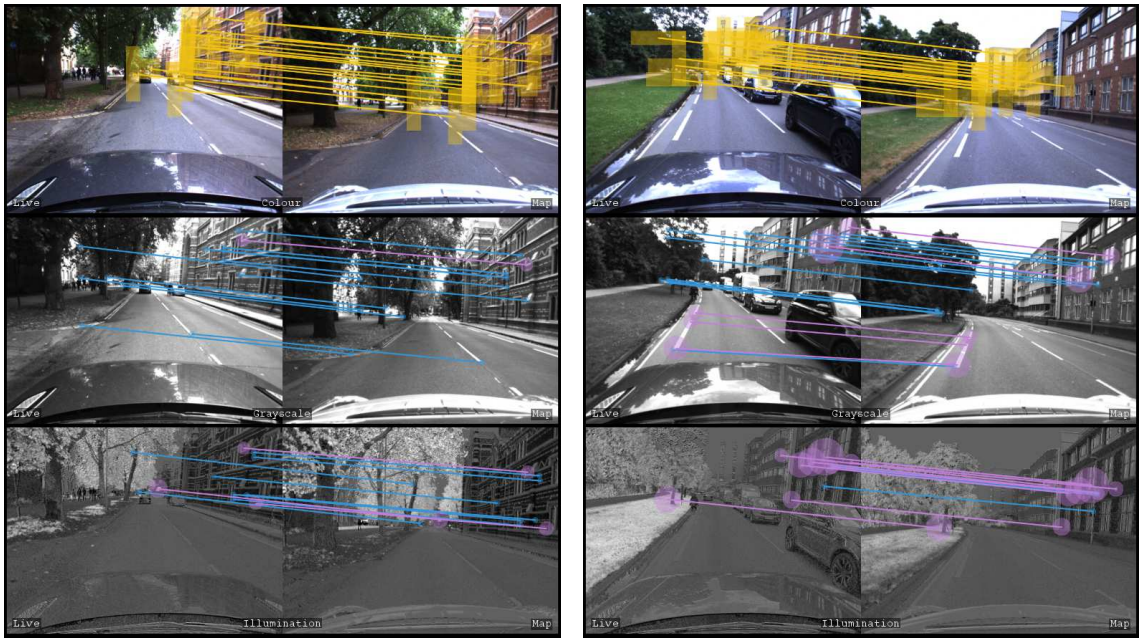
7.4.5 Lateral Translation

In these experiments, we have assumed that localisation failure is caused predominantly by changes in weather, lighting, season, and scene structure. However, an additional cause of localisation failure is lateral deviation away from the mapping trajectory. The datasets used in this evaluation are collected under manual driving, by a number of different drivers. While the drivers are briefed on the route to follow, they are not constrained on where to position the vehicle in the lane, or which lane to drive in if more than one is available.

This makes the localisation problem significantly more difficult. The localisation techniques described in this thesis are viewpoint dependent, so lateral deviations change the appearance of landmarks. Using point feature techniques under ideal conditions, we would expect localisation to succeed for roughly 1 m either side of a taught trajectory, however this tolerance decreases sharply as the degree of appearance change increases.

We could make the localisation problem easier by requiring that the vehicle is driven precisely over the same trajectory every time, however we have deliberately chosen not to do this. Where possible, we want to avoid assumptions about how the vehicle is driven, since in a city environment the vehicle may need to move around obstacles or position itself in the lane based on external factors other than how the route was originally driven.

We are robust to the problem of lateral deviation in two ways. Firstly, the experience framework naturally handles the problem of lateral deviation, since localisation failures caused by viewpoint change simply triggers the addition of a new experience. This has an impact on how many experiences are required to map large areas, but we have not found this to be a constraint in our system. Secondly, we are able to train translation invariance into the higher-level visual features, which we find increases the localiser's robustness to viewpoint change.



(a) **(b)**

Figure 7.13: Successful localisation performed using a Bumblebee XB3 mounted on a JLR Range Rover, against a map created with a camera mounted to a Nissan LEAF. The camera mounted on the Range Rover is pitched approximately 7 degrees down with respect to the camera mounted on the LEAF, causing the diagonal data associations observed here.

7.4.6 Localisation Between Different Vehicles

In this work, we have focused primarily on the case that a single vehicle performs unsupervised mapping and localisation as it traverses the environment. If there is change in the environment, the vehicle has to physically observe the change with its own camera before the map will be updated. However, within the context of autonomous cars and other applications, there are potentially many autonomous vehicles traversing the same network of roads at any given instant. The ability to share data and maps between robots would be hugely beneficial, since environmental change observed by one robot could be relayed to others.

However, this requires that the localisation techniques be able to work across different types of camera, and in spite of variations in the camera's mounting configuration on a particular vehicle. While this is not the focus of our work, we include

here a simple experiment to test the viability of inter-camera localisation. As discussed previously, the Central Oxford dataset was collected using a modified Nissan LEAF, with a forward-facing Bumblebee XB3 mounted above the windshield. Using maps generated with the LEAF, we attempted to localise using a camera mounted to a JLR Range Rover. A different Bumblebee XB3 was mounted to the Range Rover, and was pitched down approximately 7 degrees more than the camera mounted on the Nissan LEAF, as well as being mounted approximately 40 cm higher due to the larger vehicle size of the Range Rover.

Figure 7.13 illustrates successful data associations between the map created using the Nissan LEAF, and the localisation attempt using the JLR Range Rover. The diagonal data associations illustrate the noticeable difference in mounting configuration. While we have not had time to collect sufficient data in order to perform a compelling study of the cross-vehicle localisation performance, initial trials certainly seem promising.

7.4.7 Limitations of a Single Camera System

The focus of this thesis is vision-only localisation using a single stereo camera. This section discusses and illustrates some of the limitations and failure modes of a single camera system.

A common failure mode of single camera systems is that of harsh lighting conditions. Figure 7.14a shows the effect of driving into bright sun, for example during the early morning or late afternoon. There is some structure still visible on either side of the road, so localisation may still succeed, but nevertheless it does make localisation more challenging. Similarly, Figure 7.14b shows the limitations of the camera's dynamic range, where it is not able to expose the whole scene correctly in a single frame. In this example, the road was exposed correctly while the buildings were over-exposed. A multi-camera rig, with cameras oriented in different directions,

(a) Bright sun **(b)** Limited dynamic range **(c)** Feature-sparsity

(d) Poorly lit streets **(e)** Rain on lens **(f)** Bright lights and rain

Figure 7.14: Examples of scenarios which make localisation difficult or impossible. Mitigation strategies are discussed in Section 7.4.7.

would offer redundancy in these scenarios.

The localisation techniques described here rely on associating features in the live image with landmarks nearby the robot. In open fields, shown in Figure 7.14c, the only distinctive features are found along the horizon. The robot can only use these distant landmarks to optimise for the orientation of the robot, but not its position. This is a limitation of the image resolution and baseline of the stereo camera. In this situation, GPS could be used as a redundant system since the lack of tall buildings and tree cover would enable the best-case scenario GPS performance.

Poorly lit environments are fundamentally challenging for passive systems, since a dark image contains very little information about the environment and in turn makes localisation difficult or impossible. A future solution could investigate actively lighting the local area around the car, possibly with infrared light. Alternatively, in the scenario that some street lights are faintly visible, the street lights could be characterised and used as landmarks for localisation (Nelson et al., 2015a).

However, we note that if there is sufficient ambient light from street lights or the car's headlights, the night time condition is simply represented as an additional experience in the map which can be used for localisation.

While the localisation system is capable of localising while it is raining, Figure 7.14e shows the resulting poor image quality if rain drops fall directly on the lens. The rain drops cause the image to blur, which causes point feature techniques to fail. An extreme example of this failure mode is shown in Figure 7.14f, where the effects of rain on the lens are compounded by bright lights during the late evening. The vehicles used to perform data collection partially cover the camera lenses, so rain drops landing directly on the lens are uncommon, but nevertheless this problem would be easiest solved with a hardware solution.

7.5 Summary

This chapter has presented a vision-only localisation and mapping system called Dub4. The system harnesses a suite of techniques to perform robust localisation, including the use of multiple experiences to model a changing world, traditional point feature techniques, an illumination invariant transformation, and place-dependent patch detectors. While some of these techniques were presented separately in earlier chapters, this chapter has presented a system which harnesses their different strengths to provide redundancy and robustness.

The localisation system was validated on over 1200 km of data from three challenging environments, including the Central Oxford, Milton Keynes and Cornbury Off-Road datasets. These datasets contained a huge variety of appearance change, and tested the system's ability to map and localise over large distances. The probability of localisation failure for longer than 20 m was observed to be 0.00366, 0.000992 and 0.0009 for the Central Oxford, Milton Keynes and Cornbury Off-Road datasets,

respectively. In an experiment which tested the accuracy of the pose estimates, the system exhibited longitudinal, lateral and heading RMS errors of 0.27 m, 0.19 m, and 0.33 deg, respectively. In addition to this extensive offline testing, the system has been demonstrated in closed-loop on a number of autonomous platforms, including the Oxford University RobotCar and the LUTZ PathFinder trials. This system represents a significant capability in outdoor vision-only localisation, providing robust and accurate pose estimates in spite of changes in weather, lighting, season and scene structure.

Chapter 8

Conclusion

8.1 Summary

This thesis has addressed the problem of vision-only localisation in outdoor environments using a single stereo camera. In particular, this work has focused on the development of a camera-based localisation system for autonomous vehicles operating over large distances in outdoor environments. The use of cameras is appealing due to their low cost and information-rich sensing modality. However, appearance change makes vision-only localisation a challenging problem. In outdoor environments, the appearance of the world can change drastically and unpredictably with variations in weather, lighting, season and scene structure. This thesis has introduced and described a number of new techniques in the field of vision-only localisation to address these challenges.

Chapter 3 presented five large datasets, totalling over 1500 km of data, which were collected and used in the development of the algorithms presented in this thesis. Most notably, this included 100 repeats of a 10 km route through Central Oxford over a period of 18 months, in all weather conditions, at all times of the day and night, and in all seasons. Additionally, significant structural change took place along

the route over the period of data collection – including where the road network was physically altered. We also collected 200 km of data in an off-road environment. This presents a different challenge for localisation, since only natural landmarks such as bushes and trees are available for localisation, as opposed to the man-made structures found in urban environments. These datasets have proved invaluable in the testing and evaluation of our localisation system.

In Chapter 4, we discussed our approach to robust localisation using the experience paradigm. The chapter described an experience as a visual memory, or snapshot of the world under particular conditions. Over time, the robot experiences the world under many different environmental conditions (e.g. sun, rain, winter, summer, morning, evening, etc.), and so experiences model the world under different appearances. A graph-based implementation of the experience framework was presented, where multiple, overlapping experiences are accumulated in the map over time to model the full spectrum of change in the environment. By improving the quality of the representation of the world, localisation is more robust at runtime. The experience paradigm is an essential component of the localisation system presented here.

Chapter 5 described the problem of experience density. While the experience paradigm enables significant performance gains in localisation, some areas may exhibit a wide range of appearance change and require a correspondingly high number of experiences to model the environment. We described how this can become a problem for resource-constrained machines, where the robot must do more work in order to ensure optimal localisation performance. To handle this, a probabilistic technique for intelligent memory management was presented. The method is unsupervised and learns to predict which experiences are most likely to localise the live image successfully. This technique was demonstrated on over 200 km of data from three different datasets, showing that the computational requirements were reduced

by half, while maintaining the same localisation performance.

Chapter 6 presented an alternative to traditional point feature techniques. Traditional point feature techniques such as SIFT, SURF and BRIEF, extract and describe low-level features in an image. These are used to perform data associations between a live image and an image in the map. However, these techniques are not robust to the levels of appearance change encountered in outdoor environments. Previous chapters approached this problem with the use of experiences, however some applications may not allow repeat visits to the environment under a range of appearance conditions to generate a robust map. To address this problem, we presented a new pose estimation technique which is significantly more robust than traditional point feature approaches. We asserted that larger visual elements in the environment are more distinctive, and less brittle to appearance change, than low-level point feature techniques. The chapter described an unsupervised technique which mines mid-level distinctive landmarks from the environment, modelling their appearance with linear SVM classifiers. The training method generates robust detectors from a single traverse through the environment, and does not require GPS or manual alignment of datasets. Results are presented on over 200 km of challenging data from the Central Oxford dataset, including in bright sun, rain, cloud, and night time conditions, and across seasonal change. We show that the probability of localisation failure is reduced by a factor of 6, when compared with traditional point feature techniques.

Chapter 7 represented the culmination of the work performed in this thesis, where we presented a complete vision-only localisation system called Dub4. The system draws together a number of different threads, including the experience-based paradigm of Chapters 4 and 5, the patch detectors of Chapter 6, traditional point feature techniques such as SURF and BRIEF, and an illumination invariant image transformation. These various techniques have complementary failure modes,

resulting in a combined system which is significantly more robust than any of the individual components alone. The performance of the localisation system is evaluated on over 1200 km of data from central Oxford, Milton Keynes, and an off-road environment in Cornbury Park, North Oxfordshire. We show that the probability of localisation failure over distances longer than 20 m was 0.00366, 0.000992 and 0.0009 for the Central Oxford, Milton Keynes and Cornbury Off-Road datasets, respectively. In terms of localisation accuracy, we show that the system exhibits longitudinal, lateral, and heading RMS errors of 0.27 m, 0.19 m, and 0.33 deg, respectively. In addition to this extensive offline testing, Dub4 currently serves as the primary localisation source on a number of autonomous vehicles, including a major public demonstration of autonomous vehicles in London as part of the GATEway Project.

8.2 Future Work

Lifelong navigation remains an active topic of research within the robotics community. While this thesis presents advancements in vision-only localisation, our approach to the problem brings with it new challenges and opportunities for research.

Characterise the relationship between robustness and accuracy

In Chapter 7, the localisation system was described as using point features and patch detectors in parallel for localisation. We discussed how point features were accurate in their measurements, but brittle to appearance change; while patch detectors were robust to appearance change, but coarse in their metric accuracy. Using both point features and patch detectors allowed the system to benefit from the metric accuracy of the point features when they were available, but also facilitated the use of robust, but less accurate, patches under extreme appearance change. The development of

uncertainty metrics which capture this trade off in accuracy would be valuable in assessing localisation performance at run-time.

Effect of additional experiences on pose accuracy

This thesis has explored how the robustness of the localisation system improves over time as more experiences are added to the map. However, we have not investigated how the accuracy of localisation estimates might change over time as the map is augmented. We would expect that the experience graph would become more connected as more experiences are added to the map, since the incremental changes between different appearances are better captured. This would reduce the length of search paths through the graph, resulting in less accumulated error in pose estimates.

Improving the accuracy of high-level visual features

In Chapter 6, a training algorithm was described which identified and described distinctive landmarks in the environment. An important feature of this training algorithm was that landmarks were modelled as 3D points in the world. However, there are many distinctive visual elements in the environment where this assumption does not hold, and the accuracy of the pose estimate degrades. Future work could investigate how more accurate pose estimates could be obtained from these patch correspondences.

Multiple vehicles

In this thesis, we have considered a single vehicle surveying the environment and learning from its past experience. While we have presented some promising initial results of localisation between different vehicles in Chapter 7, we have not investigated this in depth. We have not considered how this might scale to thousands of vehicles operating at the same time, and how experience might be shared between robots. Data management would be an important issue given the huge volumes of

data being captured, processed and shared.

Deleting experiences

This thesis has considered appearance change which is mainly cyclic, for example changes in lighting, weather and season. While we have encountered many examples of structural change in the environment, we have not attempted to permanently delete old experiences which represent the world's previous state. A solution to this problem would need to distinguish between experiences which are only temporarily not relevant (e.g. changes in the time of day), and those which will never be used again (e.g. structural change). An unsupervised solution to this problem is important for life-long navigation.

Sensor fusion

Chapter 7 showed how the use of multiple localisation techniques offered redundancy within the vision-only localiser. This could be extended to fusing estimates from multiple cameras, and additionally from multiple sensor modalities. For example, GPS estimates could be fused with the camera-based localisation estimates, and wheel odometry could be used as an additional source of relative motion, together with visual odometry. While none of these sensors offer perfect information, a sensor fusion approach would provide redundancy and overall increased robustness.

8.3 Concluding Remark

This thesis has presented Dub4, a vision-only localisation system for autonomous vehicles. A key message in this thesis is that no single algorithm or method provides perfect robustness, in all places, or under all environmental conditions. Rather, we gain robustness and redundancy by harnessing a suite of complementary techniques with *different* failure modes. In concert, these algorithms produce a step change

in our ability to perform life-long navigation under challenging and unpredictable conditions. It is hoped that the algorithms and insights shared here will contribute towards the development of truly autonomous vehicles.

Appendix A

Acronyms

EBN Experience-Based Navigation

VO Visual Odometry

INS Inertial Navigation System

SIFT Scale-Invariant Feature Transform

SURF Speeded-Up Robust Features

FAST Features from Accelerated Segment Test

BRIEF Binary Robust Independent Elementary Features

HOG Histogram of Orientated Gradients

ACF Aggregated Channel Features

RANSAC Random Sample and Consensus

P3P Perspective From Three Points

SVM Support Vector Machine

Bibliography

Atanasov, N., Zhu, M., Daniilidis, K., and Pappas, G. J. (2016). Localization from semantic observations via the matrix permanent. *The International Journal of Robotics Research*, 35(1-3):73–99.

Bailey, T. and Durrant-Whyte, H. (2006). Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117.

Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. (2008). SURF: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110:346–359.

Bosse, M., Newman, P. M., Leonard, J. J., and Teller, S. (2004). SLAM in Large-scale Cyclic Environments using the Atlas Framework. *The International Journal of Robotics Research*, 23(12):1113–1139.

Brooks, R. (1985). Visual map making for a mobile robot. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 824–829. IEEE.

Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). BRIEF: Binary robust independent elementary features. *Computer Vision–ECCV 2010*, pages 778–792.

Canny, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679–698.

Churchill, W. (2012). *Experience-Based Navigation: Theory, Practice and Implementation*. PhD thesis, University of Oxford, Oxford, United Kingdom.

Churchill, W. and Newman, P. (2013). Experience-Based Navigation for long-term localisation. *The International Journal of Robotics Research*, 32(14):1645–1661.

Churchill, W., Tong, C. H., Gurău, C., Posner, I., and Newman, P. (2015). Know your limits: Embedding localiser performance models in teach and repeat maps. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 4238–4244. IEEE.

Cummins, M. and Newman, P. (2008). FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 27(6):647–665.

Cummins, M. J. and Newman, P. M. (2010). FAB-MAP: Appearance-based place recognition and mapping using a learned visual vocabulary model. In *Proc. of the 27th International Conference on Machine Learning (ICML-10)*, pages 3–10.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 886–893. IEEE.

Dissanayake, M. G., Newman, P., Clark, S., Durrant-Whyte, H. F., and Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on robotics and automation*, 17(3):229–241.

Doersch, C., Singh, S., Gupta, A., Sivic, J., and Efros, A. A. (2012). What makes Paris look like Paris? *ACM Transactions on Graphics (SIGGRAPH)*, 31(4):101:1–101:9.

Dollár, P., Appel, R., Belongie, S., and Perona, P. (2014). Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1532–1545.

Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: Part I. *IEEE robotics & automation magazine*, 13(2):99–110.

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

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

Foster, D. H. (2011). Color constancy. *Vision Research*, 51(7):674–700.

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

Grassia, F. S. (1998). Practical parameterization of rotations using the exponential map. *Journal of Graphics Tools*, 3(3):29–48.

Hamilton, W. R. (1844). LXXVIII. On quaternions; or on a new system of imaginaries in algebra: To the editors of the Philosophical Magazine and Journal.

Harris, C. and Stephens, M. (1988). A combined corner and edge detection. In *Proceedings of The Fourth Alvey Vision Conference*, pages 147–151.

Hartley, R. and Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge University Press.

Huber, P. J. (2011). *Robust statistics*. Springer.

Kendall, A., Grimes, M., and Cipolla, R. (2015). Posenet: A convolutional network for real-time 6-DOF camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946.

Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

Krusi, P. (2016). *Autonomous Navigation in Complex Nonplanar Environments Based on Laser Ranging*. PhD thesis, ETH Zurich.

Leutenegger, S., Chli, M., and Siegwart, R. (2011). BRISK: Binary robust invariant scalable keypoints. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 2548 –2555.

Levinson, J. and Thrun, S. (2010). Robust vehicle localization in urban environments using probabilistic maps. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4372–4378. IEEE.

Linegar, C., Churchill, W., and Newman, P. (2015). Work smart, not hard: Recalling relevant experiences for vast-scale but time-constrained localisation. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 90–97. IEEE.

Linegar, C., Churchill, W., and Newman, P. (2016). Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 787–794. IEEE.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

MacTavish, K., Paton, M., and Barfoot, T. D. (2016). Beyond a shadow of a doubt: Place recognition with colour-constant images. In *Field and Service Robotics*, pages 187–199. Springer.

Maddern, W., Pascoe, G., Linegar, C., and Newman, P. (2017). 1 year, 1000 km: The Oxford Robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15.

Maddern, W., Pascoe, G., and Newman, P. (2015). Leveraging experience for large-scale lidar localisation in changing cities. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1684–1691. IEEE.

Maddern, W., Stewart, A., McManus, C., Upcroft, B., Churchill, W., and Newman, P. (2014a). Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles. In *Proc. IEEE Int. Conference on Robotics and Automation (ICRA), Hong Kong, China*.

Maddern, W., Stewart, A. D., and Newman, P. (2014b). LAPS-II: 6-DoF day and night visual localisation with prior 3D structure for autonomous road vehicles. In *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*, pages 330–337. IEEE.

Matas, J., Chum, O., Urban, M., and Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of the British Machine Vision Conference*, pages 36.1–36.10. BMVA Press.

Matthies, L. (1989). *Dynamic stereo vision*. PhD thesis, Carnegie Mellon University.

McManus, C., Churchill, W., Maddern, W., Stewart, A. D., and Newman, P. (2014). Shady dealings: Robust, long-term visual localisation using illumination invari-

ance. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 901–906. IEEE.

McManus, C., Furgale, P., Stenning, B., and Barfoot, T. D. (2012). Visual teach and repeat using appearance-based lidar. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 389–396. IEEE.

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

Milford, M. and Wyeth, G. (2012). SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *IEEE International Conferece on Robotics and Automation (ICRA 2012)*, pages 1643–1649. IEEE.

Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B., et al. (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Aaai/iaai*, pages 593–598.

Mühlfellner, P., Bürki, M., Bosse, M., Derendarz, W., Philippsen, R., and Furgale, P. (2015). Summary maps for lifelong visual localization. *Journal of Field Robotics*.

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT press.

Murray, R. M., Li, Z., Sastry, S. S., and Sastry, S. S. (1994). *A mathematical introduction to robotic manipulation*. CRC press.

Nelson, P., Churchill, W., Posner, I., and Newman, P. (2015a). From dusk till dawn: Localisation at night using artificial light sources. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 5245–5252. IEEE.

Nelson, P., Linegar, C., and Newman, P. (2015b). Building, curating, and querying large-scale data repositories for field robotics applications. In *International Conference on Field and Service Robotics (FSR)*, Toronto, ON, Canada.

Nistér, D., Naroditsky, O., and Bergen, J. (2004). Visual odometry. In *Computer Vision and Pattern Recognition, 2004*, volume 1, pages I–652. IEEE.

Nistér, D., Naroditsky, O., and Bergen, J. (2006). Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1):3–20.

Ortiz, R. (2012). FREAK: Fast retina keypoint. In *of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 510–517. IEEE Computer Society.

Paton, M., MacTavish, K., Ostafew, C. J., and Barfoot, T. D. (2015a). It's not easy seeing green: Lighting-resistant stereo visual teach & repeat using color-constant images. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1519–1526. IEEE.

Paton, M., MacTavish, K., Warren, M., and Barfoot, T. D. (2016a). Bridging the appearance gap: Multi-experience localization for long-term visual teach and repeat. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 1918–1925. IEEE.

Paton, M., Pomerleau, F., and Barfoot, T. D. (2015b). Eyes in the back of your head: Robust visual teach & repeat using multiple stereo cameras. In *Computer and Robot Vision (CRV), 2015 12th Conference on*, pages 46–53. IEEE.

Paton, M., Pomerleau, F., and Barfoot, T. D. (2016b). In the dead of winter: Challenging vision-based path following in extreme conditions. In *Field and Service Robotics*, pages 563–576. Springer.

Ratnasingam, S. and Collins, S. (2010). Study of the photodetector characteristics of a camera for color constancy in natural scenes. *JOSA A*, 27(2):286–294.

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

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

Salas-Moreno, R. F., Newcombe, R. A., Strasdat, H., Kelly, P. H., and Davison, A. J. (2013). Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1352–1359.

Scaramuzza, D. and Fraundorfer, F. (2011). Visual odometry: Part I - the first 30 years and fundamentals. *IEEE Robotics Automation Magazine*.

Sibley, G., Mei, C., Reid, I., and Newman, P. (2010a). Planes, trains and automobiles – autonomy for the modern robot. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 285–292. IEEE.

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

Sivic, J., Zisserman, A., et al. (2003). Video google: A text retrieval approach to object matching in videos. In *ICCV*, volume 2, pages 1470–1477.

Stuelpnagel, J. (1964). On the parametrization of the three-dimensional rotation group. *SIAM review*, 6(4):422–430.

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

Valgren, C. and Lilienthal, A. J. (2010). SIFT, SURF and seasons: Appearance-based long-term localization in outdoor environments. *Robotics and Autonomous Systems*, 58(2):149 – 156.

Vapnik, V. (1995). *The nature of statistical learning theory*. Springer science & business media.

von Hundelshausen, F. and Sukthankar, R. (2012). D-Nets: Beyond patch-based image descriptors. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2941–2948. IEEE.

Warren, M., McKinnon, D., and Upcroft, B. (2013). Online calibration of stereo rigs for long-term autonomy. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3692–3698. IEEE.