Learning Place-Dependant Features For Long-Term Vision-Based Localisation

Colin McManus

Lady Margaret Hall

Supervisor:

Professor Paul Newman

Mobile Robotics Group

Department of Engineering Science

University of Oxford

October 2014

Colin McManus
Lady Margaret Hall

Doctor of Philosophy
October 2014

Learning Place-Dependant Features For Long-Term Vision-Based Localisation

Abstract

In order for autonomous vehicles to achieve life-long operation in outdoor environments, navigation systems must be able to cope with visual change—whether it's short term, such as variable lighting or weather conditions, or long term, such as different seasons. As a Global Positioning System (GPS) is not always reliable, autonomous vehicles must be self sufficient with onboard sensors. This thesis examines the problem of localisation against a known map across extreme lighting and weather conditions using only a stereo camera as the primary sensor. The method presented departs from traditional techniques that blindly apply out-of-the-box interest-point detectors to all images of all places. This naive approach fails to take into account any prior knowledge that exists about the environment in which the robot is operating. Furthermore, the point-feature approach often fails when there are dramatic appearance changes, as associating low-level features such as corners or edges is extremely difficult and sometimes not possible. By leveraging knowledge of prior appearance, this thesis presents an unsupervised method for learning a set of distinctive and stable (i.e., stable under appearance changes) feature detectors that are unique to a specific place in the environment. In other words, we learn place-dependent feature detectors that enable vastly superior performance in terms of robustness in exchange for a reduced, but tolerable metric precision. By folding in a method for masking distracting objects in dynamic environments and examining a simple model for external illuminates, such as the sun, this thesis presents a robust localisation system that is able to achieve metric estimates from night-today or summer-to-winter conditions. Results are presented from various locations in the UK, including the Begbroke Science Park, Woodstock, Oxford, and central 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.

Colin McManus, Lady Margaret Hall

Funding

The work described in this thesis was funded by Nissan Motors.

Dedication

To my brother. My best friend.

Acknowledgements

None of this would have been possible without the continued support, mentoring, friendship, and motivation from my supervisor, Professor Paul Newman. I am in awe of everything Professor Newman has accomplished and continues to achieve. Words cannot express how deeply grateful and honoured I am to have been a part of the MRG family at Oxford. It has been nothing short of spectacular.

I wish to thank Nissan Motors for their generous funding of my DPhil. I would not be here without them.

I extend my deepest gratitude towards Dr. Winston Churchill and Alex Stewart for all of their help over the years. They played a pivotal role in my degree and I will always be grateful.

I also wish to acknowledge the years of guidance and support from my former supervisor, Professor Tim Barfoot, and former colleague, Dr. Paul Furgale, both of whom have been an inspiration to me.

I want to thank Hugo Grimmett, Dr. Pedro Pinies, and Dr. Chi Hay Tong for proof reading and putting up with me during the final hours of the writeup. All of you were fantastic.

Lastly, and most importantly, I wish to acknowledge my family and friends. They have always supported me in every endeavour and without their strength and encouragement, I would not be the man I am today. I am truly grateful for everyone that has been in my life.

Thank you all for everything.

Colin McManus

December 5, 2014

Notation

Symbol	Description
a	A real-valued scalar: $a \in \mathbb{R}$
$\mathbf{a}$	A real-valued $N \times 1$ column vector: $\mathbf{a} \in \mathbb{R}^{N \times 1}$
$\mathbf{A}$	A real-valued $N \times M$ matrix: $\mathbf{A} \in \mathbb{R}^{N \times M}$
$\underrightarrow{\mathcal{F}}$	A reference frame defined by three unit vectors
$\mathbf{R}_{a,b}$	The 3×3 SO(3) rotation matrix that rotates points from $\underrightarrow{\mathcal{F}}_b$ to $\underrightarrow{\mathcal{F}}_a$: $\mathbf{p}_a = \mathbf{R}_{a,b}\mathbf{p}_b$
$\mathbf{t}_a^{b,a}$	A vector pointing from $\underrightarrow{\mathcal{F}}_a$ to $\underrightarrow{\mathcal{F}}_b$ and expressed in $\underrightarrow{\mathcal{F}}_a$
$\boldsymbol{t}_a^{b,a}$	The homogenous form of $\mathbf{t}_a^{b,a}$: $\boldsymbol{t}_a^{b,a} = \begin{bmatrix} \mathbf{t}_a^{b,a} \\ 1 \end{bmatrix}$
$\mathbf{T}_{a,b}$	The 4×4 SE(3) transformation matrix that transforms points from $\underrightarrow{\mathcal{F}}_b$ to $\underrightarrow{\mathcal{F}}_a$: $\boldsymbol{p}_a = \mathbf{T}_{a,b}\boldsymbol{p}_b$
$\mathbf{1}$	The identity matrix
$\mathbf{0}$	The zero matrix
$\sim \mathcal{N}(\mathbf{x}, \mathbf{P})$	Normally distributed with mean $\mathbf{x}$ and covariance $\mathbf{P}$
$(\cdot)^\wedge$	Matrix-cross operator
$(\cdot)^\vee$	Inverse matrix-cross operator

Contents

Chapter 1

Introduction

1.1 Motivation

For robots to autonomously navigate outdoors over vast scales and long periods of time, they must be able to answer the question, "where am I?", regardless of time of day, time of year, or the weather. Put simply, they must be able to cope with visual change, both sudden and gradual.

Although it may be tempting to assume that a Global Positioning System (GPS) would be a good solution for the problem of localisation (i.e., knowing where you are), GPS can be very unreliable in terms of availability and accuracy due to signal blockage and/or satellite drift. Thus, vehicles must use onboard sensing and be completely self sufficient.

This thesis explores the task of long-term, persistent localisation and asks what is possible using a single stereo camera along with a map of the environment. Cameras are an appealing option for onboard sensing as they are low-cost, commercial off-the-shelf devices that provide a rich source of visual information.

(a) Oxford: night, pitch black. (b) Oxford: afternoon, sunny. (c) Oxford: evening, sunny (d) Oxford: afternoon, overcast.

(e) Begbroke: night, dark and rainy. (f) Begbroke: afternoon, sunny. (g) Begbroke: afternoon, snowy. (h) Begbroke: evening, dark.

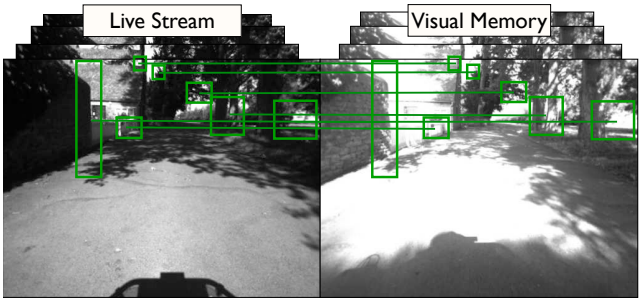
Figure 1.1: Images of the same place but taken at different seasons and/or times of day. Top four: Oxford. Bottom four: Begbroke Science Park.

1.2 Coping with Change

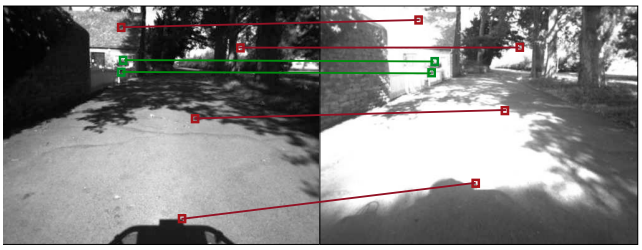
Why is dealing with visual change difficult? For a person, it may be easy to recognise that the images in each row of Figure 1.1 are all of the same place, but under different environmental conditions. However, for the current state of the art in vision-based localisation, this remains a considerable challenge, owing primarily to the reliance on point features for data association.

For decades, the typical approach in robotics has been to search for point correspondences across images, which are typically represented by low-level structure such as edges or corners. Even at different scales, these types of feature detectors are limited to searching for basic primitives and not mid-level or higher-level content in the scene, such as distinctive landmarks. Thus, when faced with significant appearance changes as shown in Figure 1.1, these techniques typically fail to find associations.

There is also something unsettling with the idea of blindly applying the same fixed detection procedure across all images from all places, especially if prior knowl-



(a) By matching *scene signatures* from a live stream (left) to a memory (right), we are able to *successfully* localise our vehicle.



(b) By matching point features from a live stream (left) to a memory (right), we are unable to successfully localise our vehicle.

Figure 1.2: Illustration of feature matching using scene signatures, which are distinctive visual elements such as fences, windows, tree lines, etc., versus the traditional point-feature approach. Using point features for data association under extreme appearance changes often fails because point features only consider low-level structure, like edges or corners. Scene signatures are more robust since they are large, distinctive elements.

edge of the environment exists. In this case, it makes sense to leverage knowledge of prior appearance and structure to learn what is important in a scene and what to look for.

The primary contribution of this thesis is the idea of learning place-dependent feature detectors for persistent outdoor localisation. These place-dependent feature detectors are engineered to detect mid-level patches representing distinctive visual elements, such as windows, tree silhouettes, or signs. This enables rough metric position and orientation (pose) estimation across extreme lighting and weather conditions; this is not possible with the point-feature counter part (see Figure 1.2).

It should be made clear that this thesis is not taking the stance that point features are bad. They have their place in a number of systems and applications.

For relative-motion estimation, like Visual Odometry (VO), point features work extremely well as viewpoint and lighting conditions typically do not change much from frame to frame[1]. Or if one is localising in an environment without much visual change, then point features could be a good solution. However, they do not seem to be well suited for the task of localising across extreme appearance differences where point correspondences are either not possible or very sparse.

Although it is the core contribution of the thesis, it should be noted that the topic of place-dependent feature detectors will be covered in a later chapter. The thesis begins with earlier work addressing other agents of change, which follows the natural progression of the research—it was the limitations of our standard feature-based system that led to the idea of the new approach. The various sources of visual change that will be addressed in this work are:

1. Dynamic objects (i.e., visual distractions)
2. Illumination changes
3. Weather and seasonal changes

Each topic will be addressed in the chapters to come.

Although there a numerous applications for this work, the thesis will focus in particular on self-driving vehicles operating in urban environments. From early pioneering work in the late 70s to 80s (Tsugawa et al., 1979; Dickmanns and Zapp, 1987; Pomerleau, 1989) to a series of international competitions from 2004-2007 organised by DARPA (Thrun et al., 2005b; Urmson et al., 2008), autonomous vehicle technologies have advanced to the point where public availability will be a near certainty within the coming decades. Onboard navigation systems must be able to satisfy the requirements stated earlier—long-term, persistent localisation in ever-changing environments.

[1]In fact, VO is one of the core vision tools used throughout this thesis for egomotion estimation.

1.3 Contributions

There have been a number of research contributions leading up to this thesis and are described below. The common thread is long-term, persistent navigation, dealing with various agents of change.

The first contribution is the development of a technique that can suppress dynamic objects in a scene for improved egomotion estimation:

> McManus, C., Churchill, W., Napier, A., Davis, B., and Newman, P. (2013). Distraction suppression for vision-based pose estimation at city scales. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany

Following this, we integrated a novel illumination-invariant image transform into a visual-perception pipeline and a vision-based localisation pipeline to address the issues of localisation and scene understanding in shadowy environments:

> McManus, C., Churchill, W., Maddern, W., Stewart, A., and Newman, P. (2014a). Shady dealings: Robust, long-term visual localisation using illumination invariance. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China

> 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 *Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation*, Hong Kong, China

Upcroft, B., McManus, C., Churchill, W., Maddern, W., and Newman, P. (2014). Lighting invariant urban street classification. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China

The last area of work concerns the core contribution, which is the concept of learning place-dependent feature detectors to enable robust visual localisation across extreme appearance changes:

McManus, C., Upcroft, B., and Newman, P. (2014b). Scene signatures: Localised and point-less features for localisation. In *Proceedings of Robotics Science and Systems (RSS)*, Berkley, CA, USA

McManus, C., Upcroft, B., and Newman, P. (2015). Learning place-dependent feature detectors for long-term, outdoor navigation. *In Preperation for Submission to Autonomous Robots, special issue on selected papers from RSS 2014.*

Although not related to research, there was also some noteworthy systems-related work for Oxford's Autonomous Car Project[2], which included contributing to a core pose-management library and a visual teach-and-repeat system that was successfully used for in-the-loop control.

[2]See `http://mrg.robots.ox.ac.uk/robotcar/` for more details and videos.

1.4 Thesis Roadmap

The story of this thesis is life-long, vision-based localisation for autonomous road vehicles. The common thread in this research is leveraging knowledge of prior appearance and structure. It is therefore assumed that a 3D point cloud along with a stereo-image sequence is available during runtime.

As discussed earlier, various sources of visual change will be covered. The first is concerned with operating in dynamic environments with large obstacles that act as visual distractions. The next topic examines operating in illumination-varying environments with shadows, which prove to be remarkably problematic for point-feature systems. Still not addressing the more long-term issues, such as seasonal changes, we depart from the traditional feature-based approach and present a method for learning place-dependent feature detectors to enable localisation across extreme lighting, weather, and seasonal conditions.

The following is a more detailed description of each chapter and its relevance to the overarching theme of the thesis (Figure 1.3 presents the thesis roadmap graphically).

Chapter 2: Introduces the state estimation machinery used throughout this thesis as well as an overview of our Visual Odometry (VO) and localisation pipeline.

Chapter 3: Presents distraction suppression for robust egomotion estimation in highly dynamic environments. Accurate relative motion estimation is important for a localisation system, since dead reckoning estimates are used for position tracking in between localisation updates.

Chapter 4: Presents an illumination-invariant colour space and analyses its impact on a visual localisation pipeline. This colour space is used in a later

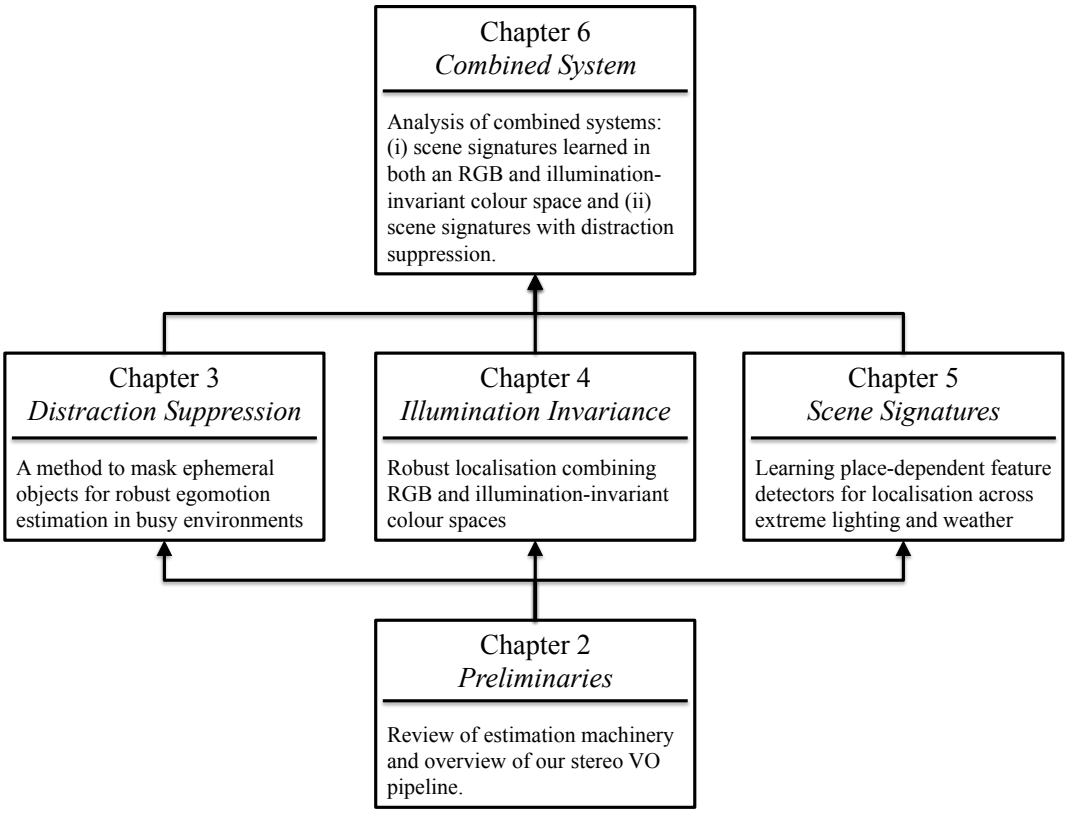


Figure 1.3: A diagram providing a summary of the chapters as well as their inter-dependencies.

chapter to asses performance gains achievable with a simple model of an external illuminant.

Chapter 5: Presents the core idea of learning place-dependent feature detectors, called *scene signatures*, for localisation across extreme appearance changes.

Chapter 6: Analyses possible performance gains achievable by combining illumination invariance and distraction suppression within the scene-signature pipeline.

The last chapter concludes with a summary of the thesis, its contributions, and the future directions of the work.

Chapter 2

Preliminaries

2.1 Estimation

This section provides a brief overview of the estimation theory and tools that underpin the methods described in this thesis. We begin with a sketch of the estimation problem from a Bayesian framework and then show how this can be cast as a non-linear, least-squares optimisation problem. Following this, we present some useful techniques and tools that will be used for linear error propagation and deriving the Jacobians of our sensor models.

We state the problem in its most general form, which involves estimating both the robot pose and map simultaneously. We then discuss the recent trends in the field that have been moving towards more application-specific approaches, which relax the requirement for concurrently estimating both the pose and map.

More formally, we state the problem as follows. At any given time, t_k, we wish to compute the joint posterior density of the pose of a robot, $\mathbf{x}_k$, as well as the map of the environment, $\mathbf{p}_k$, which we represent by 3D landmarks, using all past/present noisy sensor measurements, $\mathbf{z}_{0:k}$, which we represent by image coordinates, control inputs, $\mathbf{u}_{0:k}$, which we represent by motion estimates, and prior knowledge of the

pose, $\hat{\mathbf{x}}_0$. This posterior, called the *belief*, is given by

$$p(\mathbf{x}_k, \mathbf{p}_k | \mathbf{z}_{0:k}, \mathbf{u}_{0:k}, \hat{\mathbf{x}}_0), \tag{2.1}$$

and can be estimated in a number of different ways.

The historical approach to estimation in robotics began with the famous Kalman Filter (KF) (Swerling, 1958; Kalman, 1960; Kalman and Bucy, 1961), which was a recursive method for estimating the state of a linear-Gaussian system online. Nonlinear extensions of the KF soon followed, such as the Extended Kalman Filter (EKF)[1], the derivative-free Unscented Kalman Filter (UKF)[2] (Julier et al., 1995) and Particle Filter (PF) (Thrun et al., 2001), all of which can be thought of as approximations to the Bayes filter[3] (Jazwinski, 1970).

There have been numerous variations of the filters, including the inverse covariance form, which is ideal when the measurement dimension is large, the square-root form (Potter and Stern, 1963), which is more numerically stable, and iterated versions that iterate the measurement-update step for improved accuracy (Denham and Pines, 1966)[4]. The two common assumptions used in most of these filters are the following: (i) the motion models, measurement models, and belief are all represented by Gaussian pdfs[5] and (ii) the system follows a first-order Markov process (Markov, 1906). The Markov assumption is very powerful as it means that all prior knowledge

[1] According to Grewal and Andrews (2010), the idea behind the EKF is credited to Stanley Schmidt from NASA, who helped develop the onboard guidance and navigation system for the Apollo mission.

[2] Although both derivative-free filters, the UKF and PF are very different. The UKF belongs to a family of Linear Regression Kalman Filters (Ito and Xiong, 1999; Norgaard et al., 2000; Lefebvre et al., 2001), which use a deterministic sampling scheme for passing pdfs through a nonlinearity. The PF represents the state as a collection of random samples drawn from the posterior and uses *importance sampling* for updating final posterior (Thrun et al., 2005a).

[3] The KF was not actually derived from the Bayes filter. As Gelb states, "Kalman formulated and solved the Wiener problem for gauss-markov sequences through use of state-space representation and the viewpoint of conditional distributions and expectations." (Gelb, 1974, p. 105)

[4] According to the authors, the idea of the Iterated EKF is credited to J. V. Breakwell.

[5] The PF being an exception.

of the system is incapsulated in the current state (i.e., the conditional probability of a future state only depends on the present state and not past ones). This can be a limiting assumption, since all accrued errors are essentially baked into estimate with no way of going back.

Additionally, examining Equation (2.1), we see that it involves the joint task of building a map while concurrently using this map for localisation. This problem is known as Simultaneous Localization and Mapping (SLAM) (Durrant-Whyte, 1988; Smith et al., 1990), and was considered the "holy grail" in the robotics community for decades (Durrant-Whyte and Bailey, 2006)[6].

Early approaches to SLAM relied on recursive filtering techniques, such as the EKF (Moutarlier and Chatila, 1989a,b; Leonard and Durrant-Whyte, 1991). However, these approaches did not scale well with the size of the map[7]. Casting the problem in its information form introduced computational benefits over the standard covariance form owing to the resulting sparsity in the inverse covariance matrix (e.g., Thrun et al. (2004) for online SLAM and Thrun and Montermerlo (2006) for offline SLAM).

However, as scaling to larger environments still presented an issue, many SLAM systems began to adopt relative map representations (Newman, 1999; Bosse et al., 2004; Williams, 2001; Sibley et al., 2010; Konolige et al., 2010) to make the problem tractable. Around the same time, batch-style optimisation approaches began to gain popularity as they can provide more accuracy per unit of computing time (Strasdat et al., 2010). As such, they have dominated most of the more recent visual SLAM systems (Kaess et al., 2008; Konolige et al., 2010; Sibley et al., 2010; Piniés et al., 2010; Kaess et al., 2012).

Despite these improvements, for online SLAM, loop closure detection remains a

[6]SLAM was originally called Concurrent Mapping and Localisation.

[7]For a naive implementation of the EKF, the computational complexity scales cubically with the number of landmarks in the map, due to the inverse of the covariance matrix.

very challenging problem. Loop closures occur when the robot revisits a previously visited location and re-observes features in that location. Using the estimated pose of the vehicle to detect loop closures was the initial approach, but suffered from the fact that pose estimates drift over time. Appearance-based approaches, such as FABMAP (Cummins and Newman, 2008, 2007), offered a promising way forward as they work independently of metric information; however, avoiding false-positive loop closures remains a challenging task. The critical question being asked in the community was whether or not online SLAM was actually necessary for most practical applications. Must the "L" and "M" occur concurrently? In what situations is this a requirement?

In the domain of autonomous road vehicles operating over vast scales, the trend has been to push the SLAM problem offline so that maps can be updated, annotated, and sanity checked before any vehicle needs to operate in that environment. For realtime operation, the vehicle can then use these maps for localisation and decision making. This significantly reduces the complexity of the system as a whole as both problems do not have to be done online.

Thus, what we see are two predominant trends in robotics over the past decades: (i) casting the estimation framework as a batch-like, least-squares optimisation problem is preferred over filtering and (ii) online SLAM is not necessary for many real-world applications, such as autonomous road vehicles.

This thesis follows the same paradigm and asserts that for autonomous road vehicles, mapping is a process best left offline, and for realtime operation, the goal then become localisation against a known map:

$$p(\mathbf{x}_k|\mathbf{p}_k, \mathbf{z}_{0:k}, \mathbf{u}_{0:k}, \hat{\mathbf{x}}_0). \tag{2.2}$$

This seemingly simple change from moving the the map, $\mathbf{p}_k$, from the left-hand

side of the conditional (2.1) to the right allows us to ask interesting questions. For example, what navigation abilities are possible if one has vast amounts of prior structure and appearance data of the environment? Is it possible to leverage this prior information in an intelligent way for reliable and robust localisation?

For the remainder of this thesis, we focus on the problem of localisation against a known map. The next subsection briefly sketches out how we cast this into a batch, nonlinear least-squares framework.

2.1.1 Batch, Least Squares Formulation

The main benefit of a batch, least-squares approach comes from the fact that we no longer use the Markov assumption, and as such, we reintroduce past states, $\mathbf{x}_{0:k}$, and landmarks, $\mathbf{p}_{0:k}$, in (2.2) to give us the posterior density we seek to estimate:

$$p(\mathbf{x}_{0:k}|\mathbf{p}_{0:k}, \mathbf{z}_{0:k}, \mathbf{u}_{0:k}, \hat{\mathbf{x}}_0). \tag{2.3}$$

We now factor this density using Bayes rule (Bayes, 1764),

$$\underbrace{p(\mathbf{x}_{0:k}|\mathbf{p}_{0:k}, \mathbf{z}_{0:k}, \mathbf{u}_{0:k}, \hat{\mathbf{x}}_0)}_{\text{posterior belief}} = \eta \quad \underbrace{p(\mathbf{z}_{0:k}|\mathbf{x}_{0:k}, \mathbf{u}_{0:k}, \mathbf{p}_{0:k}, \hat{\mathbf{x}}_0)}_{\text{observed belief (likelihood)}} \quad \underbrace{p(\mathbf{x}_{0:k}|\mathbf{u}_{0:k}, \mathbf{p}_{0:k}, \hat{\mathbf{x}}_0)}_{\text{predicted belief}}, \tag{2.4}$$

where η is a normalisation factor that does not depend on the states. It should also be mentioned that although we present this as a batch estimation framework, for online performance, a sliding window is often used to limit the size of the problem (Sibley et al., 2010).

The observation model, motion model, and prior take the following form:

$$\mathbf{z}_k = \mathbf{h}(\mathbf{x}_k, \mathbf{p}_k, \mathbf{v}_k), \qquad \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k), \tag{2.5}$$

$$\mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k), \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k), \tag{2.6}$$

$$\mathbf{x}_0 = \hat{\mathbf{x}}_0 + \mathbf{n}_0, \qquad \mathbf{n}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_0), \tag{2.7}$$

where $\{\mathbf{v}_k, \mathbf{w}_k, \mathbf{n}_0\}$ are zero-mean Gaussian noise with covariances $\{\mathbf{R}_k, \mathbf{Q}_k, \mathbf{P}_0\}$, respectively. The observation model, $\mathbf{h}(\cdot)$, may, for example, explain where a point in 3D space appears in the image plane. The motion model, $\mathbf{f}(\cdot)$, typically takes interoceptive measurements, such as wheel odometry, to predict how the vehicle moved. This prediction is then refined using extereoceptive measurements from $\mathbf{h}(\cdot)$. As will be seen, the prior acts as a penalty term in the objective function and helps constrain the estimate to be within the vicinity of the prior.

By making the assumption that the noise variables are uncorrelated, we can use Bayes rule to write the observed and predicted belief as,

$$p(\mathbf{z}_{0:k}|\mathbf{x}_{0:k}, \mathbf{u}_{0:k}, \mathbf{p}_k, \hat{\mathbf{x}}_0) = \prod_{i=0}^{k} p(\mathbf{z}_i|\mathbf{x}_i, \mathbf{p}_i), \tag{2.8}$$

$$p(\mathbf{x}_{0:k}|\mathbf{u}_{0:k}, \mathbf{p}_k, \hat{\mathbf{x}}_0) = p(\mathbf{x}_0|\hat{\mathbf{x}}_0) \prod_{i=1}^{k} p(\mathbf{x}_i|\mathbf{x}_{i-1}, \mathbf{u}_i), \tag{2.9}$$

where,

$$p(\mathbf{z}_i|\mathbf{x}_i, \mathbf{p}_i) \propto \exp\left(-\frac{1}{2}\left(\mathbf{z}_i - \mathbf{h}(\mathbf{x}_i, \mathbf{p}_i, \mathbf{v}_i)\right)^T \mathbf{R}_i^{-1} \left(\mathbf{z}_i - \mathbf{h}(\mathbf{x}_i, \mathbf{p}_i, \mathbf{v}_i)\right)\right), \tag{2.10}$$

$$p(\mathbf{x}_i|\mathbf{x}_{i-1}, \mathbf{u}_i) \propto \exp\left(-\frac{1}{2}\left(\mathbf{x}_i - \mathbf{f}(\mathbf{x}_{i-1}, \mathbf{u}_i, \mathbf{w}_i)\right)^T \mathbf{Q}_i^{-1} \left(\mathbf{x}_i - \mathbf{f}(\mathbf{x}_{i-1}, \mathbf{u}_i, \mathbf{w}_i)\right)\right), \tag{2.11}$$

$$p(\mathbf{x}_0|\hat{\mathbf{x}}_0) \propto \exp\left(-\frac{1}{2}\left(\mathbf{x}_0 - \hat{\mathbf{x}}_0\right)^T \mathbf{P}_0^{-1} \left(\mathbf{x}_0 - \hat{\mathbf{x}}_0\right)\right), \tag{2.12}$$

Taking the log of the posterior yields a weighted-least squares system:

$$\log p(\mathbf{x}_{0:k}|\mathbf{p}_{0:k}, \mathbf{z}_{0:k}, \mathbf{u}_{0:k}, \hat{\mathbf{x}}_0) = \log(\kappa) + \frac{1}{2}\sum_{i=0}^{k} \left(\mathbf{z}_i - \mathbf{h}(\mathbf{x}_i, \mathbf{p}_i, \mathbf{v}_i)\right)^T \mathbf{R}_i^{-1} \left(\mathbf{z}_i - \mathbf{h}(\mathbf{x}_i, \mathbf{p}_i, \mathbf{v}_i)\right) + \frac{1}{2}\sum_{i=1}^{k} \left(\mathbf{x}_i - \mathbf{f}(\mathbf{x}_{i-1}, \mathbf{u}_i, \mathbf{w}_i)\right)^T \mathbf{Q}_i^{-1} \left(\mathbf{x}_i - \mathbf{f}(\mathbf{x}_{i-1}, \mathbf{u}_i, \mathbf{w}_i)\right) + \frac{1}{2}\left(\mathbf{x}_k - \hat{\mathbf{x}}_0\right)^T \mathbf{P}_0^{-1} \left(\mathbf{x}_k - \hat{\mathbf{x}}_0\right), \tag{2.13}$$

where κ is a constant that does not depend on the state. Defining the following quantities:

$$\mathbf{e}_{\mathbf{h},i} := \mathbf{z}_i - \mathbf{h}(\mathbf{x}_i, \mathbf{p}_i, \mathbf{v}_i), \qquad \mathbf{e}_{\mathbf{f},i} = \mathbf{x}_i - \mathbf{f}(\mathbf{x}_{i-1}, \mathbf{u}_i, \mathbf{w}_i), \qquad \mathbf{e}_{\mathbf{p}} := \mathbf{x}_0 - \hat{\mathbf{x}}_0, \tag{2.14}$$

and stacking all the error terms together gives us

$$\log p(\mathbf{x}_{0:k}|\mathbf{p}_{0:k}, \mathbf{z}_{0:k}, \mathbf{u}_{0:k}, \hat{\mathbf{x}}_0) = \log(\kappa) + \frac{1}{2}\mathbf{e}_{\mathbf{h}}^T \mathbf{R}^{-1} \mathbf{e}_{\mathbf{h}} + \frac{1}{2}\mathbf{e}_{\mathbf{f}}^T \mathbf{Q}^{-1} \mathbf{e}_{\mathbf{f}} + \frac{1}{2}\mathbf{e}_{\mathbf{p}}^T \mathbf{P}^{-1} \mathbf{e}_{\mathbf{p}}, \tag{2.15}$$

where,

$$\mathbf{e}_{\mathbf{h}} := \begin{bmatrix} \mathbf{e}_{\mathbf{h},0} \\ \vdots \\ \mathbf{e}_{\mathbf{h},K} \end{bmatrix}, \qquad \mathbf{e}_{\mathbf{f}} := \begin{bmatrix} \mathbf{e}_{\mathbf{f},1} \\ \vdots \\ \mathbf{e}_{\mathbf{f},K} \end{bmatrix}, \qquad \mathbf{P} := \mathbf{P}_0 \tag{2.16}$$

$$\mathbf{R}^{-1} := \operatorname{diag}(\mathbf{R}_0^{-1}, \ldots, \mathbf{R}_K^{-1}), \qquad \mathbf{Q}^{-1} := \operatorname{diag}(\mathbf{Q}_1^{-1}, \ldots, \mathbf{Q}_K^{-1}). \tag{2.17}$$

We now define our objective function as the terms in (2.15) that depend on the state:

$$J(\mathbf{x}) := \frac{1}{2}\mathbf{e}_{\mathbf{h}}^T \mathbf{R}^{-1} \mathbf{e}_{\mathbf{h}} + \frac{1}{2}\mathbf{e}_{\mathbf{f}}^T \mathbf{Q}^{-1} \mathbf{e}_{\mathbf{f}} + \frac{1}{2}\mathbf{e}_{\mathbf{p}}^T \mathbf{P}^{-1} \mathbf{e}_{\mathbf{p}} \tag{2.18}$$

$$= \frac{1}{2}\mathbf{e}^T \mathbf{\Sigma}^{-1} \mathbf{e}, \tag{2.19}$$

where,

$$\mathbf{e} := \begin{bmatrix} \mathbf{e_h} \\ \mathbf{e_f} \\ \mathbf{e_p} \end{bmatrix}, \qquad \mathbf{\Sigma}^{-1} = \text{diag}(\mathbf{R}^{-1}, \mathbf{Q}^{-1}, \mathbf{P}^{-1}), \tag{2.20}$$

which is known as a Mahalonobis distance (Mahalanobis, 1936).

We have now recast the estimation task as an unconstrained non-linear optimisation problem, which, importantly, does not make the limiting Markov assumption. The next subsection explains how this nonlinear, least-squares system is iteratively solved.

2.1.2 Nonlinear Numerical Solution

2.1.2.1 Newton-Raphson Method

As (2.19) is nonlinear, the standard approach is to perform a local, iterative nonlinear optimisation. We begin with Newton's method, also known as the Newton-Raphson method[8], which is an iterative root-finding technique. Geometrically, it approximates the function as being quadratic near the current guess, and uses second-order information to step towards the minimum.

Consider a second-order Taylor series expansion of our objective function, $J(\mathbf{x})$, around some operating point, $\bar{\mathbf{x}}$:

$$J(\bar{\mathbf{x}} + \delta\mathbf{x}) \approx J(\bar{\mathbf{x}}) + \mathbf{J}_{\bar{\mathbf{x}}}\delta\mathbf{x} + \frac{1}{2}\delta\mathbf{x}^T\boldsymbol{\mathcal{H}}_{\bar{\mathbf{x}}}\delta\mathbf{x}, \tag{2.21}$$

where,

$$\mathbf{J}_x := \left.\frac{\partial J(\mathbf{x})}{\partial \mathbf{x}}\right|_{\bar{\mathbf{x}}}, \qquad \boldsymbol{\mathcal{H}}_{\bar{\mathbf{x}}} := \left.\frac{\partial^2 J(\mathbf{x})}{\partial \mathbf{x}\partial \mathbf{x}}\right|_{\bar{\mathbf{x}}}, \tag{2.22}$$

are the Jacobian and Hessian, respectively.

[8]Newton originally wrote of this method in 1669 (Newton, 1968) but never published. Independently, Raphson published a very similar method in 1690 (Raphson, 1690).

Taking the derivative with respect to the perturbation and setting it to zero gives us the following:

$$\frac{\partial J(\bar{\mathbf{x}} + \delta\mathbf{x})}{\partial \delta\mathbf{x}} = \mathbf{J}_{\bar{\mathbf{x}}} + \delta\mathbf{x}^T \boldsymbol{\mathcal{H}}_{\bar{\mathbf{x}}} = \mathbf{0} \tag{2.23}$$

$$\Rightarrow \quad \boldsymbol{\mathcal{H}}_{\bar{\mathbf{x}}} \delta\mathbf{x} = -\mathbf{J}_{\bar{\mathbf{x}}}^T. \tag{2.24}$$

After we solve for our optimal step, $\delta\mathbf{x}^*$, we update our estimate according to

$$\mathbf{x} \leftarrow \bar{\mathbf{x}} + \delta\mathbf{x}^*, \tag{2.25}$$

and continue until convergence.

2.1.2.2 The Gauss-Newton Method

As the covariance term, $\boldsymbol{\Sigma}$, is symmetric, positive-definite by construction, we can factor it into its Cholesky factors, $\boldsymbol{\Psi}\boldsymbol{\Psi}^T := \boldsymbol{\Sigma}$, to rewrite (2.19) in a simpler form:

$$J(\mathbf{x}) = \frac{1}{2}\mathbf{e}^T\boldsymbol{\Sigma}^{-1}\mathbf{e} = \frac{1}{2}\mathbf{e}^T\left(\boldsymbol{\Psi}\boldsymbol{\Psi}^T\right)^{-1}\mathbf{e} = \frac{1}{2}\mathbf{e}^T(\boldsymbol{\Psi}^T)^{-1}\boldsymbol{\Psi}^{-1}\mathbf{e} = \frac{1}{2}\mathbf{r}^T\mathbf{r} \tag{2.26}$$

where we have defined $\mathbf{r} := \boldsymbol{\Psi}^{-1}\mathbf{e}$. This allows us to express the Jacobian and Hessian, defined in Equation 2.22, as follows:

$$\mathbf{J}_{\bar{\mathbf{x}}} = \mathbf{r}(\bar{\mathbf{x}})^T \left(\frac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}}\bigg|_{\bar{\mathbf{x}}} \right), \tag{2.27}$$

$$\boldsymbol{\mathcal{H}}_{\bar{\mathbf{x}}} = \left(\frac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}}\bigg|_{\bar{\mathbf{x}}} \right)^T \left(\frac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}}\bigg|_{\bar{\mathbf{x}}} \right) + \sum_i r_i(\bar{\mathbf{x}}) \left(\frac{\partial^2 r_i(\mathbf{x})}{\partial \mathbf{x} \partial \mathbf{x}}\bigg|_{\bar{\mathbf{x}}} \right). \tag{2.28}$$

If computing second-order terms is too difficult or expensive, we can make use of the fact that the residuals, $r_i(\bar{\mathbf{x}})$, near the optimum should be very small. Given

this assumption, we can approximate the Hessian as,

$$\boldsymbol{\mathcal{H}}_{\bar{\mathbf{x}}} \approx \left(\frac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}} \bigg|_{\bar{\mathbf{x}}} \right)^T \left(\frac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}} \bigg|_{\bar{\mathbf{x}}} \right). \tag{2.29}$$

Thus, our final system of equations is given by the following:

$$\left(\frac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}} \bigg|_{\bar{\mathbf{x}}} \right)^T \left(\frac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}} \bigg|_{\bar{\mathbf{x}}} \right) \delta\mathbf{x} = - \left(\frac{\partial \mathbf{r}(\mathbf{x})}{\partial \mathbf{x}} \bigg|_{\bar{\mathbf{x}}} \right)^T \mathbf{r}(\bar{\mathbf{x}}), \tag{2.30}$$

or in terms of our original parameters,

$$\left(\mathbf{E}_{\bar{\mathbf{x}}}^T \boldsymbol{\Sigma}^{-1} \mathbf{E}_{\bar{\mathbf{x}}} \right) \delta\mathbf{x} = - \left(\mathbf{E}_{\bar{\mathbf{x}}}^T \boldsymbol{\Sigma}^{-1} \right) \bar{\mathbf{e}}, \tag{2.31}$$

where $\mathbf{E}_{\bar{\mathbf{x}}} := \frac{\partial \mathbf{e}(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\bar{\mathbf{x}}}$ and $\bar{\mathbf{e}} := \mathbf{e}(\bar{\mathbf{x}})$. This approximation to Newton's method results in the normal equations for the linearised least-squares problem and is referred to as the Gauss-Newton method.

2.1.2.3 The Levenberg Marquardt (LM) Method

The problem with our first-order linearisation is that $\left(\mathbf{E}_{\mathbf{x}}^T \boldsymbol{\Sigma}^{-1} \mathbf{E}_{\mathbf{x}} \right)$ might be a poor approximation to the true Hessian around the current guess. If the approximate Hessian (Equation 2.29) is ill-conditioned, then the solution may diverge. To remedy this, we use a trust-region method called Levenberg Marquardt (LM) (Levenberg, 1944), which introduces a positive-definite matrix, $\boldsymbol{\Lambda}$ into the coefficient matrix:

$$\left(\mathbf{E}_{\bar{\mathbf{x}}}^T \boldsymbol{\Sigma}^{-1} \mathbf{E}_{\bar{\mathbf{x}}} + \boldsymbol{\Lambda} \right) \delta\mathbf{x} = - \left(\mathbf{E}_{\bar{\mathbf{x}}}^T \boldsymbol{\Sigma}^{-1} \right) \bar{\mathbf{e}}. \tag{2.32}$$

Even if $\left(\mathbf{E}_{\bar{\mathbf{x}}}^T \boldsymbol{\Sigma}^{-1} \mathbf{E}_{\bar{\mathbf{x}}}\right)$ is rank deficient, adding a positive-definite matrix will ensure the overall coefficient matrix is positive definite[9].

Levenberg Marquardt (LM) is a straightforward method that sets $\boldsymbol{\Lambda} = \lambda \mathbf{1}$, where $\mathbf{1}$ is the identity matrix and $\lambda > 0$, which is called the *damping parameter*. This damping parameter is adapted during the optimisation to change the convergence properties from either steepest descent, if the objective function is increasing, to Gauss-Newton if the optimisation is well behaved.

We use LM as the non-linear optimisation technique due to it's robustness to poorly conditioned Hessian matrices and it's ease of implementation. The LM algorithm works as follows (Hartley and Zisserman, 2004):

1. Begin with an initial estimate for the state, $\bar{\mathbf{x}}_0$.

2. Begin with an initial damping parameter, $\lambda = \lambda_0$.

3. Solve for the optimal step, $\delta \mathbf{x}^*$, given by

$$\left(\mathbf{E}_{\bar{\mathbf{x}}}^T \boldsymbol{\Sigma}^{-1} \mathbf{E}_{\bar{\mathbf{x}}} + \lambda \mathbf{1}\right) \delta \mathbf{x}^* = -\left(\mathbf{E}_{\bar{\mathbf{x}}}^T \boldsymbol{\Sigma}^{-1}\right) \bar{\mathbf{e}}. \tag{2.33}$$

4. If $|\delta \mathbf{x}^*| <$ threshold, then stop. Else, continue with the next steps.

5. Update our estimate, $\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{prev}} + \delta \mathbf{x}^*$.

6. If $J(\mathbf{x}_{\text{new}}) - J(\mathbf{x}_{\text{prev}}) > 0$, $\lambda = \beta\lambda$, where $\beta > 1$. Else, $\lambda = \eta\lambda$, where $\eta < 1$.

7. Return to Step 3.

[9]Note that $\left(\mathbf{E}_{\bar{\mathbf{x}}}^T \boldsymbol{\Sigma}^{-1} \mathbf{E}_{\bar{\mathbf{x}}}\right)$ will at least be positive semi-definite. Trivially, if $\mathbf{A} \geq 0$ and $\mathbf{B} > 0$, then $\mathbf{x}^T (\mathbf{A} + \mathbf{B}) \mathbf{x} = \underbrace{\mathbf{x}^T \mathbf{A} \mathbf{x}}_{\geq 0} + \underbrace{\mathbf{x}^T \mathbf{B} \mathbf{x}}_{>0} > 0$.

2.1.2.4 Robust Estimation

As least-squares systems are sensitive to outliers, we use a robust error metric to reduce the contribution of these outliers. Example robust estimators, or M-estimators, include Huber, Geman McClure, L1 norm, L2 norm, Cauchy, and Tukey, to name a few (Zhang, 1997). The Huber cost function is typically recommended for most problems (Zhang, 1997), so we use it in our work here.

The Huber cost function (Huber, 1981) is given as:

$$\rho(c) = \begin{cases} c^2 & |c| < \alpha \\ 2\alpha|c| - \alpha^2 & \text{otherwise,} \end{cases} \tag{2.34}$$

where α can be thought of as a threshold on the maximum allowable Mahalonobis distance, which we determine through experimentation. Returning to our objective function given by (2.26) we have,

$$J(\mathbf{x}) = \frac{1}{2}\mathbf{r}^T\mathbf{r} = \frac{1}{2}\sum_{i=0}^{i=M} \underbrace{\mathbf{r}_i^T\mathbf{r}_i}_{=:c_i^2} = \frac{1}{2}\sum_{i=0}^{i=M} c_i^2. \tag{2.35}$$

We replace each c_i term with our robust cost function (2.34), to give us a modified objective function, $\tilde{J}(\mathbf{x})$:

$$\tilde{J}(\mathbf{x}) := \frac{1}{2}\sum_{i=0}^{i=M} \rho(c_i). \tag{2.36}$$

The important aspect of this substitution is what happens to the Jacobians. Con-

sider the Jacobian of our modified objective function:

$$\frac{\partial \tilde{J}}{\partial \delta \mathbf{x}} = \frac{1}{2}\sum_{i=0}^{i=M}\frac{\partial \rho(c_i)}{\partial \delta \mathbf{x}} \tag{2.37}$$

$$= \frac{1}{2}\sum_{i=0}^{i=M}\left(\frac{\partial \rho(c_i)}{\partial ci}\right)\left(\frac{\partial c_i}{\partial \delta \mathbf{x}}\right) \tag{2.38}$$

$$= \frac{1}{2}\sum_{i=0}^{i=M}\left(\frac{\partial \rho(c_i)}{\partial ci}\right)\left(\frac{1}{2c_i}\frac{\partial c_i^2}{\partial \delta \mathbf{x}}\right) \tag{2.39}$$

$$= \frac{1}{2}\sum_{i=0}^{i=M}\left(\frac{\partial \rho(c_i)}{\partial ci}\right)\left(\frac{1}{2c_i}\right)\left(\frac{\partial c_i^2}{\partial \delta \mathbf{x}}\right). \tag{2.40}$$

Thus, we have

$$\frac{\partial \tilde{J}}{\partial \delta \mathbf{x}} = \begin{cases} \frac{1}{2}\sum\limits_{i=0}^{i=M}\frac{\partial c_i^2}{\partial \delta \mathbf{x}} & |c| < \alpha \\ \frac{1}{2}\sum\limits_{i=0}^{i=M}\frac{\alpha}{c_i}\left(\frac{\partial c_i^2}{\partial \delta \mathbf{x}}\right) & \text{otherwise,} \end{cases} \tag{2.41}$$

which can also be expressed as,

$$\frac{\partial \tilde{J}}{\partial \delta \mathbf{x}} = \frac{1}{2}\sum_{i=0}^{i=M} w_i \frac{\partial c_i^2}{\partial \delta \mathbf{x}}, \tag{2.42}$$

where,

$$w_i = \begin{cases} 1 & |c| < \alpha \\ \frac{\alpha}{c_i} & \text{otherwise.} \end{cases} \tag{2.43}$$

Writing it this way, we see that

$$\frac{\tilde{J}(\bar{\mathbf{x}} + \delta \mathbf{x})}{\partial \delta \mathbf{x}} = \frac{1}{2}\sum_{i=0}^{i=M} w_i \frac{\partial c_i^2}{\partial \delta \mathbf{x}} = \frac{1}{2}\sum_i w_i \frac{\partial\left(\mathbf{e}_i^T \boldsymbol{\Sigma}_i^{-1}\mathbf{e}_i\right)}{\partial \delta \mathbf{x}} = \frac{1}{2}\sum_{i=0}^{i=M}\frac{\partial\left(\mathbf{e}_i^T \tilde{\boldsymbol{\Sigma}}_i^{-1}\mathbf{e}_i\right)}{\partial \delta \mathbf{x}}, \tag{2.44}$$

where $\tilde{\boldsymbol{\Sigma}}_i := w_i \boldsymbol{\Sigma}_i$. This results in a final system given by the following:

$$\left(\mathbf{E}_{\bar{\mathbf{x}}}^T \tilde{\boldsymbol{\Sigma}}^{-1}\mathbf{E}_{\bar{\mathbf{x}}}\right)\delta \mathbf{x} = -\left(\mathbf{E}_{\bar{\mathbf{x}}}^T \tilde{\boldsymbol{\Sigma}}^{-1}\right)\bar{\mathbf{e}}, \tag{2.45}$$

with $\tilde{\mathbf{\Sigma}} := \text{diag}(w_0\mathbf{\Sigma}_0, \ldots, w_M\mathbf{\Sigma}_M)$, where the weights are given by (2.43).

In effect, all this is doing is ensuring that as the error increases beyond a predefined threshold, the contribution from the Jacobians in these directions is reduced (i.e., we do not take steps in the direction of the outliers). Interestingly, as just derived, this can be seen as re-weighting the inverse covariance terms during the iterative solve, leading to an alternative interpretation of increasing the uncertainty of the measurement as it's error becomes too large.

2.1.3 State Parameterisation

When it comes to representing the robot pose, special care is needed with the choice of orientation parameters, of which there are several (Stuelpnagel, 1964). Common choices for representing the orientation include: (i) Euler angles, (ii) rotation matrices, and (iii) unit-length quaternions. Euler angles suffer from singularities, which make rotation matrices and quaternions the more common choice. However, neither of these representations live in a vector space, which is a requirement for the estimation machinery derived above. Additionally, rotation matrices and quaternions introduce constraints since there are only 3 degrees of freedom for any rotation. More specifically, the updated rotation matrix must remain orthonormal, $\mathbf{R}\mathbf{R}^T = \mathbf{1}$, and the updated quaternion must be unit length $\mathbf{q}^T\mathbf{q} = 1$.

Regardless of what choice is taken, when solving Equation 2.45, the orientation representation must be in a minimal 3×1 form. The real trick comes in the update step afterwards, since, in the case for rotation matrices and quaternions, they must be updated to preserve their respective constraints, as previously mentioned[10].

In our work we use the rotation matrix to represent the orientation of the vehicle. Rotation matrices belong to the so-called *special orthogonal* SO(3) group defined

[10]For a review of how to linearise and update rotation matrices and quaternions using a unified notation, the reader is referred to Barfoot et al. (2011).

as[11]:

$$\mathrm{SO}(3) := \{\mathbf{R} \in \mathbb{R}^{3\times 3} | \mathbf{R}\mathbf{R}^T = \mathbf{1}, \det\mathbf{R} = 1\}. \tag{2.46}$$

SO(3) is a type of *matrix Lie Group*, which is a set of square matrices that are closed under products, inverses, and nonsingular limits[12] (Stillwell, 2008). The issue is that not all of the regular operations for vector spaces apply in SO(3). For example, rotation matrices are not closed under addition (i.e., you cannot add two rotation matrices and get a valid rotation matrix out). Fortunately, it turns out that we can gain access to a vector space element through the tangent space of the Lie Group[13], denoted by $\mathfrak{so}(3)$, which is closed under vector sum operations and multiplication by real numbers (Stillwell, 2008). We obtain this vector space element through the exponential map, $\mathbf{R} = e^{\mathbf{W}}$, where $\mathbf{W} \in \mathfrak{so}(3)$, which is defined as:

$$\mathfrak{so}(3) := \{\mathbf{W} \in \mathbb{R}^{3\times 3} | \mathbf{W} = \boldsymbol{\phi}^\wedge, \boldsymbol{\phi} \in \mathbb{R}^{3\times 1}\}, \tag{2.47}$$

and we have introduced the matrix-cross operator $(\cdot)^\wedge$, defined as (Murray et al., 1994):

$$\boldsymbol{\phi} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \end{bmatrix}, \qquad \mathbf{W} = \boldsymbol{\phi}^\wedge := \begin{bmatrix} 0 & -\phi_3 & \phi_2 \\ \phi_3 & 0 & -\phi_1 \\ -\phi_2 & \phi_1 & 0 \end{bmatrix}. \tag{2.48}$$

Using the well-known Euler-Rodrigues rotation formula (Euler, 1770; Rodrigues, 1816), we can express any rotation matrix in terms of an angle of rotation, ϕ, about some unit-length axis, $\mathbf{a} \in \mathbb{R}^{3\times 1}$, according to

$$\mathbf{R} = e^{\phi\mathbf{a}^\wedge} = \cos\phi\mathbf{1} + (1 - \cos\phi)\mathbf{a}\mathbf{a}^T + \sin\phi\mathbf{a}^\wedge. \tag{2.49}$$

[11] We follow the notation by Stillwell (2008) and use uppercase SO(3) to represent the Lie group and $\mathfrak{so}(3)$ to represent the Lie Algebra.

[12] A set is said to be closed under an operation if performing the operation on any member of the set always produces another member of the set.

[13] The tangent space at the identity of the Lie Group along with a set of binary operations, called the *Lie Bracket*, constitute the *Lie Algebra* (Stillwell, 2008).

For reasons that will become clear shortly, it is then useful to define the following:

$$\boldsymbol{\phi} := \phi \mathbf{a}, \tag{2.50}$$

where $\phi := ||\boldsymbol{\phi}||$. This changes the rotation formula to

$$\mathbf{R} = e^{\boldsymbol{\phi}^\wedge} = \cos\phi \mathbf{1} + \frac{(1 - \cos\phi)}{\phi^2}\boldsymbol{\phi}\boldsymbol{\phi}^T + \frac{\sin\phi}{\phi}\boldsymbol{\phi}^\wedge. \tag{2.51}$$

Our 3×1 minimal representation, $\boldsymbol{\phi}$, is known as a rotation vector.

Defining our state vector as

$$\mathbf{x} := \begin{bmatrix} \mathbf{t}_c^{w,c} \\ \boldsymbol{\phi}_c \end{bmatrix}, \tag{2.52}$$

we can construct an SE(3) transformation matrix[14] according to

$$\mathbf{T}_{c,w}(\mathbf{x}) = \begin{bmatrix} \mathbf{R}_{c,w}(\boldsymbol{\phi}_c) & \mathbf{t}_c^{w,c} \\ \mathbf{0} & 1 \end{bmatrix}, \tag{2.53}$$

where $\mathbf{R}_{c,w}(\boldsymbol{\phi}_c)$ is given by Equation (2.51) and represents the rotation from $\underrightarrow{\mathcal{F}}_w$ to $\underrightarrow{\mathcal{F}}_c$ and $\mathbf{t}_c^{w,c}$ represents the vector from $\underrightarrow{\mathcal{F}}_c$ to $\underrightarrow{\mathcal{F}}_w$ but expressed in $\underrightarrow{\mathcal{F}}_c$.

As mentioned earlier, we require members of a vector space to use the estimation tools derived above. Thus, we need to linearise $\mathbf{T}(\mathbf{x})$ in order to gain access to our vector representation, $\mathbf{x}$. There are two ways to do this. Firstly, it is important to note the following[15]:

$$\mathbf{T}(\bar{\mathbf{x}} + \delta\mathbf{x}) \not\approx \mathbf{T}(\delta\mathbf{x})\mathbf{T}(\bar{\mathbf{x}}). \tag{2.54}$$

Furgale (2011) presents a derivation of how to linearise $\mathbf{T}(\bar{\mathbf{x}} + \delta\mathbf{x})$ using a pertur-

[14] Another type of Lie Group.

[15] For rotation matrices, it can be shown that up to first order: $\mathbf{R}(\bar{\boldsymbol{\phi}} + \delta\boldsymbol{\phi}) \approx \mathbf{R}(\delta\boldsymbol{\phi})\mathbf{R}(\bar{\boldsymbol{\phi}})$. However, for transformation matrices, this is not the case.

bation method. The other approach considers the right-hand side of (2.54) and parameterises a delta rotation about the current estimate, as is done in several works (Churchill, 2012; Strasdat, 2012; Sibley et al., 2010). In this way, the initial transformation estimate is held fixed and the goal becomes optimising a transformation around this initial condition. We adopt this approach, but instead of using generator matrices to compute the Jacobians, we show how to obtain them using the perturbation method shown in Barfoot et al. (2011).

2.1.4 Jacobians for Expressions Involving Rotation Matrices

We shall present this method through an example, but begin by first introducing homogeneous operators as they prove to be useful when transforming points with transformation matrices.

The homogeneous operator, $\boldsymbol{g}(\cdot)$, and it's inverse, $\mathbf{g}(\cdot)$, defined as follows:[16]

$$\boldsymbol{p} := \boldsymbol{g}(\mathbf{p}) = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix}, \qquad \mathbf{p} = \mathbf{g}(\boldsymbol{p}) = \frac{1}{p_N} \begin{bmatrix} p_1 \\ \vdots \\ p_{N-1} \end{bmatrix}, \tag{2.55}$$

for $\boldsymbol{p} \in \mathbb{R}^{N\times 1}$. Note that we use boldface for $\mathbf{p} \in \mathbb{R}^{3\times 1}$ and bold italicised for its homogenous coordinate $\boldsymbol{p} \in \mathbb{R}^{4\times 1}$.

Consider a point, $\mathbf{p}_w$, expressed in the world frame, $\underrightarrow{\mathcal{F}}_w$, and we wish to transform this point into the camera frame, $\underrightarrow{\mathcal{F}}_c$, through the transformation matrix, $\mathbf{T}_{c,w}$. We can accomplish this transformation using the homogeneous operators:

$$\mathbf{p}_c = \mathbf{g}(\mathbf{T}_{c,w}\boldsymbol{p}_w) = \mathbf{R}_{c,w}\mathbf{p}_w + \mathbf{t}_c^{w,c}, \tag{2.56}$$

[16]Although $\boldsymbol{h}(\cdot)$ may seem more appropriate, recall that $\mathbf{h}(\cdot)$ is reserved for the observation model and we don't wish to overload any operator.

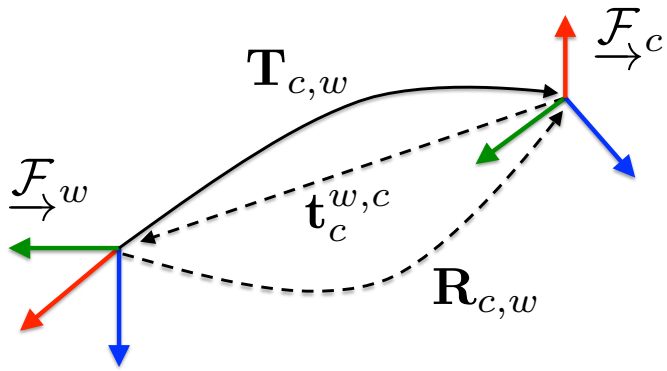


Figure 2.1: Coordinate reference frames along with notation representation transformation matrices, rotation matrices, and a position vector. Note the subscripts and superscripts, and their meaning with regards to direction.

where $\mathbf{R}_{c,w}$ is the transformation matrix from $\underrightarrow{\mathcal{F}}_w$ to $\underrightarrow{\mathcal{F}}_c$ and $\mathbf{t}_c^{w,c}$ is the translation from $\underrightarrow{\mathcal{F}}_c$ to $\underrightarrow{\mathcal{F}}_w$ expressed in $\underrightarrow{\mathcal{F}}_c$ (see Figure 2.1.4). Now consider perturbing our transformation matrix from its current guess:

$$\bar{\mathbf{p}}_c + \delta\mathbf{p}_c = \mathbf{g}(\mathbf{T}(\delta\mathbf{x})\bar{\mathbf{T}}_{c,w}\boldsymbol{p}_w) = \mathbf{R}(\delta\boldsymbol{\phi})\bar{\mathbf{R}}_{c,w}\mathbf{p}_w + \mathbf{R}(\delta\boldsymbol{\phi})\bar{\mathbf{t}}_c^{w,c} + \delta\mathbf{t}_c. \tag{2.57}$$

At this stage, we turn to Equation (2.51) and use the fact that we are considering an infinitesimal rotation about the axis of rotation, which gives us the following approximation:

$$\mathbf{R}(\delta\boldsymbol{\phi}) \approx \mathbf{1} + \delta\boldsymbol{\phi}^\wedge, \qquad \text{for } \delta\phi \ll 1, \tag{2.58}$$

Substituting Equation (2.58) into (2.57), we have

$$\begin{aligned}
\bar{\mathbf{p}}_c + \delta\mathbf{p}_c &\approx (\mathbf{1} + \delta\boldsymbol{\phi}_c^\wedge)\,\bar{\mathbf{R}}_{c,w}\mathbf{p}_w + (\mathbf{1} + \delta\boldsymbol{\phi}_c^\wedge)\,\bar{\mathbf{t}}_c^{w,c} + \delta\mathbf{t}_c, && (2.59)\\
&= \underbrace{\bar{\mathbf{R}}_{c,w}\mathbf{p}_w + \bar{\mathbf{t}}_c^{w,c}}_{=\bar{\mathbf{p}}_c} + \delta\boldsymbol{\phi}_c^\wedge \underbrace{\left(\bar{\mathbf{R}}_{c,w}\mathbf{p}_w + \bar{\mathbf{t}}_c^{w,c}\right)}_{=\bar{\mathbf{p}}_c} + \delta\mathbf{t}_c && (2.60)\\
&= \bar{\mathbf{p}}_c + \underbrace{\delta\boldsymbol{\phi}_c^\wedge\bar{\mathbf{p}}_c}_{=-\bar{\mathbf{p}}_c^\wedge\delta\boldsymbol{\phi}_c} + \delta\mathbf{t}_c && (2.61)\\
&= \bar{\mathbf{p}}_c - \bar{\mathbf{p}}_c^\wedge\delta\boldsymbol{\phi}_c + \delta\mathbf{t}_c && (2.62)\\
\implies \delta\mathbf{p}_c &= -\bar{\mathbf{p}}_c^\wedge\delta\boldsymbol{\phi}_c + \delta\mathbf{t}_c, && (2.63)
\end{aligned}$$

where we have made use of the identity, $\mathbf{a}^{\wedge}\mathbf{b} = -\mathbf{b}^{\wedge}\mathbf{a}$, for any 3×1 vectors $\mathbf{a}$ and $\mathbf{b}$.

Noting that

$$\delta\mathbf{p}_c = \frac{\partial \mathbf{g}}{\partial \mathbf{x}_c}\delta\mathbf{x}_c, \tag{2.64}$$

we arrive at

$$\frac{\partial \mathbf{g}}{\partial \mathbf{x}_c}\delta\mathbf{x}_c = \begin{bmatrix}\mathbf{1} & -\bar{\mathbf{p}}_c^{\wedge}\end{bmatrix}\begin{bmatrix}\delta\mathbf{t}_c \\ \delta\boldsymbol{\phi}_c\end{bmatrix} \tag{2.65}$$

$$\implies \boxed{\frac{\partial \mathbf{g}}{\partial \mathbf{x}_c} = \begin{bmatrix}\mathbf{1} & -\bar{\mathbf{p}}_c^{\wedge}\end{bmatrix}} \tag{2.66}$$

since

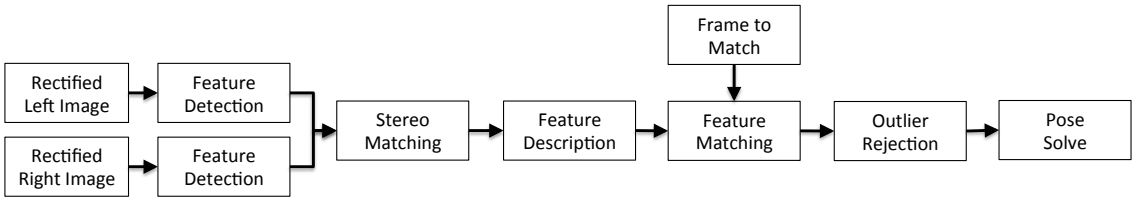
$$\delta\mathbf{x}_c := \begin{bmatrix}\delta\mathbf{t}_c \\ \delta\boldsymbol{\phi}_c\end{bmatrix}. \tag{2.67}$$

To summarise, we took a nonlinear point-transformation equation (2.56), pre-multiplied our current state guess, $\bar{\mathbf{T}}_{c,w}$, by a delta transformation matrix, which was a function of a delta translation and rotation vector (2.67). We then used a small angle approximation to express the exponential map of $\delta\boldsymbol{\phi}_c$ in a simplified state (2.58). Through basic matrix manipulations, we were able to factor out $\delta\mathbf{x}_c$ from our equation and solved for the derivative.

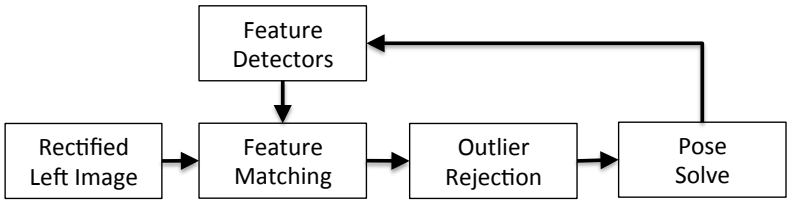
2.2 Stereo Visual Odometry and Localisation

As it is an essential tool used throughout this thesis, this section briefly sketches our keyframe-based stereo Visual Odometry (VO) pipeline.

Simply stated, the task of frame-to-frame VO is to match currently observed visual features with previously observed visual features and then estimate the pose of the vehicle given these associations. The pose estimation task can be divided into two main components: (i) the visual front-end responsible for matching features



(a) Stereo VO/Localisation pipeline, where features are detected, described and matched to some other frame. For frame-to-frame VO, the "Frame to Match" would be the previous frame and for localisation, the "Frame to Match" would be a map frame..



(b) Our localisation approach described in Chapter 5, which uses a bank of place-dependent feature detectors to perform the detection and data association concurrently.

Figure 2.2: Top figure: our feature-based, stereo VO/localisation pipeline. Bottom figure: our proposed localiser approach .

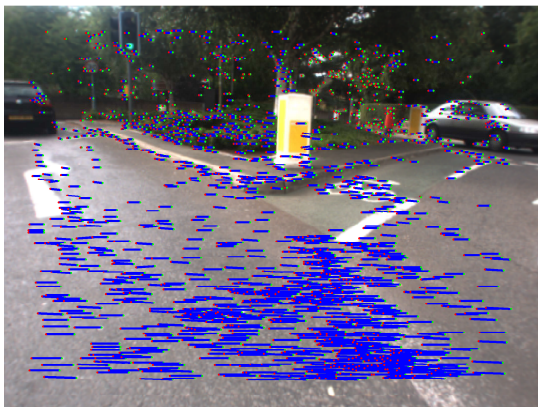
from the live view to some previous view, and (ii) the estimation backend, which performs the nonlinear solve using the techniques described in Section 2.1.

The main steps involved in our VO system are outlined below and shown graphically in Figure 2.2(a).

1. **Undistortion and Rectification**: Makes the images appear to have come from an idealised pin-hole camera model and enforces parallel epipolar lines between the stereo pair, which is important during the stereo matching step.

2. **Feature Detection**: Detect 2D interest points at multiple scales. Interest points are points in the image that can be detected reliably under (modest) viewpoint changes. A vast amount of research has been done to engineer various feature detectors for performance and speed. Popular corner detectors include Harris Corners (Harris and Stephens, 1988) and FAST (Rosten et al., 2005), while blob detection can be performed with the Laplacian of Gaussian or MSER (Matas et al., 2002). In our system, we use the FAST detector,

(a) Raw FAST corners detected in the left image of the stereo pair. Each detection has a score for how "corner-like" it is. Typically a minimum score threshold is set to prune the thousands of potential candidates.



(b) Left-to-right stereo matches rendered onto the left image. The green dot represents the pixel location in the left image and the red dot in the right. The measurements are shown as blue lines to highlight the fact that the disparity, $d := u_l - u_r$, is larger in the near field than far field. This has implications regarding measurement uncertainties, which will be discussed in the next chapter.

(c) Temporal matching of the features with the previous frame and rendered on the left image. The direction of the feature matches illustrates the motion of the camera. Note the outliers in this initial round of matching (ground and car in the back).

(d) Post-RANSAC feature matches. Note that when the majority of matches belong to the background, RANSAC is able to remove outliers on moving vehicles (top right). However, if a significant portion of the feature matches belong to moving objects, RANSAC can often fail, since it's designed to provide the guess that produces the biggest consensus set. In the next chapter, we present a technique that can mask distractions and prevent this from occurring.

Figure 2.3: Examples of feature detection, stereo matching, temporal matching, and outlier rejection in a standard VO pipeline.

which finds thousands of candidate corners (see Figure 2.3(a)).

3. **Stereo Matching**: Match features between the left and right images using scanline matching. Since the images have been rectified, we can search along each row for associations. This works as follows. For each FAST corner in the left image, we perform a 1D search in the right image to find the position with the lowest Sum of Absolute Difference (SAD) score. After we have matched left-to-right, we have a collection of stereo measurements of the form, $\mathbf{z} = [u_l, v_l, d]^T$, which contains the pixel positions in the left images, along with the disparity, which is defined as $d := u_l - u_r$. Some example stereo measurements are shown in Figure 2.3(b).

4. **Feature Description**: Compute descriptors for each interest point, which act as unique identifiers that are used to find associations from one frame to the next. Again, there are numerous descriptors to choose from. Examples include Scale Invariant Feature Transform (SIFT) (Lowe, 2004), Speeded-Up Robust Features (SURF) (Bay et al., 2008), Binary Robust Independent Elementary Features (BRIEF) (Calonder et al., 2012), and Oriented FAST and Rotated BRIEF (ORB) (Rublee et al., 2011), to name a few. We use the BRIEF descriptor due to its fast matching performance.

5. **Feature Matching**: Match features to previous frame using descriptors to produce a candidate set of matches. In our system, we also perform sub pixel refinement using Efficient Second-Order Minimisation (Mei et al., 2008), which uses SAD as the cost function. An example of temporal matching is shown in Figure 2.3(c). The initial round of feature matching is used to seed the patch-based SAD matching.

6. **Outlier Rejection**: Since not all of the candidate matches are inliers, we require outlier rejection before proceeding with the pose solve. Three-point

RANSAC is used, which iterates over N trials using the following strategy. For each trial, one randomly selects three points to construct a transformation matrix, $\mathbf{T}_{c,w}$, which transforms points from $\underrightarrow{\mathcal{F}}_w$ to $\underrightarrow{\mathcal{F}}_c$. Note that in our case, $\underrightarrow{\mathcal{F}}_w$, would be the previous frame, but in the more general case, this could be a map reference frame. This candidate transformation is then used to reproject the points defined in $\underrightarrow{\mathcal{F}}_w$ into the image plane, $\underrightarrow{\mathcal{F}}_c$, according to our stereo measurement model:

$$\boxed{\mathbf{z}_c := \mathbf{h}(\mathbf{K}\mathbf{T}_{c,w}\boldsymbol{p}_w) = \frac{1}{z}\begin{bmatrix} x\ f_u + z\ c_u \\ y\ f_v + z\ c_v \\ f_u\ b \end{bmatrix}} \tag{2.68}$$

where

$$\mathbf{K} := \begin{bmatrix} f_u & 0 & c_u & 0 \\ 0 & f_v & c_v & 0 \\ 0 & 0 & 0 & f_u\ b \\ 0 & 0 & 1 & 0 \end{bmatrix}, \tag{2.69}$$

and $\{f_u, f_v\}$ are the horizontal and vertical focal lengths, b is the baseline, $\{c_u, c_v\}$ are the horiztonal and vertical positions of the optical centre, $\boldsymbol{p}_w$ is a point defined in $\underrightarrow{\mathcal{F}}_w$, and the $\{x, y, z\}$ values are components of $\mathbf{p}_c$. The reprojection error is computed for each point and the number of inliers is recorded. The procedure repeats and the transformation that produces the largest inlier set is chosen as the initial guess for the pose solve. Outlier matches are removed from the set (see Figure 2.3(d)).

7. **Pose Solve**: Using the techniques described earlier, we can iteratively solve a non-linear, least-squares system of the form shown in (2.45), which in this

case is given by,

$$\left(\mathbf{H}_{\mathbf{x}}^{T}\mathbf{R}^{-1}\mathbf{H}_{\mathbf{x}} + \lambda\mathbf{1}\right)\delta\mathbf{x} = -\left(\mathbf{H}_{\mathbf{x}}^{T}\mathbf{R}^{-1}\right)\bar{\mathbf{e}}, \tag{2.70}$$

where

$$\mathbf{H}_{\mathbf{x}} := \frac{\partial\mathbf{h}}{\partial\mathbf{x}_c}, \quad \bar{\mathbf{e}} := \begin{bmatrix} \mathbf{z}_0 - \mathbf{h}(\mathbf{T}_{c,w}, \mathbf{p}_w^0) \\ \vdots \\ \mathbf{z}_M - \mathbf{h}(\mathbf{T}_{c,w}, \mathbf{p}_w^M) \end{bmatrix}, \quad \mathbf{R} := \mathrm{diag}(\mathbf{R}_1, \ldots, \mathbf{R}_M). \tag{2.71}$$

The Jacobian term, $\mathbf{H}_{\mathbf{x}}$, can be separated into two factors according to the chain rule:

$$\frac{\partial\mathbf{h}}{\partial\mathbf{x}_c} = \frac{\partial\mathbf{h}}{\partial\mathbf{p}_c}\frac{\partial\mathbf{p}_c}{\partial\mathbf{x}_c}. \tag{2.72}$$

The first term is easily computed as

$$\frac{\partial\mathbf{h}}{\partial\mathbf{p}_c} := \begin{bmatrix} f_u/z & 0 & -f_u x/z^2 \\ 0 & f_v/z & -f_v y/z^2 \\ 0 & 0 & -f_u\ b/z^2 \end{bmatrix}. \tag{2.73}$$

Recalling the point-transformation function (2.56),

$$\mathbf{p}_c := \mathbf{g}(\mathbf{T}_{c,w}\boldsymbol{p}_w) = \mathbf{R}_{c,w}\mathbf{p}_w + \mathbf{t}_c^{w,c}, \tag{2.74}$$

we note that the second term, $\partial\mathbf{p}_c/\partial\mathbf{x}_c = \partial\mathbf{g}/\partial\mathbf{x}_c$, was already derived in the previous section. Thus, our stereo model Jacobian is given by the following:

$$\boxed{\frac{\partial\mathbf{h}}{\partial\mathbf{x}_c} = \begin{bmatrix} f_u/z & 0 & -f_u x/z^2 \\ 0 & f_v/z & -f_v y/z^2 \\ 0 & 0 & -f_u\ b/z^2 \end{bmatrix} \begin{bmatrix} \mathbf{1} & -\mathbf{p}_c^{\wedge} \end{bmatrix}} \tag{2.75}$$

We will refer to this Jacobian later in Chapter 5.

Note that by simply swapping out the"previous frame" with an archived frame from the map, we can turn the VO pipeline into a localisation system, which is used in Chapters 3 and 4. In Chapter 5, we will present a new localisation system that replaces the point-feature front end (i.e., it replaces the feature detection and description blocks in the pipeline) with a bank of pre-trained, place-specific classifiers. This approach does not simply apply the same rigid procedure to all incoming images, but rather, is specifically tuned to the vehicle's current location (see Figure 2.2(b)).

2.3 Summary

This chapter began with a theoretical discussion surrounding the estimation tools used in the remainder of this thesis. The estimation problem was initially presented in a Bayesian framework and we showed how to extend this interpretation to a non-linear least-squares problem. The nonlinear optimisation, robust cost function, and state parameterisation were then discussed, which are all important topics that will be mentioned in the chapters to come. We then introduced a core vision technique called Visual Odometry (VO), which will be referenced throughout the thesis and will be used as the baseline for comparison in Chapter 5.

As discussed in the introduction, the problem being addressed in this thesis is how to cope with visual change for long-term, persistent localisation. We identified a number of sources of visual change, including: (i) dynamic objects (i.e., distractions), (ii) illumination changes, (iii) weather, and (iv) seasonal change. The next chapter will examine the problem of navigating in heavily cluttered environments and will discuss why this is challenging and how we can combat these problems by leveraging knowledge of prior 3D structure.

Chapter 3

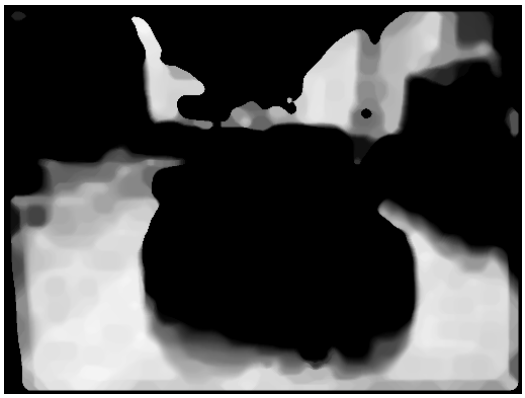
Distraction Suppression

3.1 Introduction

For vision-based navigation systems, operating in highly dynamic environments is a challenging problem as extreme scene motion can degrade standard outlier rejection schemes and result in erroneous motion estimates. In this chapter, we present an approach to the problem of pose estimation in heavily cluttered urban environments by leveraging knowledge of prior 3D structure for distraction suppression in images. In other words, given prior knowledge of how the world "should look", our system is able to focus its attention on just the static parts of the scene for motion estimation, even in situations where most of the image is completely obscured by dynamic objects (see Figure 3.1 for an example).

Although one may approach this problem with a trained detector and tracking system (e.g., Horbert et al. (2011); Ess et al. (2010); Leibe et al. (2008)), we note that these techniques require a great deal of time to train, are challenging to implement, and require knowledge of all of the various distraction classes. In contrast, we present a straightforward and effective approach that exploits prior 3D structure to generate *background-likelihood images*, which effectively mask ephemeral objects of any type. Offline, we use a 3D object detection technique to produce segmented background

(a) Image taken in central London during the Olympics, where large parts of the scene are occupied by dynamic objects, which can distract and impede egomotion estimation. We present two techniques that leverage knowledge of prior structure to enable robust pose estimation, even in cases where most of the scene is moving.



(b) Using knowledge of prior 3D structure, we can generate probability masks that indicate which regions in the image are likely to belong to the static background (white). These masks are used in our front-end visual odometry pipeline to improve pose estimation in the presence of significant scene motion.

Figure 3.1: This chapter presents a technique to exploit knowledge of prior 3D structure to enable accurate pose estimation in heavily cluttered, highly dynamic environments. Having driven a route at least once, we are able to leverage prior information to suppress distracting objects in the image and focus on just the static parts of the scene, which is represented as a *background-likelihood image* (see 3.1(b)). This background-likelihood image is used to mask ephemeral objects, thereby enabling accurate feature matching, even in situations where most of the scene is moving.

priors, but online we do not require any object detectors since we are just comparing observed structure with predicted structure.

Thus, we are not specifically interested in object detection per se, but rather, *scene relevance*—what should we be focusing on in the scene, given that we have prior knowledge of its structure and simply wish to localise and perform egomotion estimation. As will be shown later, egomotion estimation is critical to our localisation system as it fills in the gaps between localisation estimates and also predicts the vehicle's location in the map. This chapter will present results on kilometres of data collected in busy urban environments, demonstrating how these techniques can improve the robustness of VO.

3.2 Related Work

In the area of road-vehicle navigation, leveraging prior surveys to improve motion estimation is a common approach. Numerous techniques exist for both vision and laser and include: (i) combining vision with aerial images (Napier et al., 2010; Pink and Stiller, 2010), synthetic overhead images (Napier and Newman, 2012), or prior visual experiences (Churchill and Newman, 2012), (ii) combining 2D laser rangefinders with 2D priors (Bosse and Zlot, 2008) (iii) combining 2D laser rangefinders with 3D priors (Baldwin and Newman, 2012), (iv) combining 3D laser rangefinders with 3D priors (Levinson et al., 2010), and (v) combining vision with 3D priors (Stewart and Newman, 2012). This work considers the latter case of using vision sensors in conjunction with a prior 3D survey generated from a laser scanner. The goal is to identify areas in an image that have a high likelihood of belonging to the static background, even if 90% of the image is obscured by dynamic objects. These background-likelihood scores are used in the front-end VO system to mask features detected on ephemeral objects and thus, improve outlier rejection in the VO system.

The methods described in this chapter rely on the idea of background subtraction, which have traditionally been applied to static camera systems for surveillance operations (Haritaoglu et al., 2000; Wren et al., 1997). The typical approach is to learn a statistical model of the background (e.g., Mixture of Gaussians (MoG) for each pixel in the image) and compare current views with the background model to identify large discrepancies. Over time, the background models are adapted to account for both immediate and long-term temporal changes of the environment (see Piccardi (2004) for a review of the various statistical models that have been used). Connected-component analysis is often used to cluster these outlying pixels for tracking.

Salas and Tomatsi (2011) presented an object detection and tracking system

using a Kinect sensor. They considered a static indoor setup and assumed they had an uncluttered background model a priori. They incorporated two levels of detection: one based on a MoG background subtraction technique using the 3D prior, and the other based on a HOG detector in appearance space. Kaestner et al. (2010) presented a MoG background subtraction method for 3D laser data taken from a stationary platform. They learn both the statistical background model along with the model parameters online. Li et al. (2008) presented a detection method for a camera surveillance system that used a 3D model of the environment to improve their search strategy. Their method worked by performing the object search within the 3D grid and rectifying sub-images to account for perspective distortion.

The above mentioned methods were designed for static camera and/or laser setups and are therefore not applicable to mobile platforms. Various techniques for moving systems have been proposed, such as estimating a planar homography and applying the standard statistical techniques for foreground/background detection (Ren et al., 2003; Hayman and olof Eklundh, 2003); however, these methods are only valid under rotational motion. Plane-parallax constrains were introduced to compensate for rotational and translational motions (Yuan et al., 2007; Irani and Anandan, 1998), but assume that a dominant 3D plane is present. Sheikh et al. (2009) offer a different solution that estimates a background trajectory based on a rank constraint for a sequence of tracked point trajectories. However, these methods have only demonstrated results under modest displacements and not in outdoor settings with fast-moving vehicles in cluttered environments. Additionally, these methods do not combine two different sensing modalities, which is one of the novel aspects of our work.

Segmenting the scene based on motion cues is another common approach. The various strategies for motion segmentation include clustering dense point trajectories based on optical flow (Namdev et al., 2012; Narayanan Sundaram and Brox, 2011),

dense scene flow (Alcantarilla et al., 2012; Wedel et al., 2011), sparse scene flow (Lenz et al., 2011), or geometric consistency and scale (Muller et al., 2008). However, these methods could potentially breakdown in situations where the dominant motion in the scene is generated from dynamic objects that were initially at rest.

Taneja et al. (2011) presented an offline monocular-based background subtraction technique for detecting and updating changes in a prior 3D model. Their method uses prior structure to reproject pixels from the current camera frame into a collection of neighbouring frames to identify geometric inconsistencies. The problem is then formulated as the minimisation of a Gibbs energy function to find the optimal labelling of their voxelised prior (i.e., changed or unchanged). Similar to this work, we use prior 3D structure to identify regions of change in camera images, but take two very different approaches that attempt to account for uncertainties resulting from localisation errors. Furthermore, we demonstrate how to generate background-likelihood images and integrate them into a VO pipeline.

3.3 System Overview

Our system operates with the requirement that the workspace must be pre-mapped by a survey vehicle equipped with 3D laser sensors, cameras, and an Inertial Navigation System (INS). More specifically, we assume that preprocessed 3D point-clouds and stereo imagery of the environment will be available (see Figure 3.2). Furthermore, we assume that these point clouds are free of most ephemeral objects.

At a high-level, our system works as follows. At runtime, we match live stereo images against prior visual experiences (i.e., visually distinct image sequences). Since each prior visual experience has an associated 3D point-cloud, we are able to synthesise depth images from estimated camera poses in this 3D prior. These synthetic depth images are then used to compare the current structure of the scene (given

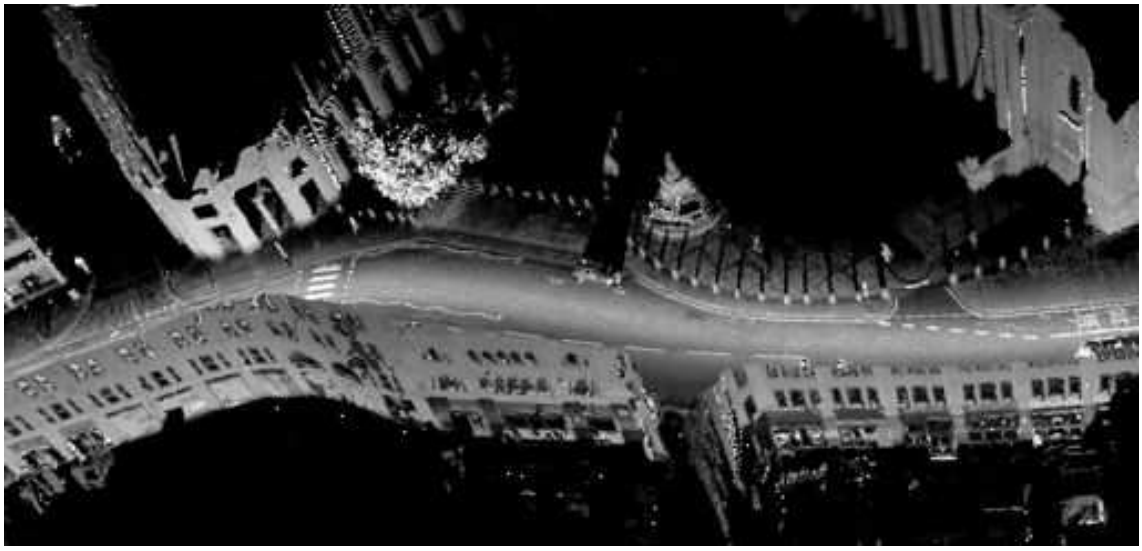


Figure 3.2: A laser-generated 3D point cloud in central London. This data was collected with a mobile sensing suite mounted on a commercial vehicle, equipped with a stereo camera, planar laser rangefinder, and INS. At runtime our system uses the stereo input to compare the observed structure of the world with the prior structure to identify ephemeral objects.

by our live imagery) with the static structure of scene (given by the prior) to identify large discrepancies. This provides us with a clean segmentation of the image into foreground and background elements, without the need for an object detection system.

This chapter presents two vision-based techniques for a moving platform that exploit prior 3D structure from a laser-generated point cloud to create distraction masks, or *background-likelihood images*, which provide pixel-wise likelihood scores for belonging to the background. These likelihood images are used in the front-end of the VO pipeline to reject candidate features on ephemeral objects and improve pose estimation, which is of great importance for autonomous vehicles operating in urban environments. Figure 3.3 shows where this sits in the VO pipeline introduced in Section 2.2.

The proceeding sections will describe in more detail how the system generates synthetic camera views from the 3D scene prior, how this information is used to

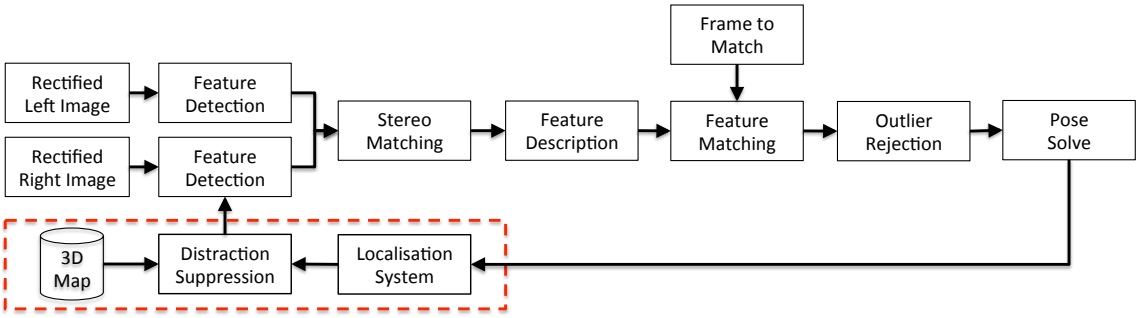


Figure 3.3: Illustration of where distraction suppression fits within the VO pipeline introduced in Section 2.2. The distraction suppression module uses a 3D scene prior and a localisation estimate to produce a distraction mask, which is used in the feature detection step to prune features detected on ephemeral objects. Note that the "localiser" depends on the VO pose for motion prediction.

generate background-likelihood images, and how these likelihood scores are incorporated in our VO pipeline.

3.3.1 3D Sceneprior

The 3D sceneprior is constructed offline using the output of the VO for motion estimation. The laser measurements come in as scanlines with range and reflectance values at fixed angular positions. Using the camera-to-laser calibration allows us to express the i^{th} point at time t_k in the camera frame, which we represent as $\mathbf{p}_k^i$. We can then interpolate between the two closest transformations from the VO pose chain, $\{\mathbf{T}_{0,m}, \mathbf{T}_{0,n}\}$, with $t_m \leq t_k < t_n$ to compute the pose of the laser expressed in the base frame of the map: $\mathbf{p}_0^i = \hat{\mathbf{T}}_{0,k}\mathbf{p}_k^i$.

Note that we store all of the points in a global reference frame for convenience, but at runtime, transform points in a local window around the estimated camera position, making this a completely relative approach. Figure 3.4 illustrates this point. As our VO is subject to drift, the position estimates in a global frame will be incorrect. However, since we are only interested in making observations and acting locally, we need not concern ourselves with constructing a globally consistent map.

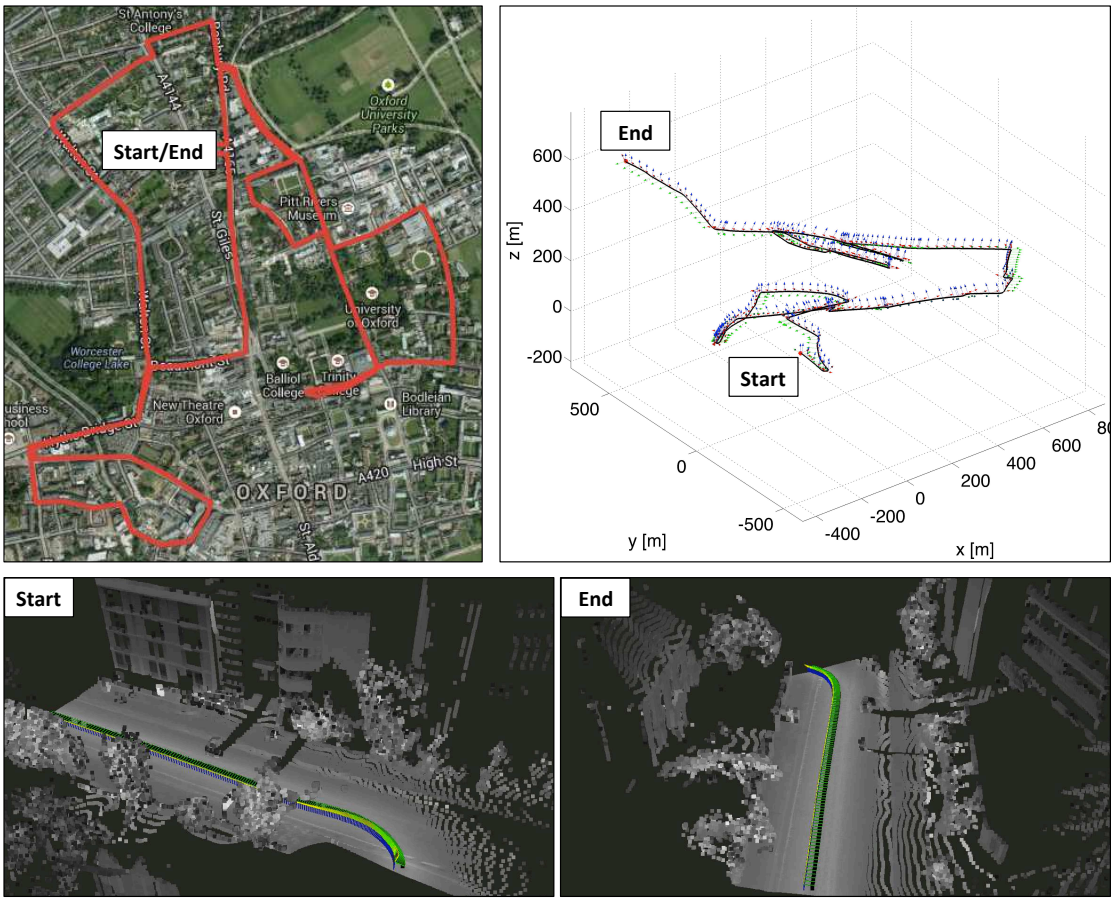


Figure 3.4: Our map representation. The top left figure represents the route driven in a globally consistent frame. The top right figure represents the output of our VO, which is prone to significant drift as a function of distance traveled. For the task of localisation, we do not need to create a globally consistent map, since we are only concerned with local observations within this map. Thus, by transforming a local window of points around the vehicle's position, we can construct locally consistent maps that can be used for navigation (bottom two images).

After constructing the scene prior, we use the 3D detection method of Wang et al. (2012) to detect cyclists, pedestrians, and vehicles, which are removed from the point cloud. Their technique solves the binary classification task of foreground vs. background first and then uses an unsupervised graph-based clustering technique to segment the point cloud into multiple entities.

3.3.2 Generating Synthetic Camera Views

During operation, our vision-based localisation system provides an estimate of the pose of the vehicle from the map frame, $\underrightarrow{\mathcal{F}}_m$, to the camera frame, $\underrightarrow{\mathcal{F}}_c$ denoted by the transformation matrix, $\mathbf{T}_{c,m}$. Using this estimated pose, we reproject all of the points from the 3D scene prior into the camera frame, producing a synthetic depth image (see Figure 3.5). For reasons of efficiency, we restrict the size of the 3D scene prior by using a sliding window about the estimated camera position[1]. Thus, for every point in the map frame, $\mathbf{p}_m^i$, we transform the point into the camera frame according to the point-transformation equation:

$$\mathbf{p}_c = \mathbf{g}(\mathbf{T}_{c,m}\boldsymbol{p}_m) = \mathbf{R}_{c,m}\mathbf{p}_m + \mathbf{t}_c^{m,c}. \tag{3.1}$$

As we are interested in the depth of a given pixel, z_i, we take the last component of our transformed point:

$$z_i = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{p}_c^i = \underbrace{\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathbf{g}\left(\mathbf{T}_{c,m}\boldsymbol{p}_m^i\right)}_{=:z(\mathbf{T}_{c,m}\boldsymbol{p}_m^i)}, \tag{3.2}$$

where we have defined our depth-extraction function, $z\left(\mathbf{T}_{c,m}\boldsymbol{p}_m^i\right)$, for convenience.

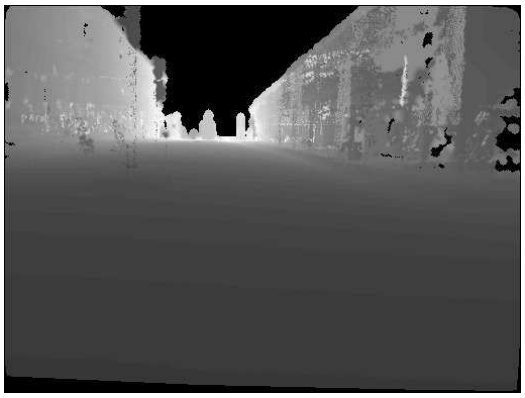
Due to the sparsity and sub-pixel values of the reprojections in the image, we

[1]Through experimentation, we found that a window of 40 m was sufficient to capture enough detail in the fair field.

(a) Camera image of the scene for reference.

(b) 3D scene prior coloured with laser reflectance values.

(c) Reprojected laser-intensity image at the estimated camera pose in the prior.



(d) Reprojected depth image, where lighter colours represent larger depths.

Figure 3.5: Illustration of generating a synthetic depth image. Using the estimated camera pose in the point cloud, all points within a local window are reprojected into the image plane. As these reprojections fall within sub-pixel values, bilinear interpolation is performed with neighbouring points, provided they are within a closeness threshold.

perform bilinear interpolation and then apply a median filter for smoothing. We only perform interpolations on pixels that are within a specified threshold of their reprojected neighbours.

3.3.3 Disparity-Based Distraction Suppression

As the vehicle to be localised has a stereo camera, we can, online, perform dense stereo to generate a live disparity image (Geiger et al., 2010). Using the background

depth image from 3.3.2, we can also generate a synthetic disparity image containing only the background by using the relationship between depth and disparity. Recall that our stereo equation is given by the following:

$$\mathbf{z}_c = \begin{bmatrix} u_l \\ v_l \\ d \end{bmatrix} = \frac{1}{z} \begin{bmatrix} x\ f_u + z\ c_u \\ y\ f_v + z\ c_v \\ f_u\ b \end{bmatrix}. \tag{3.3}$$

The third element of our stereo equation relates disparity to depth:

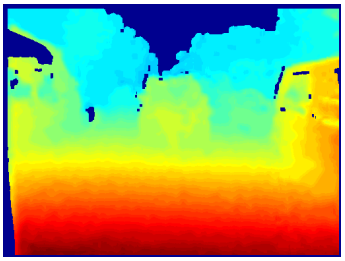
$$d = \frac{fb}{z}, \tag{3.4}$$

where $\{f, b\}$ are the intrinsic focal length and baseline. Thus, provided that the estimate of the camera pose used to generate the synthetic prior is reasonably accurate (e.g., sub meters in translation), any discrepancies between the real and synthetic disparity images represent ephemeral objects in the live stream (see Figures 3.6(a), 3.6(b), and 3.6(c)).

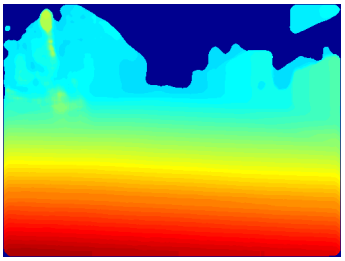
Although it is tempting to simply take the difference between the disparity images, there are two problems with this approach. Firstly, we note that calibration and localisation errors can lead to large disagreements in the foreground because of the inverse relationship between depth and disparity (i.e., noise on smaller depth values will produce large noise in disparity; see Figure 3.6(d)). Secondly, disparity differences for distant objects will naturally be smaller, meaning that we need some way of amplifying these weaker signals. By accounting for the uncertainties in generating the synthetic depth images, it turns out, that we are able to address both of these issues.

We therefore take a probabilistic approach and weight the disparity differences by their associated measurement uncertainties. For every pixel, i, in the image,

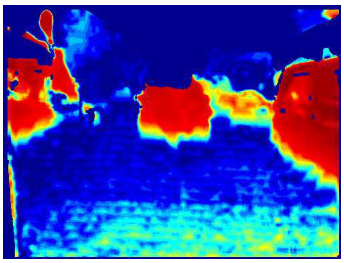
(a) Camera image for reference.



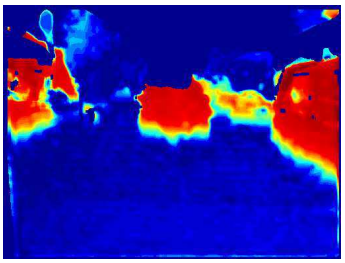
(b) Stereo disparity image using the method of Geiger et al. Geiger et al. (2010).



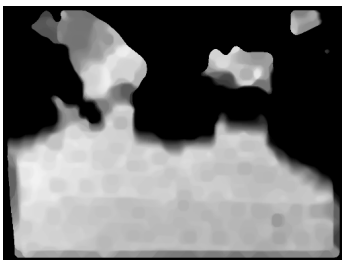
(c) Synthetic disparity image generated from the 3D scene prior.



(d) Disparity difference image (i.e., the absolute value of the difference between 3.6(b) and 3.6(c)).



(e) Uncertainty-weighted disparity difference image. Note that the signals in the near field that are present in Figure 3.6(d) are significantly down weighted when taking the uncertainty into account.



(f) Background-likelihood image, where black represents a likelihood of 0 and white represents a likelihood of 1. This image is used to weight the feature detection scores in the front-end of our VO pipeline.

Figure 3.6: Generating a disparity-based background-likelihood image. Beginning at the top, 3.6(b) shows the true disparity image captured from a live video stream, while 3.6(c) shows the synthetic disparity image generated by using the 3D scene prior (see Figure 3.5). Since this scene prior is absent of dynamic objects, there is a clear visual dissimilarity between the true and synthetic disparity images. Taking the difference of these images and weighting by the uncertainties resulting from localisation errors, we obtain a clean segmentation of the foreground and background elements 3.6(e), allowing us to create a background-likelihood image 3.6(f) to be used in our VO pipeline.

we define a disparity measurement from the dense-stereo, d_i^c, and synthetic depth image, d_i^s, as follows:

$$d_i^c := \bar{d}_i^c + \delta d_i^c, \qquad \delta d_i^c \sim \mathcal{N}(0, \sigma_{d_i^c}^2), \tag{3.5}$$

$$d_i^s := \frac{fb}{z\left(\mathbf{T}(\delta\mathbf{x})\bar{\mathbf{T}}_{c,m}\boldsymbol{p}_m^i\right)}, \qquad \delta\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_x), \tag{3.6}$$

where δd_i^c is normally distributed pixel noise with standard deviation $\sigma_{d_i^c}^2$. Recall from Section 2.1.3 that we cannot write $\bar{\mathbf{x}} + \delta\mathbf{x}$ if the orientation is expressed as a rotation matrix. Thus, we model noise on the current estimate, $\bar{\mathbf{T}}_{c,m}$, as a delta transformation matrix, $\mathbf{T}(\delta\mathbf{x})$, such that $\mathbf{T}_{c,m} = \mathbf{T}(\delta\mathbf{x})\bar{\mathbf{T}}_{c,m}$. Dropping the pixel subscript for convenience, we now define a disparity difference measurement as,

$$e_d := d^c - d^s \approx \underbrace{\bar{d}^c - \bar{d}^s}_{=:\bar{e}_d} + \underbrace{\delta d^c + \frac{fb}{(\bar{z}^s)^2}\left(\frac{\partial z^s}{\partial \mathbf{x}}\right)\delta\mathbf{x}}_{=:\delta e_d}, \tag{3.7}$$

where $\bar{z}^s := z(\bar{\mathbf{T}}_{c,m}\boldsymbol{p}_m^i)$, $\bar{d}^s := fb/\bar{z}^s$, and we have performed a first-order Taylor series expansion on the inverse depth term. The associated measurement noise is given by the following,

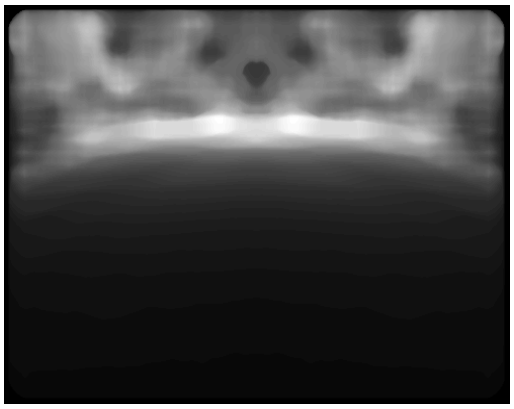
$$\begin{aligned} \sigma_{e_d}^2 &:= \mathrm{E}(\delta e_d \delta e_d^T) && (3.8)\\ &= \sigma_{d^c}^2 + \frac{(fb)^2}{(\bar{z}^s)^4}\left(\frac{\partial z^s}{\partial \mathbf{x}}\right)\mathbf{P}_x\left(\frac{\partial z^s}{\partial \mathbf{x}}\right)^T. && (3.9) \end{aligned}$$

Note that the Jacobian, $\partial z^s/\partial\mathbf{x}$, represents the change in depth that occurs given small perturbations of the vehicle's pose. At present, we have no efficient means of computing this quantity as numerical techniques are too slow. As such, we use the following approximation. To begin, let us define,

$$Z_x := \sqrt{\left(\frac{\partial z^s}{\partial \mathbf{x}}\right)\mathbf{P}_x\left(\frac{\partial z^s}{\partial \mathbf{x}}\right)^T}, \tag{3.10}$$

which provides an estimate of the depth change at a particular pixel location, given the localisation uncertainty. Figure 3.7(a) shows an example image where Z_x has been numerically computed for each pixel location. Examining this image, it becomes clear that the regions with the most uncertainty occur at large depths (due to the oblique angle between the plane and the optical axis), as well as non-smooth

(a) Representative Jacobian image given by Equation (3.10) (i.e., evaluating $\partial z^s/\partial \mathbf{x}$ for each pixel). Camera image provided for reference.



(b) An average depth-Jacobian image produced by averaging over 500 depth-Jacobian images.

Figure 3.7: An illustration of a representative depth-Jacobian image 3.7(a) and the average depth-Jacobian image 3.7(b). Light colours represent larger sensitivities to pose changes.

surfaces (e.g., trees). To approximate this Jacobian, we precomputed an *average depth-Jacobian image* by averaging over 500 images from a separate dataset. This depth-Jacobian image is shown in Figure 3.7(b). It should be noted that this approximation works well because we are operating in urban environments, where the structure of the scene remains relatively constant. Denoting this approximation as $\hat{Z}_x$, we have

$$\sigma_{e_d}^2 \approx \sigma_{d^c}^2 + \frac{(fb)^2}{(\bar{z}^s)^4}\hat{Z}_x^2, \tag{3.11}$$

allowing us to define our Mahalanobis distance as,

$$\tilde{e}_d := \sqrt{e_d^2/2\sigma_{e_d}^2}. \tag{3.12}$$

In Chapter 6 we revisit the Jacobian approximation (3.10) and introduce a different approximation that is more suitable for realtime operation and is not based on averaging.

Figure 3.6(e) shows the result of applying our measurement uncertainty to obtain

the uncertainty-weighted disparity difference. The effect is that errors in the near field are down-weighted, which naturally brings out differences with objects that are farther away (i.e., the weaker signals for distant objects appear stronger since the foreground noise is reduced). The background-likelihood image is then obtained by

1. Thresholding the uncertainty-weighted disparity (i.e., set $\tilde{e}_d > \tau_d = \tau_d$ for all pixels).

2. Using a max-filter to amplify the disparity disagreements. This helps reduce fluctuations in the mask by enforcing that a pixel takes on the max value in its local neighbourhood.

3. Scaling the image between $[0, 1]$ and taking the complement (see Figure 3.6(f)).

3.3.4 Flow-Based Distraction Suppression

This section presents an alternative method for generating a background-likelihood image, which relies on optical flow instead of dense stereo, making it applicable to monocular-based systems. To create a synthetic optical flow image at time t_k, the synthetic depth image and camera image at t_{k-1} are used to create a coloured point cloud. The motion estimate between times t_{k-1} and t_k, is applied and the coloured point cloud is reprojected into the estimated camera pose at time t_k to create a synthetic camera image[2] (see Figure 3.8). Regions without any data (i.e., pixel locations where the nearest reprojected point is beyond a certain distance) are filled in with the intensity values from the true camera image. This is necessary in order to ensure that we can create a full image without missing data, otherwise the optical flow algorithm will produce an extremely noisy result. After reprojecting the coloured point cloud and filling in missing regions, we apply bilinear interpolation,

[2]Note that we use VO for the motion estimate in our experiments, but wheel odometry could be used instead to make it a truly monocular system.

(a) Camera image of the scene for reference. Note that all vehicles in this scene are in motion.

(b) Synthetic camera image generated by reprojecting the coloured point cloud into the image plane. Large residuals with the true camera image (see left) are highlighted in red.

Figure 3.8: Illustrating the generation of a synthetic camera image based on prior 3D structure. The motion estimate between time t_{k-1} and t_k is applied and the coloured points are reprojected into the current camera frame. In this example, the camera hardly moved between frames, meaning that most points reprojected in roughly the same place in the image. However, as the vehicle on the right was actually in motion, there is a large discrepancy between the synthetic camera image and the true image.

followed by a Gaussian low-pass filter to smooth the image.

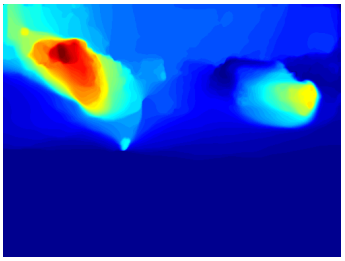
Once we have generated a synthetic intensity image at time t_k, we use the method of Liu (2009) to compute the expected optical flow (i.e., between the true image at t_{k-1} and the synthetic image at t_k) and the true optical flow (i.e., between the true image at t_{k-1} and the true image at t_k). See Figure 3.9(b) and 3.9(c) for an example. We define the true optical flow measurement, f^c, and synthetic optical flow measurement, f^s, for pixel i as,

$$f_i^c := \bar{f}_i^c + \delta f_i^c, \qquad \delta f^c \sim \mathcal{N}(0, \sigma_{f_i^c}^2), \tag{3.13}$$

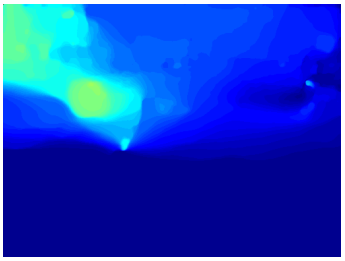
$$f_i^s := f_i^s \left(z^s \left(\mathbf{T}(\delta\mathbf{x}) \bar{\mathbf{T}}_{c,m} \boldsymbol{p}_m^i \right) \right), \qquad \delta\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_x). \tag{3.14}$$

In a similar fashion as before, and dropping the subscript, we define a difference

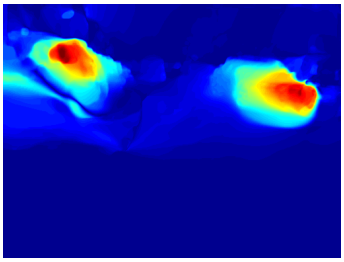
(a) Camera image for reference.



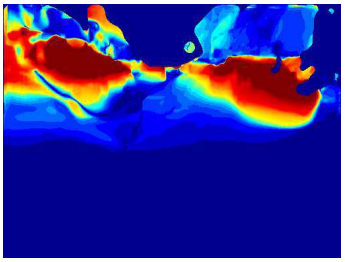
(b) True optical flow.



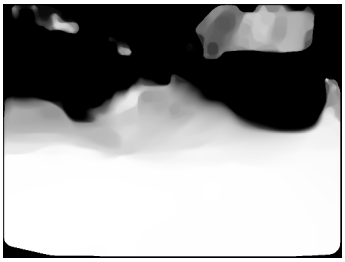
(c) Synthetic optical flow using the 3D prior.



(d) Optical flow difference.



(e) Depth-weighted optical flow difference.



(f) Background-likelihood image.

Figure 3.9: Generating a flow-based background-likelihood image. Beginning at the top, 3.9(b) shows the true optical flow from a live video stream, while 3.9(c) shows the synthetic optical flow generated by using the 3D scene prior (see Figure 3.8). Since this scene prior is absent of dynamic objects, there is a clear visual dissimilarity between the true and synthetic flow fields. Taking the difference of these images and weighting by the synthetic depth, we obtain a clean segmentation of the foreground and background elements 3.9(e), allowing us to create a background-likelihood image 3.9(f) to be used in our VO pipeline.

measurement and its associated uncertainty as,

$$e_f := f^c - f^s \approx \underbrace{\bar{f}^c - \bar{f}^l}_{=:\bar{e}_f} + \underbrace{\delta f^c - \frac{\partial f^s}{\partial z^s}\left(\frac{\partial z^s}{\partial \mathbf{x}}\right)\delta\mathbf{x}}_{=:\delta e_f}, \tag{3.15}$$

$$\sigma^2_{e_f} := \sigma^2_{f^c} + \left(\frac{\partial f^s}{\partial z^s}\right)^2 \left(\frac{\partial z^s}{\partial \mathbf{x}}\right) \mathbf{P}_x \left(\frac{\partial z^s}{\partial \mathbf{x}}\right)^T. \tag{3.16}$$

Unfortunately, this derivation introduces another Jacobian term, $\partial f^s / \partial z^s$, which represents changes in optical flow due to changes in depth. This Jacobian term is far from smooth and we have no clear means of computing it at present; it involves reprojecting coloured points, interpolating a grayscale image, and running it through

an optical flow algorithm that computes local spatial and temporal derivatives. We therefore adopt an alternative solution based on the intuition that scaling 2D flow fields by their associated depth approximates the 3D velocity (Taludker et al., 2003). In our case, we scale the difference between the expected and observed flow by the expected depth to amplify large differences:

$$\tilde{e}_f := e_f z^s. \tag{3.17}$$

Although this approach is slightly more crude in that we are not explicitly accounting for uncertainties in the flow difference, we found this to work well in practice. Figures 3.9(e) shows the depth-weighted flow difference and Figure 3.9(f) shows the resulting background-likelihood image, which is formed in the same manner as described earlier. The next subsection will discuss how we use these background-likelihood images in our front-end VO pipeline.

3.3.5 Feature Score Reweighting

Recall from Section 2.2, for feature extraction in our VO front-end, we use the FAST corner detector (Rosten et al., 2005) with a low threshold to obtain thousands of candidate features. As it would be intractable to perform feature matching on all of these candidates, our system takes the top N features, ranked by their corner score, s_i. In order to ensure that the features are well distributed spatially, the image is partitioned into a number of quadrants and the desired number of features, N, is divided equally among each quadrant.

The background-likelihood images are then used to re-weight each corner score by looking up the closest likelihood weight, b_i, and re-weighting according to the

following

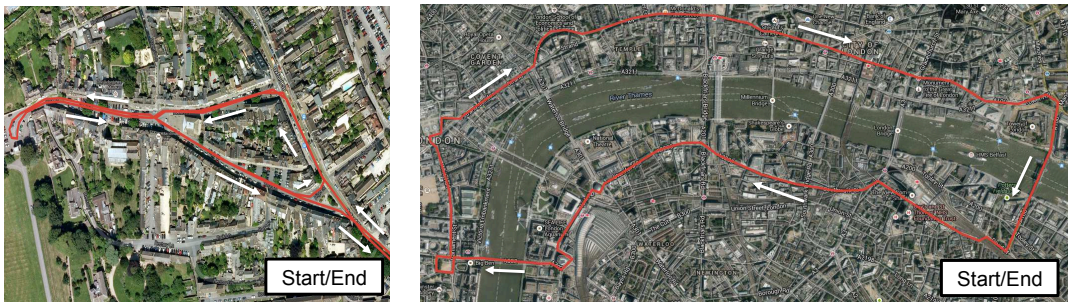
$$\tilde{s}_i = \begin{cases} 0 & \text{if } b_i < \tau_b \\ b_i s_i & \text{otherwise.} \end{cases} \tag{3.18}$$

where τ_b is a threshold for the minimum required likelihood.

3.4 Experiments and Results

We present experimental results from two different urban areas in the UK: Woodstock and London. The Woodstock datasets were collected with our Bowler Wildcat mobile platform, equipped with a Bumblebee 2 stereo camera, a SICK LMS-151, and an Oxford Technical Solutions (OxTS) RT-3042 INS for groundtruth (see Figure 3.11(a)). Our London dataset was gathered with a self-contained, vehicle-mounted mobile sensing suite, equipped with a Bumblebee 2 stereo camera, a SICK LMS-151, and a Trimble R8 GPS for groundtruth (see Figure 3.11(b)). The 3D priors were generated from the SICK lasers with VO for pose estimation; however, we wish to stress that the 3D prior could have been generated with a more sophisticated lidar sensor, such as the Velodyne. In addition, we use the 3D object detection and classification method of Wang et al. (2012) to postprocess our prior maps and remove most of the dynamic objects in the scene, which include pedestrians, vehicles, and cyclists.

To localise against the scene prior, we used the stereo localisation system described in Section 2.2. Lastly, Table 3.1 provides the system parameters used in these experiments, corresponding to the notation introduced in the previous section.

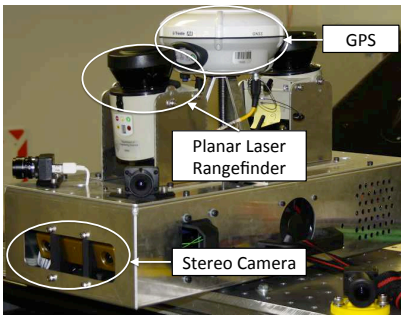


(a) Woodstock dataset collection route (approx. 2 km).

(b) London dataset collection route (approx. 10 km).

Figure 3.10: Dataset collection routes.

(a) The Bowler Wildcat platform, equipped with a wide array of sensors, including stereo cameras, monocular camera, 2D and 3D laser rangefinders and an INS. In our experiments, only the forward facing stereo camera, 2D laser, and INS were used.



(b) Our mobile sensing suite that was mounted to a commercial vehicle and driven around central London for data acquisition. Equipped with various sensors, only the stereo camera, planar laser rangefinder, and GPS were used in our experiments.

Figure 3.11: Our Bowler Wildcat dataset platform (left) and a mobile sensing suite (right). The Wildcat was used for dataset collection around Woodstock (see Figure 3.10(a)) and the mobile sensing suite was use for our dataset collection around London (see Figure 3.10(b)).

Table 3.1: System Parameters

Parameter	Description	Value
$\sigma^2_{d^c}$	Stereo disparity noise covariance [pixels^2]	0.05^2
τ_d	Mahalanobis distance threshold for disparity-based method	1
τ_f	Depth-adjusted error threshold for flow-based method	20

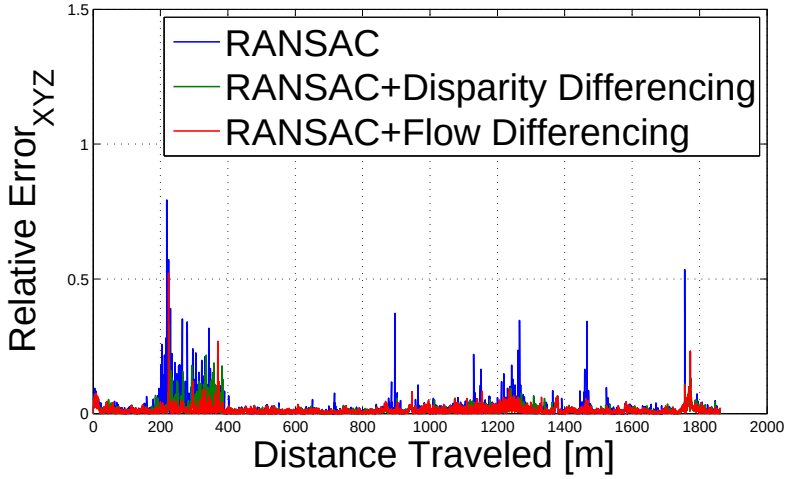


Figure 3.12: Relative frame-to-frame error (according to INS) of our standard VO system vs. the two distraction-suppression methods (Woodstock results). Spikes indicate differences in the frame-to-frame estimate when compared with INS. Note that we are, on average, always outperforming our baseline system and that both distraction-suppression techniques perform comparably. Representative cases where we outperform the baseline are shown in Figure 3.13.

3.4.1 Visual Odometry

The goal of this section is to illustrate the improvements to our VO system by incorporating background-likelihood images, as described in Section 3.3.5. We present results from Woodstock and central London during the Olympics (see Figure 3.10 for an illustration of the routes driven).

3.4.1.1 Woodstock

Two separate datasets were collected from the Begbroke Science Park to the town of Woodstock. A subset of the busiest sections of these datasets were processed, totalling approximately 2 km. As our localisation system would sometimes fail, we only present results on sections with a successful localisation, which is approximately 1.5 km.

To compute localisation error, we measure the difference between the estimated frame-to-frame pose changes and the INS measured pose change. This is a more

appropriate measure than looking at cumulative errors since a orientation error in one frame can skew the results for the rest of the trajectory. Denoting the true frame-to-frame translation as $\boldsymbol{\rho}_t$ and the estimated as $\boldsymbol{\rho}_e$, we define a frame-to-frame error measure as,

$$\mathrm{E}_{xyz} := \mid \; ||\boldsymbol{\rho}_e||_2 - ||\boldsymbol{\rho}_t||_2 \; \mid . \tag{3.19}$$

We computed this error measure for three implementations: (i) our standard VO system using RANSAC, (ii) our disparity-based method with RANSAC, and (iii) our flow-based method with RANSAC. To reiterate, the method presented in this chapter provides an extra step of outlier rejection before proceeding with RANSAC, which is why we still require RANSAC in our pipeline. The goal is to illustrate the improvements in VO by incorporating these likelihood images for feature reweighing. Figure 3.12 shows the error percentages for our disparity-based and flow-based distraction suppression techniques against our standard VO system, where we see a noticeable improvement in accuracy.

A number of representative cases where our methods outperform the baseline are shown in Figure 3.13 and occur when there are many strong candidate feature matches on moving vehicles. Although one may argue that motion segmentations systems could potentially resolve some of these issues, we note that there are several cases where most of the scene was initially static but began moving (e.g., pulling up to traffic stopped at a red light). The strength of our technique is that regardless of how much of the image is obscured, we are able to focus our attention on just the portions of the image that belong to the static background.

3.4.1.2 London

For our London datasets, we collected three 10 km loops around several landmarks sites, such as the Houses of Parliament, Trafalgar Square, and St. Paul's Cathedral.

For these experiments, signal-strength issues resulted in poor GPS measurements, which are not accurate enough to groundruth our motion estimates. We note that this is in fact a common problem in urban environments, strengthening the case that improving the robustness of relative motion estimation is a vital pursuit. Owing to this lack of groundtruth, we present qualitative evidence of our algorithms working in situations with extreme scene motion[3] (see Figure 3.14).

3.5 Summary

This chapter presented a technique to suppress the influence of dynamic objects on our vision system, which allows for robust egomotion estimation in heavily cluttered environments. This is an important competency to have for any vision-based system that aims at operating outdoors in urban environments.

Referring back to our earlier discussion on sources of visual change, we next address the problem of how to cope with illumination changes throughout the day. More specifically, the next chapter will examine the problem of navigating in shadowy regions, which can drastically change the appearance of the scene.

[3]Video results of these techniques can be viewed at `http://www.youtube.com/watch?v=7ie9fNvcDC4`. Note that the minor fluctuations in the distraction masks are due to the noise in the dense stereo algorithm.

Figure 3.13: Results from our Woodstock dataset. The top two rows showcase examples where we drove behind a vehicle that was initially at rest, but then began to move. As the vehicle makes up a large portion of the image and has distinctive features, our baseline system matched features on the vehicle across subsequent frames, leading to an erroneous motion estimate. In contrast, our distraction suppression systems ignored this vehicle and produce an accurate estimate. The last row shows a situation where RANSAC yielded a poor initial guess and the baseline system converged to an inaccurate estimate. Once again, this was not an issue with our distraction suppression methods, which can easily distinguish the foreground and background objects.

Figure 3.14: Results from our London dataset. The top row illustrates an example of a large bus obscuring the image and very slowly approaching as our vehicle began to move. Our baseline system tracked features on the bus instead of the road surface, leading to an incorrect motion estimate. We wish to stress that even though most of this image is obscured by foreground objects, our distraction suppression techniques are able to focus on the static parts of the scene, resulting in more robust estimates. The bottom two rows illustrate other examples of our baseline system (i.e., without distraction suppression) incorrectly tracking features on moving vehicles and producing erroneous estimates.

Chapter 4

Illumination Invariance

4.1 Introduction

As stated in Section 2.1, at a high level, the localisation task can be divided into two main components: (i) the visual front-end responsible for matching features from the live view to the map, and (ii) the estimation back-end responsible for computing the pose of the vehicle. While the matching problem is simply stated, its execution remains extremely challenging when faced with appearance changes caused by varying environmental conditions.

In this chapter, we examine the effects of lighting changes caused by the Sun. The primary challenge presented by lighting changes are the shadows they cast, which can obscure features in the environment and create new ones from the silhouettes, making it difficult to match features from a sunny day to a cloudy day (see Figure 5.1). To address this problem, we leverage recent work in the computer vision field for transforming RGB-coloured images into an illumination-invariant colour space (Ratnasingam and McGinnity, 2012). The ability to determine the colour of objects irrespective of an external illumination source is known as colour constancy (Ebner, 2007).

We present an approach that runs two localisation threads in parallel, one using

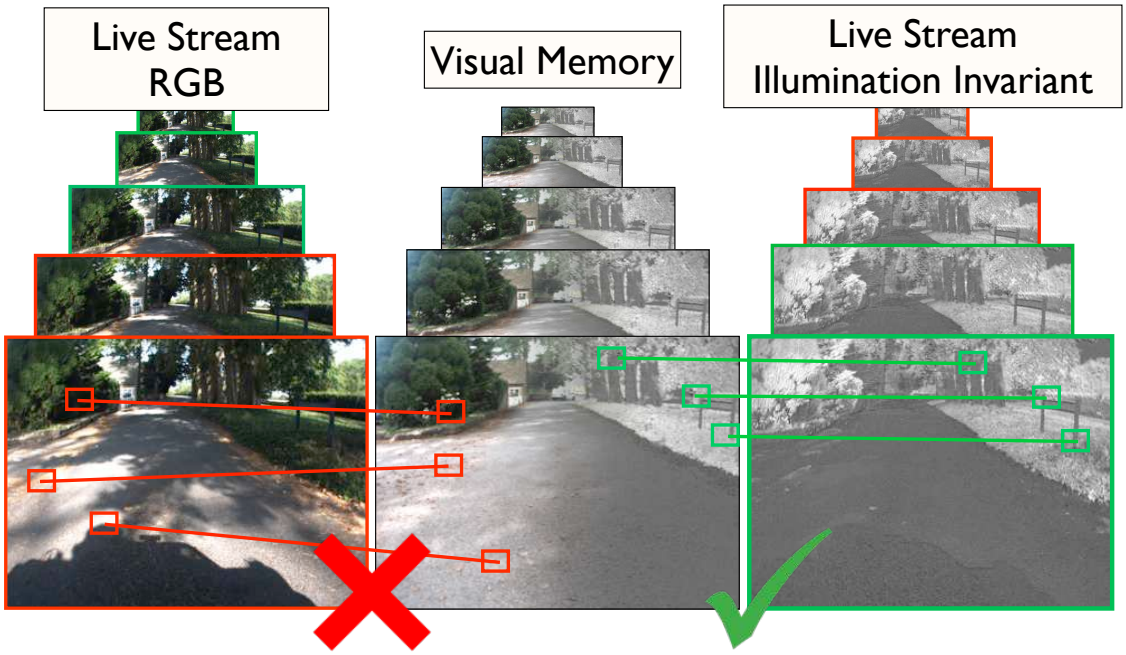


Figure 4.1: We present an approach to localisation that runs two localisers in parallel—one using images in RGB colour space and the other using images in an illumination-invariant colour space. This allows us to cope with areas that exhibit a significant amount of lighting variation. The top left image sequence represents the live video stream in RGB colour space, the right image sequence represents the live video stream in an illumination-invariant colour space, and the middle sequence represents the images in our visual memory, which include both the RGB and illumination-invariant space. In our technique, if one of the localisers fails, we switch to the other.

the original RGB images and the other using the illumination-invariant images. The system switches between the two estimates depending on the quality of the respective estimates. We demonstrate on over 10 km of data that this remarkably simple addition to the standard vision pipeline can result in significant improvements in areas that exhibit a great deal of lighting variation. In later chapters, after introducing place-dependent feature detectors for localisation, we will fold in this illumination-invariant colour space and analyse the combined system performance.

4.2 Related Work

Shadows, insufficient lighting, and changing lighting conditions have been studied in a variety of fields with different goals in mind. In this work we draw on the optics community who have paid careful attention in modelling the image formation process, considering properties of the illuminant, the camera, and the scene. Of particular relevance to this work, the optics literature shows how full colour images can be mapped to an illumination-invariant space. Finlayson et al. (2006, 2004) present a method that learns the illumination-invariant mapping by analysing images under different lighting conditions (e.g., the ground in sun and shade). Ratnasingam and McGinnity (2012); Ratnasingam and Collins (2010) instead use known properties of the camera to produce the invariant image.

In the computer vision community, the detection and removal of shadows has been performed using learnt classifiers. Guo et al. (2011) use a graph-cut framework involving image patches to remove shadows from natural scenes. Zhu et al. (2010) are able to classify shadows in greyscale images using boosting and conditional random fields. Kwatra et al. (2012) use an information theoretic method—a hybrid of the classifier and physics based approaches—to remove shadows in aerial imagery. While the results are effective, the process is relatively slow for typical image sizes.

Within the robotics community, the issues of lighting in different problems have been tackled in a variety of ways. The SeqSLAM (Milford and Wyeth, 2012) algorithm is able to achieve successful topological localisation despite extreme variations in lighting. The approach exploits the fact that sufficiently long sequences of images are distinctive enough for localisation, and they are able to localise at night against a daytime map. Corke et al. (2013) apply Finlayson's invariant image to the problem of single-image localisation to deal with the issue of shadows. They show that the transformed images of a location were more similar than the original colour

images and therefore localisation performance improved. Maddern and Vidas (2012) show that place recognition can be improved over a day-night cycle by using both a standard and thermal camera; however this required specialist hardware. McManus et al. (2012) improve the robustness of their visual teach and repeat system to lighting issues by using a lidar-based sensor, which is lighting invariant. While the sensor produces good results, it has a range of issues including cost, fragility, power requirements, frame-rate (2 Hz) and availability. The experience-based navigation system by Churchill and Newman (2012) attempts to solve the lighting problem by capturing the different visual modes of an environment with different experiences. However this was found to break down when lighting effects cause new visual patterns on every visit to a location, such as shadows cast by foliage. In this work we look to leverage the invariant image transform proposed by Ratnasingam and McGinnity (2012) to improve metric localisation performance and robustness in the face of strong and changing shadows.

4.3 System Overview

A robust localisation system should be able to answer the following questions at all times: "where should I look (spatially in the image)" and "what should I look for (appearance in the image)?" We address both of these questions in coming sections and then present our approach to combining lighting-invariant images with our baseline system.

4.3.1 Knowing Where To Look

In an effort to improve robustness in matching a live view with a survey view, we take an active searching approach similar to Davison et al. (2007), which predicts how the measurements in the survey frame should reproject in the live frame. In

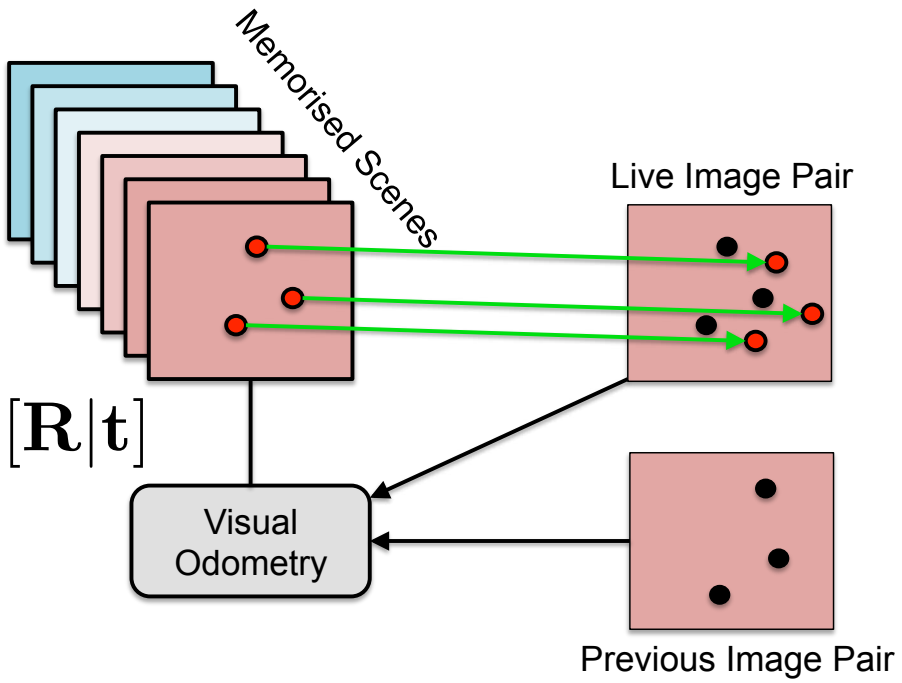


Figure 4.2: Illustration of our feature prediction approach. Using the latest VO output from the live image stream, we can predict where features in the live frame should reproject in the survey keyframe. This allows us to restrict the search space for candidate feature matches, which has two benefits: (i) improves efficiency, and (ii) reduces spurious matches that result from global matching in descriptor space.

Davison's work, they were limited to predicting the motion using a constant-velocity motion assumption. In our localisation system, however, we have access to the VO output from the live image stream. This allows us to accurately predict how we have moved relative to our survey and therefore inform where we expect to find stored features in the live view. This in turn can be used to define a search region in the live view (see Figure 4.2).

More specifically, consider a landmark defined in the map frame, $\mathbf{p}_m^j$. Using the previous localisation estimate at time t_{k-1}, denoted by $\mathbf{T}_{k-1,m}$, and the most recent relative transformation from the VO, $\mathbf{T}_{k,k-1}$, we can predict the location of the vehicle at the current time, t_k, according to $\hat{\mathbf{T}}_{k,m} = \mathbf{T}_{k,k-1}\mathbf{T}_{k-1,m}$. This allows us to use our stereo sensor model from Section 2.2, $\mathbf{y}_k = \mathbf{h}(\mathbf{T}_{k,m}\boldsymbol{p}_m^j)$, to compute the predicted location of the j^th landmark in the current camera frame. By using this

active search approach, we are able to better predict our search regions and thus, reduce the likelihood of bad data associations.

In addition to feature prediction, it is also worth noting that the motion prediction is useful for estimating the closest keyframe in the map, which allows us to cache the relevant subset of landmarks for localisation.

The next step, which is the key contribution of this chapter, is to identify *what* to look for within each of these regions. Standard methods would attempt patch-based matching or descriptor-based matching on the raw images. However, this approach is obviously inadequate under extreme lighting changes. In the next section, we will show how a simple image transformation can help improve localisation in areas with significant lighting variation.

4.3.2 Knowing What To Look For Whatever the Lighting

Given a search region for a potential match, our baseline system finds the sub-pixel location that minimises the score between the reference patch from the survey and the live image. However, as illustrated in Figure 5.1, this approach can fail when the appearance change is too significant. To remedy this problem, we wish to inform our system about the illuminate-free appearance of the scene, which requires a transformation from the standard RGB colour space.

4.3.2.1 Mapping to an Illumination-Invariant Chromacity Space

Ratnasingam and McGinnity (2012) presented a method for mapping three image sensor responses (e.g., RGB colour space) to an illumination-invariant chromacity space, $\mathcal{I}$. We begin by presenting the derivation and assumptions used in generating this illumination-invariant colour space, following the derivation presented by Ratnasingam and Collins (2010).

Figure 4.3 shows an illustration of the setup. The first assumption made is that

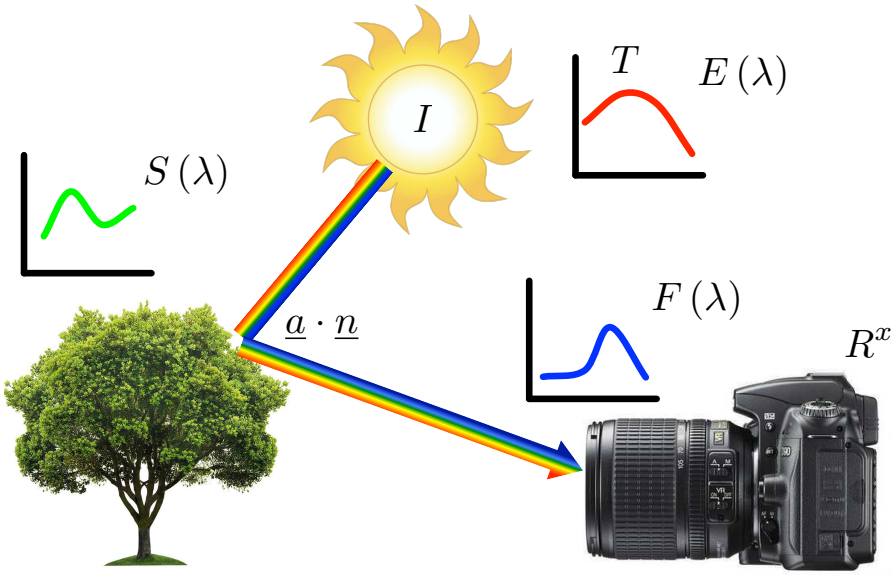


Figure 4.3: Simplified model of camera image responses under the assumption of a black-body radiator. In this setup, I is the intensity of the external iluminant, E, is the spectral power distribution of the illumination, which is only a function of temperature and is given by Planck's equation. S is the surface reflectance of the object being radiated, $\vec{a}$ is the direction of the light, $\vec{n}$ is the surface normal of the object and F is the spectral sensitivity of each image sensor. Image credit: Maddern et al. (2014a).

the external illuminant, the sun in this case, is a black body radiator. This means that the emitted radiation is only a function of temperature and is given by Planck's law. Referring to the figure, we define the following quantities:

I - the intensity of the external illuminant

$E(\lambda)$ - spectral power distribution of the illuminant, which is a function of temperature

$\vec{a}^x$ - direction of the light source imaging a point x on the surface

$\vec{n}^x$ - surface normal of the surface at some point x

$S(\lambda)^x$ - surface reflectance function

$F(\lambda)$ - spectral sensitivity of each image sensor

The response on each image sensor is then given by the following equation:

$$R_i^{x,E} = \vec{a}^x \cdot \vec{n}^x I \int S(\lambda)^x E(\lambda) F(\lambda) d\lambda \qquad i = \{R, G, B\}, \tag{4.1}$$

which follows a Lambertian reflectance model where the reflectance does not depend on the viewpoint of the camera. We now introduce a second assumption, which is that the spectral sensitivities of each image sensor, $F(\lambda_i)$, are infinitely narrow. In other words, modelling the spectral sensitivity as a Dirac delta centred at the peak spectral response for each sensor, $F_i = \delta(\lambda_i)$, we arrive at the following, were we have dropped the superscripts, $\{x, E\}$ for convenience:

$$R_i = \vec{a} \cdot \vec{n} I \underbrace{S(\lambda_i)}_{=:S_i} \underbrace{E(\lambda_i)}_{=:E_i}. \tag{4.2}$$

Taking the logarithm results in a linear combination of three components: (i) one dealing with scene geometry and intensity, (ii) a surface reflectance component, and (iii) an illuminant spectrum component:

$$\log(R_i) = \log(\vec{a} \cdot \vec{n} I) + \log(S_i) + \log(E_i). \tag{4.3}$$

The next simplification is to use Wein's approximation of a Planckian source, which gives us an expression for the spectral power distribution, E_i, leaving us with

$$\log(R_i) = \log(\vec{a} \cdot \vec{n} I) + \log(2hc^2 \lambda_i^{-5} S_i) - \frac{hc}{k_B T \lambda_i}, \tag{4.4}$$

where h is Planck's constant, c is the speed of light, k_B is Boltzmans constant, and T is temperature of the black body source.

Now we take a brief moment to consider what our goal is. We wish to use the colour information from each channel (4.4) and find a way of removing the

temperature and scene geometry dependence. If we can do this, then we have a representation that is independent of scene geometry and the external illuminant, and only dependent on the material.

Consider the following linear combination of the responses in each colour channel:

$$\begin{aligned} \mathcal{I} &:= \log(R_G) - \alpha \log(R_R) - \beta \log(R_B) && (4.5)\\ &= (1-\alpha-\beta)\left(\vec{a}\cdot\vec{n}I + \log(2hc^2)\right) + && (4.6)\\ &\quad \log(\lambda_G^{-5} S_G) - \alpha \log(\lambda_R^{-5} S_R) - \log(\lambda_B^{-5} S_B) - && (4.7)\\ &\quad \frac{hc}{k_B T}\left(\frac{1}{\lambda_G} - \frac{\alpha}{\lambda_R} - \frac{\beta}{\lambda_B}\right). && (4.8) \end{aligned}$$

If we impose the following constraints

$$\alpha + \beta = 1, \qquad \frac{1}{\lambda_G} = \frac{\alpha}{\lambda_R} + \frac{\beta}{\lambda_B}, \tag{4.9}$$

we can eliminate the geometry term and the external illuminate term as seen below:

$$\mathcal{I} = \log(\lambda_G^{-5} S_G) - \alpha \log(\lambda_R^{-5} S_R) - \log(\lambda_B^{-5} S_B). \tag{4.10}$$

Thus, we have arrived at an illumination-invariant transformation, which can be applied on a pixel-wise basis and is simply given by

$$\mathcal{I} = \log(R_G) - \alpha \log(R_R) - \beta \log(R_B), \tag{4.11}$$

where $\{R_i\}$ are the colour responses in each channel, $\{\alpha, \beta\}$ are channel coefficients given by (4.9), and $\{\lambda_R, \lambda_G, \lambda_B\}$ are the peak sensitivity wavelengths for each image sensor, which can be gathered from the sensor data sheet.

It is worth taking a moment to reflect on the fact that (4.11) involves just one line of Matlab code and is simply a weighted, linear combination of the logs of each

Figure 4.4: Example images taken around our Begbroke Science Park test site, with the raw RGB image shown on top, and the corresponding lighting invariant version shown below. Note how the image transformation is able to significantly reduce the impact of the shadows.

colour channel. This image transform comes with little-to-no cost, but provides a useful colour space for localisation in shadowy regions (see Figure 4.4 for some examples). Note, however, that this transformation adds noise, which is particularly prominent in the foreground on the road. The significance of this will be discussed in more detail in the next section.

4.3.2.2 Combined Localisation System

Transforming the live image stream using (4.11) can be performed on a per-pixel basis, and is therefore inexpensive, allowing us to run this thread alongside the baseline system. Our strategy for the combined system is quite simple. We run both VO streams in parallel and when we are able to localise using the raw images (i.e., the baseline system), we take that estimate, otherwise, we switch to the lighting-invariant estimate. The reason we default to the baseline system is highlighted in Figure 4.5, which shows a representative velocity profile both with and without using the illumination-invariant image transform. There are two main differences that can be observed. The first is that the illumination-invariant estimates are noisier, which

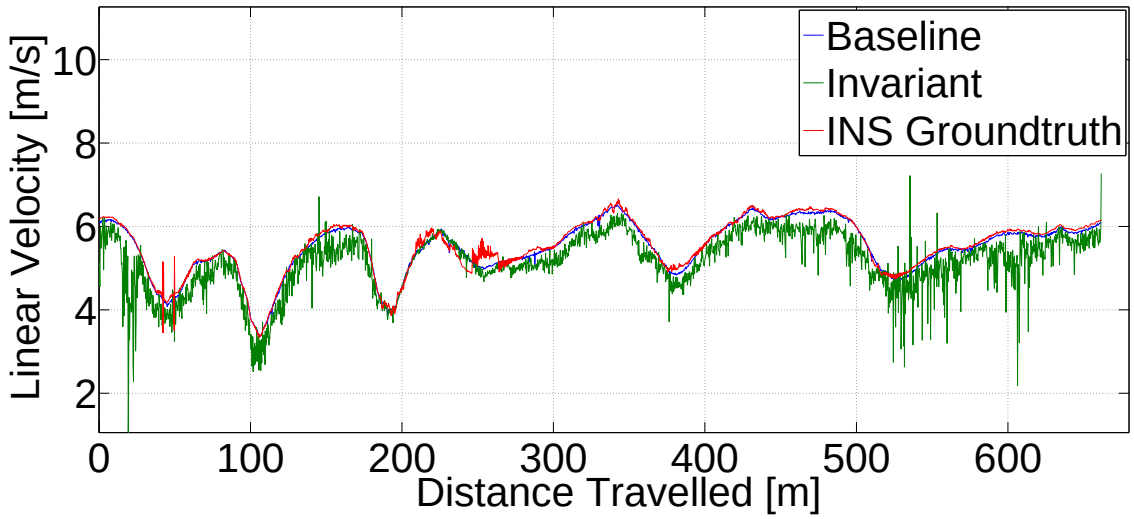
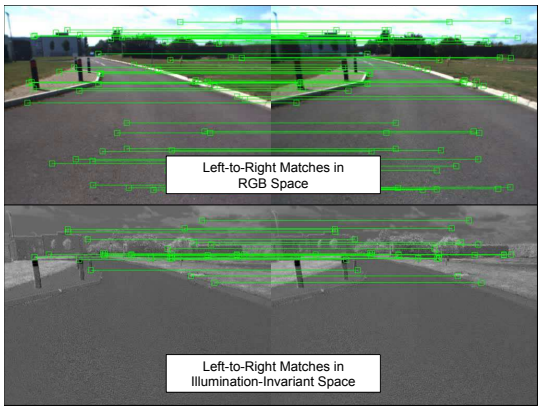
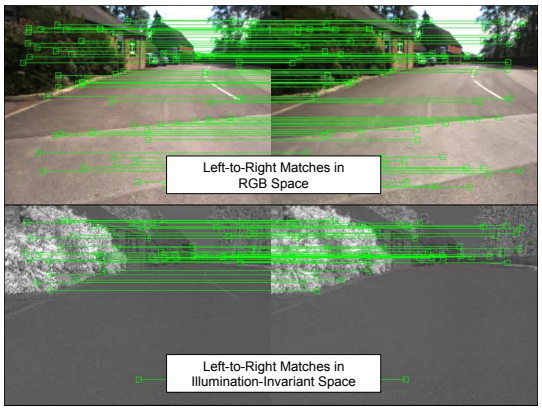


Figure 4.5: Representative velocity estimates for a loop around our Begbroke Science Park. Note how the estimates using the lighting invariant images are noisier and appear to have a slight bias when compared to groundtruth.

is likely due to the noise added by the pixel-wise transform. The second and more interesting difference is that there appears to be a slight bias in illumination-invariant estimates. We believe that this is a function of the feature distribution that results when using the illumination-invariant images. It appears that a lot of the high-frequency noise in the near field is amplified, meaning that fewer near-field features are detected. As a result, the feature distribution appears to be strongly biased towards the upper region, typically representing distant features (see Figure 4.6). As there exists a known bias in stereo (Sibley, 2007) (with a strong relationship to range), we believe this is the most likely explanation. Thus, owing to the increased noise and slight bias, fusing the estimates seemed suboptimal. Instead, we switch between the two systems, with the policy of defaulting to the baseline system when possible. A block-flow diagram of our system is provided in Figure 4.7.



Left-to-right stereo VO matches in an RGB and illumination-invariant colour space.



Left-to-right stereo VO matches in an RGB and illumination-invariant colour space.

Figure 4.6: Frame-to-frame VO matches for two locations using the RGB and illumination-invariant colour space. Each column is of the same place for comparison. Feature matches are shown from the left to the right camera. Note that the illumination-invariant VO matches are typically located near the top of the image on distance features. This feature distribution is likely to contribute to the velocity bias observed in Figure 4.5 due to the well-known range bias in the stereo model (Sibley, 2007).

4.4 Experiments and Results

In this section, we present a series of localisation results both with and without the use of lighting-invariant imagery. We collected 15 visual surveys around the Begbroke Science Park with the focus on capturing more challenging lighting conditions (see Figure 5.9 for a figure of the route). In Figure 4.9 we show some examples of the extreme visual variation encountered along parts of the route. To clarify terminology, the system that does not use invariant imagery (RGB only) is the *baseline system*, the system that uses invariant imagery only is the *invariant system*, and the system that combines them is the *combined system*. We used the same data gathering platform as outlined in Chapter 3, which includes a Point Grey Bumblebee2 camera, logging 512x384 colour image pairs at 20 Hz.

For each of the 15 datasets, we used an exhaustive leave-one-out approach, whereby each dataset was taken as the live image stream, and localisation was

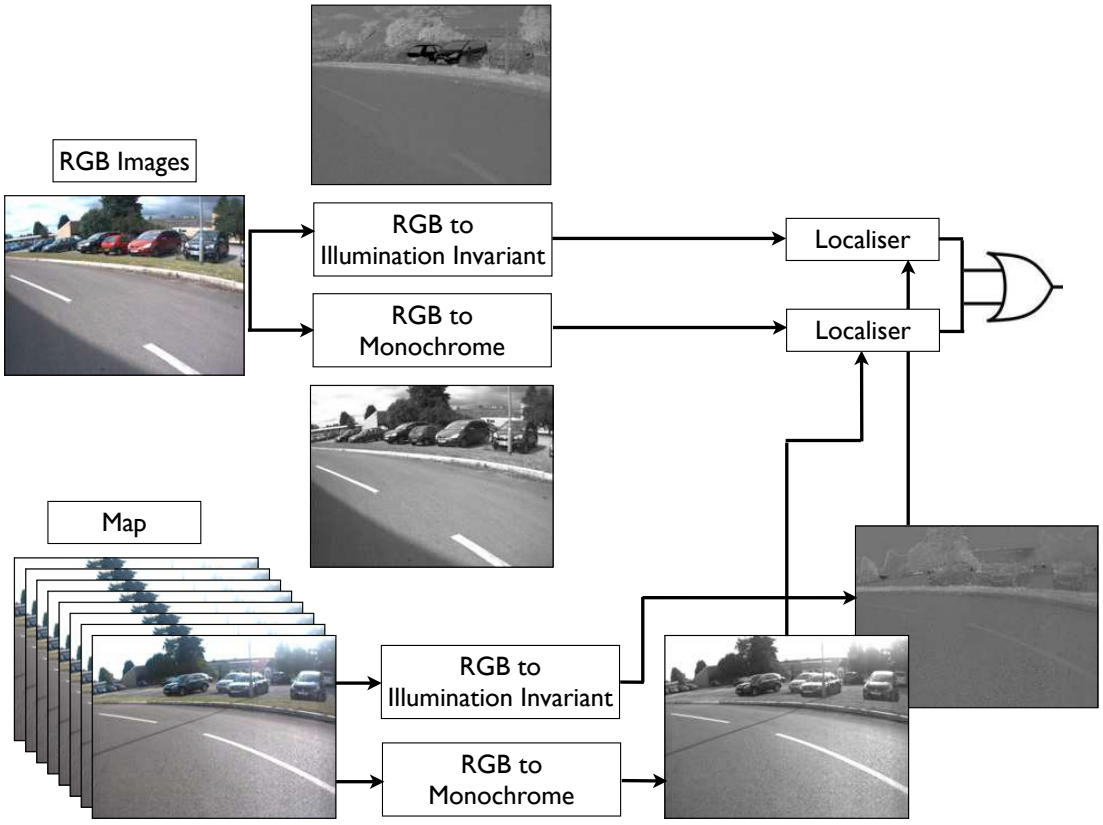


Figure 4.7: Block-flow diagram of our combined localisation approach. Note that our baseline system does not work on the raw RGB images, but actually transforms them to monochrome.

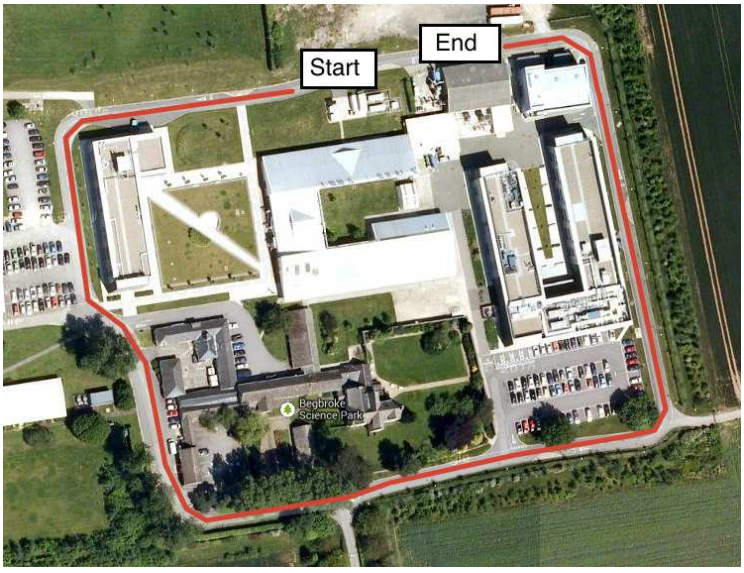


Figure 4.8: Our driven route around the Begbroke Science Park. The route is approximately 650 m in length.

Figure 4.9: Sample images gathered under a shadowy area in our Begbroke datasets. These areas prove to be very challenging for our baseline system due to the extreme variations in lighting.

performed against the remaining 14 datasets in turn.

Table 4.1 presents the percentage coverage using each of the 15 datasets as the live run. We define percentage coverage as the number of successfully localised frames versus the total number of frames, averaged over the 14 datasets compared against. We found that our INS system was not reliable for groundtruthing due to significant GPS drift (on the order of meters). Instead, we took the approach of Churchill and Newman (2012), which uses the localisation chain to predict the frame-to-frame motion and compares that with the VO estimate. If the two estimates disagree by a certain threshold then it is classified as a localisation failure. In other words, we only accept localisation estimates that are consistent with the motion of the vehicle (see Figure 4.10).

Table 4.1: Coverage results comparing our combined system versus the baseline system. Coverage is defined as the number of successfully localised frames as a fraction of the total number of captured frames, averaging over 14 training datasets per test dataset. Dataset numbers with low coverage results (e.g., 1 and 12) represent runs that are the most visually dissimilar to the others (e.g., the bottom left and bottom right images in Figure 4.9).

Dataset Number	Baseline System	Combined System
1	79.93%	83.19%
2	92.68%	95.74%
3	91.12%	94.59%
4	95.81%	96.65%
5	94.19%	95.80%
6	93.64%	95.74%
7	95.64%	98.30%
8	96.29%	97.60%
9	94.75%	97.30%
10	93.90%	95.61%
11	83.47%	89.35%
12	95.88%	97.54%
13	91.87%	95.01%
14	86.58%	89.55%
15	97.33%	98.53%
Average	**92.17**%	**94.68**%

In all cases the invariant system provides improvement to the baseline system, meaning the combined system *always* out-performs the baseline. An important result here is that our baseline system already performs well despite the difficult conditions. However, in the context of long-term autonomy for robotics, robustness is key, so any increase in reliability is important. We will show shortly that with the combined system we achieve significantly shorter distances travelling open loop during localisation failures.

Figure 4.11 shows the localisation performance of one live run versus one other dataset. In this figure, coloured points indicate a successful localisation for the specified system, while an absence of data represents a localisation failure. For this particular run, we see that the baseline system failed to localise over a 90 m section.

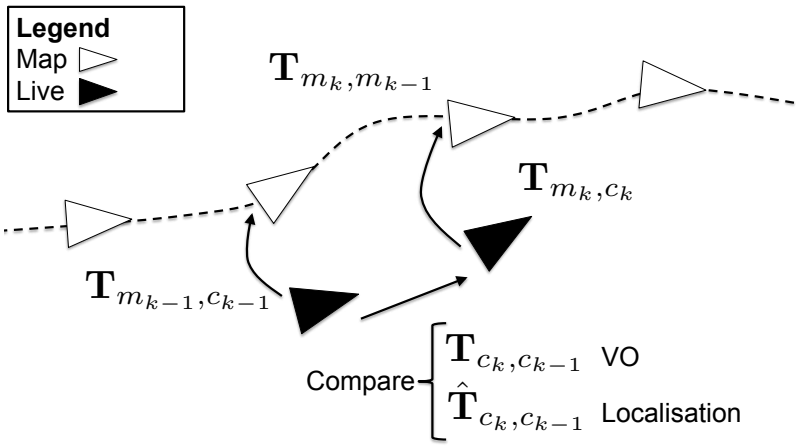


Figure 4.10: Illustration of how to measure the quality of localisation estimates by comparing with the motion estimate from VO. The white triangles represent kefyrames in the map and the black triangles represent the live poses. We can estimate the frame-to-frame motion from the localisation estimates by computing $\hat{\mathbf{T}}_{c_k,c_{k-1}} = \mathbf{T}^{-1}_{m_k,c_k}\mathbf{T}_{m_k,m_{k-1}}\mathbf{T}_{m_{k-1},c_{k-1}}$, which represents the motion of the camera from t_{k-1} to t_k.

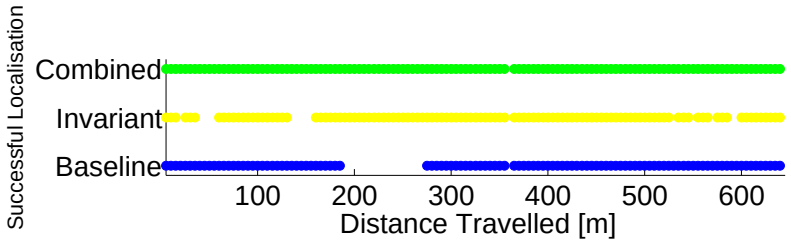


Figure 4.11: One vs. One localisation result. The localisation performance of the three systems (i.e., the baseline, the illumination invariant system, and combined). Points indicate successful localisation. Between 190 m and 280 m the invariant thread is able to localise where the baseline thread cannot. By taking the union of the two our combined system is more robust. Note that the region between 190 m and 280 m is the canopied region shown in Figure 4.9.

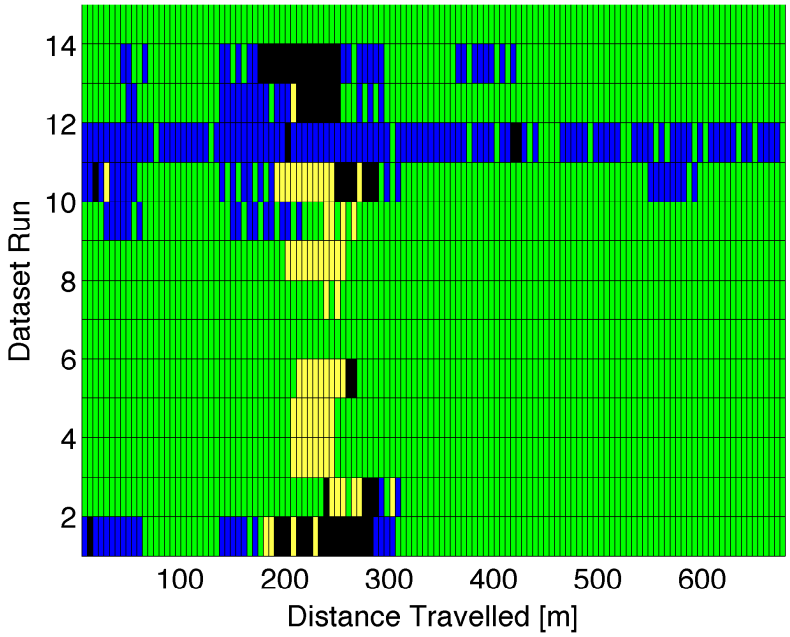
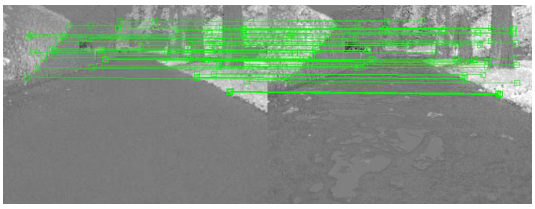


Figure 4.12: One vs. All localisation results. The localisation characteristics of a single dataset (used as the live image stream) compared against the remaining 14 datasets. Each row corresponds to one of the 14 datasets, and the x-axis shows distance travelled. Blue indicates when only the baseline system localised, yellow indicates when only the invariant system localised, green is when both the baseline and invariant successfully localised, and black areas indicate localisation failures of both systems. By incorporating the invariant system we are able to localise successfully over a larger area. Note that the region between 190 m and 280 m is the canopied region shown in Figure 4.9.

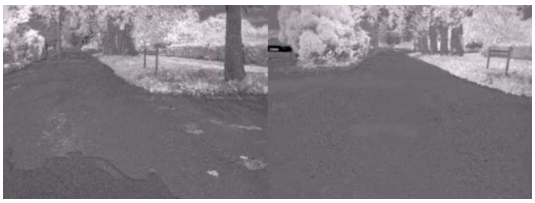


Successful localisation under the trees. Data associations shown in green.

Failed localisation under the trees. No successful matches.

Failed localisation near a car park. No successful matches.

Successful localisation near a car park. Data associations shown in green.



Failed localisation under the trees. No successful matches.

Failed localisation under the trees. No successful matches.

Figure 4.13: Examples where the lighting-invariant system helped (top row), where the RGB system helped (middle row), and where both failed (bottom row). As can be seen, the image transform adds artefacts, which can sometimes result in fewer matches. However, the benefit of running this system becomes clear when looking at regions with high visual variability caused by external illuminates.

However, because we have the invariant system running in parallel, which was able to localise in this area, the combined system is able to localise for almost all of the route. Figure 4.13 shows representative cases where the invariant localisation thread was successful while the baseline was not, and vice versa. The areas of failure for the baseline were typically around the 200 m mark under a canopied region, which resulted in significant shadowing from one dataset to the next (see Figure 4.9).

Figure 4.12 shows the performance of a single dataset used as the live image stream versus all 14 remaining datasets (along the y-axis). It is a graphical representation of one of the rows in Table 4.1. In this plot, yellow indicates regions where

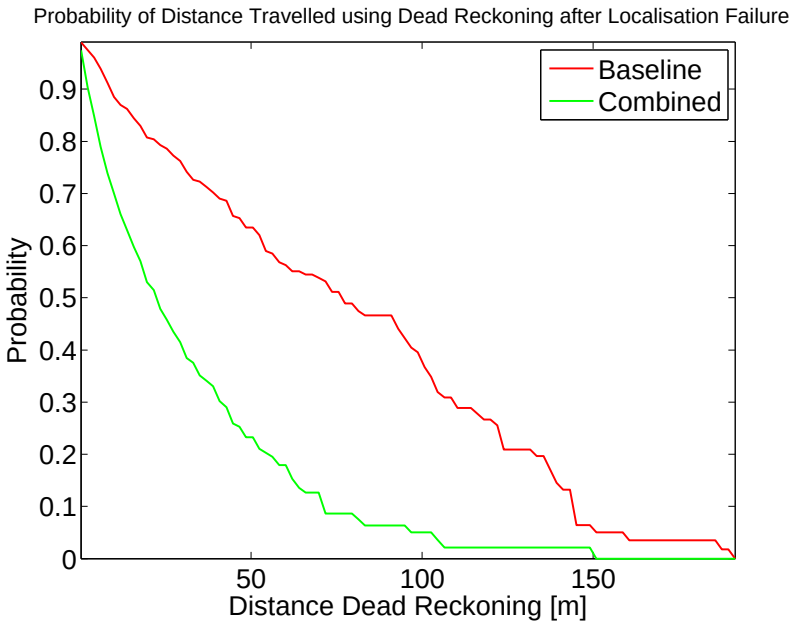


Figure 4.14: Given a localisation failure, this plot shows how far the system is likely to travel before re-acquiring a localisation, i.e. how long it will have to travel using dead reckoning alone. In other words, this is $P(\text{dropout} = X)$, where X is distance traveled. We see that the combined system is likely to travel significantly shorter distances compared to the baseline after a localisation failure.

only the invariant system could successfully localise. Here we see there is a region between 200-300 m along the route where the baseline thread repeatedly struggles, due to the challenging lighting variation (see Figure 4.9). It should also be noted that the invariant thread does not always contribute. The blue regions in Figure 4.12 indicates areas where only the baseline thread was successful. By taking the union of the two threads we have improved the robustness of our system.

4.4.1 An Alternative Metric for Performance

We refer the reader to Figure 4.14, which is the key result of this chapter and introduces a metric that will be used throughout the remainder of this thesis. Given that this is a localisation system, the primary concern is exposure to extended periods of time or travel in which we fail to localise. During these periods we must

fall back to dead reckoning from Visual Odometry—however good that is we are still effectively running "open loop".

Figure 4.14 shows that the system we propose here, which leverages illumination-invariant colour spaces, a dual-processing pipeline, and a carefully informed search policy for feature associations, produces a performance far superior to the baseline system. For example, the likelihood of the system travelling blind for up to 100 m is close to 40% with the baseline system, whereas with the combined system, the likelihood is just 5%.

Figure 4.14 presents a new perspective on how to judge the performance of a localisation system and allows us to ask some interesting questions. For instance, what is more desirable: a system that fails to localise frequently, but only over short distances, or a system that fails infrequently, but does so over long distances? The answer is obvious. For a localisation system, we would prefer the one that fails frequently with short distance intervals between the failures. As this is an important property for a localisation system, we use this metric in the subsequent sections.

4.5 Summary

This chapter showed how an illumination-invariant colour space can be incorporated into a visual localisation pipeline to significantly improve robustness in shadow-stricken environments. We presented an in-depth experimental analysis on our combined system and introduced a performance metric that considers distance traveled in between localisation failures. This is an important property that we wish to consider for our localisation systems.

However, up to this point, we are still restricting ourselves to the somewhat myopic view that the way we should approach visual localisation is to use the same rigid front-end procedure used in VO. That is to say, we still try to run our out-

of-the-box interest point detector/descriptor and expect to match scenes that could have very different appearances. Although we have presented a technique for coping with illumination changes during the day, this does not address more long-term issues, such as different weather and seasons.

To accomplish this, we require a different approach. We shall move beyond the traditional pipeline presented in Section 2.2 and explore what is possible if we leverage prior visual surveys to learn what is unique in the environment. This will be the focus of the next chapter.

Chapter 5

Scene Signatures

5.1 Introduction

This chapter presents the core contribution of the thesis—the idea of learning place-dependent feature detectors for long-term, persistent metric localisation. For decades the standard approach to vision-based localisation has been to use a point-feature-based approach, whereby salient image regions are both detected and described in a compact manner to enable efficient point correspondences across different images. Typically, these feature detectors and descriptors are rigid procedures that are applied to all images over all places without any consideration of prior or domain knowledge. This concept of just blindly applying an off-the-shelf feature detector to perform complex vision tasks is what is being challenged, as there are at least two major problems with this approach.

Firstly, matching point features for metric pose estimation typically fails under extreme appearance changes, such as different lighting conditions and weather conditions (Furgale and Barfoot, 2010; McManus, 2010; Churchill and Newman, 2012). The previous chapter presented an attempt at addressing the issue of lighting changes by using an illumination-invariant image transform (McManus et al., 2014a; Maddern et al., 2014a). However, these techniques are not suited to dealing

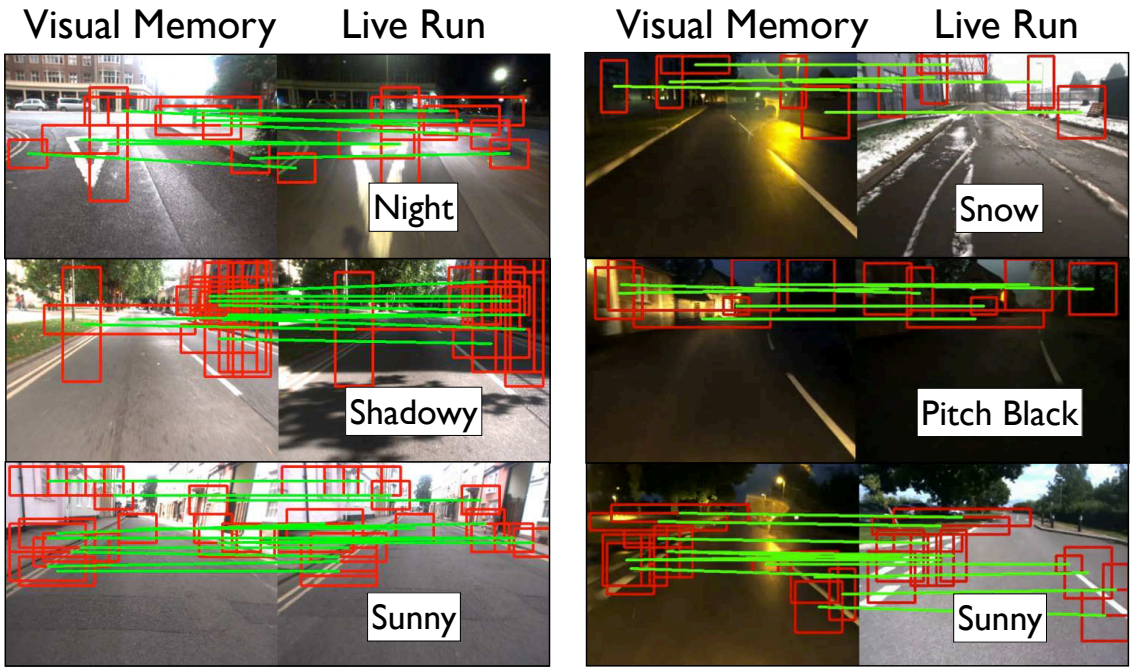


Figure 5.1: An illustration of our scene-signature approach for localisation. The left column represents a visual memory and the right column represents various live runs where we have successfully localised against the memory. Despite this challenging image sequence, we are able to match larger, distinctive elements across varying weather and lighting conditions to produce rough, metric pose estimates. Our point-feature-based system was unable to localise under such conditions.

with gross appearance changes from night-to-day or summer-to-winter. Although Valgren and Lilienthal (2010) showed topological localisation across seasons was possible with point features, metric localisation was never examined and the experiments were conducted on a very limited set of images. Milford and Wyeth (2012) showed that matching sequences of images can enable vast robustness for localising across drastic appearance conditions, but similarly, this work only considerd topological localisation.

The second and perhaps most significant issue with the standard feature-based approach is that it is somewhat naive for the task of localising against a known map. As we operate with the requirement that an autonomous vehicle can only navigate in pre-mapped location, we have at our disposal a collection of datasets representing varying appearances of the scene. The goal then becomes to leverage this prior data

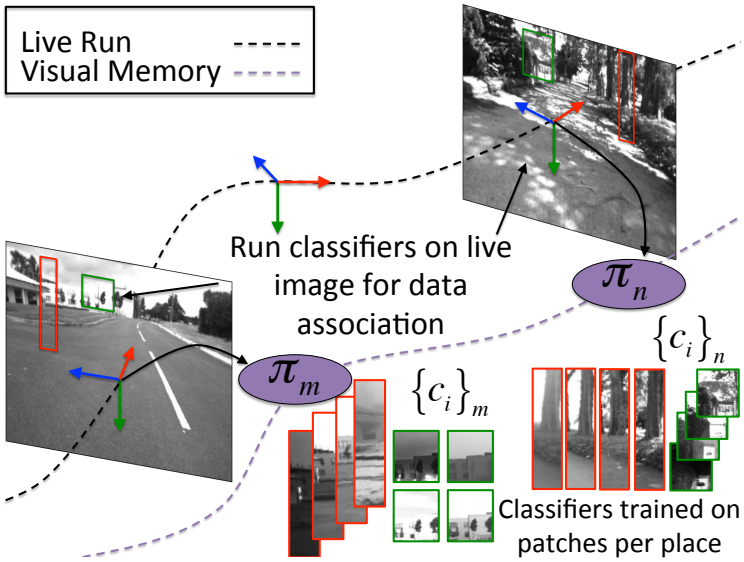


Figure 5.2: Offline, we learn scene signatures in the form of SVM classifiers, where each classifier is associated with a particular place, $\boldsymbol{\pi}_p$. At run-time, we use the bank of pre-trained classifiers associated with the nearest place, $\boldsymbol{\pi}_p$, to perform data association and then localisation. By using larger, distinctive visual elements, we are able to localise in regions with extreme appearance change, where the point-feature-based counterpart fails.

and learn what is unique and important at a given place, such as unique window facades, tree silhouettes, signs, buildings, etc.

This section presents an unsupervised method to learn a bank, or set, of *place-dependent* classifiers, $\{c_i\}_p$, that fire on unique visual elements specific to a particular place, $\boldsymbol{\pi}_p$. As the vehicle progresses through its environment, it can retrieve the classifiers, $\{c_i\}_p$, relevant to its location, $\boldsymbol{\pi}_p$, and use them to identify known structure in the live image feed. These broad level features, called *scene signatures*, are used to create a "weak localiser" of sufficient accuracy to provide coarse local, metric information about the vehicle's pose, even when faced with scenes that appear drastically different (see Figure 5.1).

Immediately one should ask, "for what tasks is such precision adequate?" Ultimately, we envision a hierarchical system in which at the top level we have very crude topological localiser which outputs the gross location of the vehicle. This

output drives the localiser described in this work which takes a topological hint and returns a metric position accurate in orientation but with perhaps tens of centimeters in translational error. We assert that for autonomous road-vehicle navigation and control, we only need a coarse metric estimate of the vehicle's pose, after which, lower level lane following and/or curb detection algorithms can be applied to refine the estimate for a vehicle controller. Again, this is a shift from the traditional methods that try and obtain centimeter-level accuracy. For a road vehicle with on-board obstacle avoidance and lane following software, we assume that global localisation accuracy to the half metre is sufficient.

At a high level, the steps involved in our localisation system work as follows:

1. Initialisation in the map (e.g., place recognition system).

2. Use dead reckoning (e.g., VO or wheel odometry) to predict what place, $\boldsymbol{\pi}_p$, the vehicle is close to and load the bank of spatially-indexed SVM classifiers, $\{c_i\}_p$, associated with that place.

3. Use each SVM classifier to search for associations in the live image within a local window in image space. In other words, there is a mask for each feature which limits the area in which to match. [1]

4. Use optimised landmark positions associated with each scene signature to solve for the optimal transformation estimate against the map.

An illustration of our system is shown in Figure 5.2.

5.2 Related Work

Recently, there has been a number of attempts to shift away from the traditional approach of blindly applying an out-of-the-box point-feature detectors/descriptors

[1] A typical number of scene signatures per place would be 40-50.

for egomotion estimation and localisation. Richardson and Olson (2013) present a method for learning an optimal feature detector for VO tasks. Their method searches the space of convolution filters to find the detector that minimises reprojection error. Although this method is aimed at improving standard detection methods for an application specific task, it still focused on using point features, which works well for VO, but not for localisation (e.g., matching a sunny day against a rainy day).

Lategahn et al. (2013) present a method for learning an optimal whole-image descriptor for place recognition. They use a genetic optimisation approach to find the optimal combination of fundamental feature blocks to construct their optimal descriptor. However, as with other methods, such as SeqSLAM (Milford and Wyeth, 2012), this can only inform the system of the topological position of the vehicle; it does not provide a metric estimate, which is important for us as we are interested in controlling a vehicle.

Rublee et al. (2011) developed a new feature called ORB, which builds upon the FAST detector (Rosten and Drummond, 2006) and the BRIEF descriptor (Calonder et al., 2012). They use a greedy learning algorithm for de-correlating BRIEF features under rotational invariance. However, as this is still based on low-level structure, data association remains hard under extreme appearance change. von Hundelshausen and Sukthankar (2012) present a noteworthy descriptor that goes beyond point features and instead constructs a network of nodes and directed edges, where each edge is a descriptor in the network, referred to as a "d-token". However, because these descriptors directly sample pixel intensities, this would not be suitable for the types of extreme appearance changes we are considering (see Figure 5.1).

Ultimately, we are concerned with the problem of long-term, robust localisation in outdoor environments, which experience a great deal of appearance changes (e.g., time of day and/or time of year). One approach to this problem would be a system like Experience-Based Navigation (Churchill and Newman, 2012), which works

as follows. As a vehicle continuously revisits the same environment it catalogues distinct image sequences for future traverses (these are referred to as *visual experiences*). Whilst revisiting the same place, the system will attempt to localise against these archived visual experiences and if unsuccessful due to significant appearance changes, the system will save the live video stream as a new distinct experience. Although this is a feasible approach, we offer an alternative that tries to learn what elements in the environment are stable across all appearances. In this way, localisation is not done against numerous experiences, but rather just a collection of distinctive scene elements.

Note that this approach is very different from the localisation and mapping systems of Davison et al. (2007); Davison and Murray (2002), which use image patches as their landmarks. These methods still rely on interest-point detection to find the patches and they use small patches (e.g., 11×11 pixels in size). By construction, scene signatures are large distinctive elements in the scene that can be matched across extreme appearance changes.

Several researchers have investigated the idea of semantic localisation, which shares a similar viewpoint that higher-level information is useful for localisation. Atanasov et al. (2014) present a system that uses Random Finite Sets (RFS) to represent semantic information from their object detector, which allows them to account for missed detections, false positives, and perform data association. They show how the RFS observation model is equivalent to a matrix permanent computation, which makes the filtering problem tractable. Renato F. Salas-Moreno and Davison (2013) present SLAM++; an *object oriented* approach to SLAM that uses 3D models of common indoor objects, such as chairs and tables, to perform real-time, full 6-Degree of Freedom (DOF) SLAM. Ko et al. (2013); Yi et al. (2009) present an approach for semantic mapping, active localisation, and local navigation and planning. Their system abstracts spatial relationships and actions to higher

level concepts (e.g., object is near or distant). Anati et al. (2012) side stepped the problem of data association by using the dense heat maps produced by the object detectors and incorporate a per-pixel likelihood score for observing a particular class. They incorporated these soft detections using a particle filter and demonstrated the system working in a large indoor environment. Bao and Savarese (2011) introduced the concept of Semantic Structure from Motion (SfM), which attempts to find the optimal maximum-likelihood estimate of camera poses, objects, and points. They show that in addition to outperforming a point-feature-based SfM system, they can also improve object detection due to extra geometric information when compared to detecting objects in images alone. Castle et al. (2007) developed a hybrid monocular SLAM system that combined traditional sparse features with known planar objects.

Although all of these aforementioned approaches shared the view that matching low-level structure isn't always the best approach for localisation, they introduce the challenging problem of scene understanding. As we will show, it is not necessarily the case that a visual element must have semantic meaning to be valuable for localisation. For instance, our algorithm is able to find unique rectangular strips that encompass various structures, such as a building, road, and vegetation. This on it's own is not a singular class, but simply a unique visual strip associated with that place. We are therefore not limited to a predefined set of classes, but instead let the algorithm find what is unique in a given place.

With regards to learning unique visual elements, Doersch et al. (2012) presented a method for extracting geo-distinctive image patches from a collection of images of London and Paris. Their method was able to find mid-level image patches of windows, balconies, and street signs which clearly distinguished the Parisian streets from the London streets. The method is, in principle, straightforward and relies on a large amount of data and an iterative, discriminative training scheme similar to Singh et al. (2012). We adopt similar machinery and use an iterative discriminative

training scheme, but have far less training data and frame the problem differently in order to find locally distinctive (i.e., in image space) mid-level patches that can be associated across different appearances. We are interested in finding discriminative features that occur frequently in the same local region in the image, but infrequently in the rest of the image.

5.3 System Overview

5.3.1 Offline Learning

There are two steps in the learning phase. The first step involves training SVM classifiers to find a set of candidate scene signatures through an unsupervised, iterative, training technique similar to Doersch et al. (2012); Singh et al. (2012), which is designed to find groups of features that occur frequently in an area of interest, but infrequently everywhere else. The second step performs landmark refinement to find optimal landmark locations for each scene signatures (i.e., optimal in that the landmark location minimises reprojection error over a window of frames). The output of these two processes yield a bank of motion-consistent classifiers, $\{c_i\}_p$, for each place, $\boldsymbol{\pi}_p$.

5.3.1.1 Training Algorithm

Our training data consist of a collection of images at approximately the same location and viewpoint under a variety of appearances (see Figure 5.3). We collected these data using a vehicle equipped with an INS system, and defined *places* as physical locations spaced 10 m apart along the driven route according to INS. The important aspect of the training data is that the viewpoint is as similar as possible. Ideally, we would like the viewpoints to be identical, but this is not possible due to inaccuracies in the INS and because the driven routes vary from one dataset to the next.

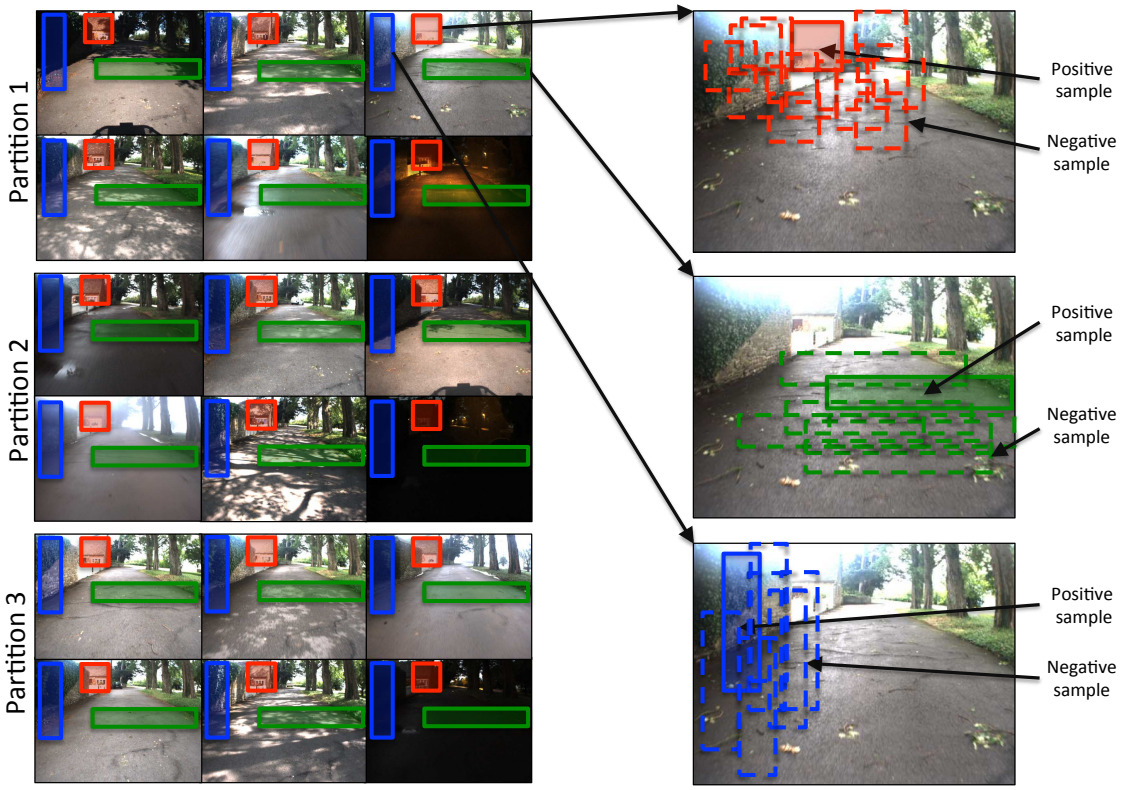


Figure 5.3: Illustration of the first stage of our training algorithm. The training data consist of images of the same place under varying appearance conditions. These images are partitioned into 3 groups, such that each group has as much visual variability as possible. Then, we sample a set of shapes from each image, represented in this figure by coloured rectangles. The steps proceed as follows. Take a shape (e.g., red), compute HOG descriptors (Dalal and Triggs, 2005) for each red shape in all images in partition 1. These will be the positive set. Sample the image around that shape and compute HOG descriptors (shown on the right column). These will be the negative set. Train a linear SVM classifier and use this classifier on partition 2, which acts as a validation set. Take the top K firings in partition 2 and use these as positives to retrain the SVM. The new SVM, which was trained on just the subset of positives in partition 2 and its respective negative samples, is then applied to partition 3 in the same fashion. The validation set becomes the training set and the process repeats, wrapping around to the first partition again. If the top K firings in each respective partition remain unchanged, then we have converged to a discriminative feature. If we have not converged within 3 iterations, we terminate the process and reject the classifier as a candidate scene signature.

Referring to Figure 5.3, the reader will see three shapes drawn on every image (i.e., the red, blue, and green rectangles). The goal is to find out which set of patches represent "stable" visual elements. By "stable," we mean that if we trained a classifier to detect these types of patches, we would expect the classifier to find the same visual elements in a validation set, regardless of appearance conditions. In other words, we want a classifier that will always fire on the same set of trees, for example, in the same physical place, regardless of time of day, time of year, or weather (i.e., we want stability across all appearances). It is important to note that not all patches represent stable visual elements. For example, one can imagine that a classifier trained to detect the green rectangle may fire anywhere along the curb (i.e., this is not a locally distinctive feature as the surrounding region looks similar in appearance). However, the other two shapes (i.e., the red and the blue shape) would likely serve as good patches from which to train, because the underlying visual element appears very distinctive. In the red patch, we see a building with a window. This appears nowhere else locally and can be associated across all the images. The blue patch is a unique strip that transitions vertically from the ground to a wedge of brick wall to a bush. We might expect this to be very distinctive as well.

The question is, how do we determine which shapes serve as a basis for a stable classifier? We could train a classifier for each candidate set of shapes (e.g., the red, blue, and green) and then use the classifier on a hold-out set to see where the detections fire relative to the groundtruth location. However, this is unappealing as we would have to define a closeness threshold in image space to label a positive detection. Additionally, it fails to take into account that not all of the positive patches are informative. For instance, in Figure 5.3, we can see that some of the blue patches encompass textureless, black regions in the image and would not be helpful to use as positive examples. Thus, we seek an unsupervised training technique that can accomplish the following two tasks: (i) it must be able to identity what types

of patches represent "stable" elements (e.g., red, blue, or green?), and (ii) it must be able to select a discriminative subset of the positives for training (e.g., ignore the textureless blue patches). Fortunately, we can use an iterative training scheme similar to Doersch et al. (2012); Singh et al. (2012) to accomplish this task.

The basic idea is as follows. Consider separating the images in Figure 5.3 into three partitions as shown[2], and training an SVM classifier on the red patches, which, initially, are all labeled as positives, with the negatives being sampled around the local region[3]. After training this classifier, we apply it to the red rectangles in the second partition. We then rank each red rectangle in the second partition according to its score and train a new SVM using the top K detections[4]. The new classifier which was trained on just the subset of positives in partition 2 and its respective negative samples, is then applied to partition 3 in the same fashion. The top K firings from the validation set become the new positive examples from which to retrain. This new SVM would then be applied back to the first partition and the cycle would continue until convergence criteria are met. The convergence criteria require that the top K firings in each respective partition do not change, as this implies that we have found a subset of discriminative patches. Note that this is a very conservative approach, as it would select only the most representative examples of the visual element we seek to classify. However, we gladly trade recall for higher precision in this context, as we are concerned with limiting the number of mis-associations for pose estimation. If the convergence criteria are not met within three iterations[5], we reject the candidate classifier as representing a "stable" visual element. Pseudocode for the training is provided in algorithms 1 and 2.

For our implementation, we used a fixed set of 296 predefined shapes for ev-

[2]As was done in Doersch et al. (2012).

[3]Somewhat similar to Torralba et al. (2007).

[4]We set $K = 5$ as done in Doersch et al. (2012).

[5]We chose three as was done in Doersch et al. (2012). Note that Singh et al. (2012) came to a similar conclusion that only 4-5 iterations are necessary.

Algorithm 1 Scene Signatures Training Algorithm

function $\{C\}$ =FindSceneSignatures($\{\mathcal{I}\}$) ▷ Collection of view-point aligned images of the same place.
  $\{\mathcal{S}\}$ = GenShapes() ▷ Produce a collection of different rectangular shapes.
  for $\forall\ \mathcal{S}_i \in \{\mathcal{S}\}$ **do**
    for $\forall\ \mathcal{I}_j \in \{\mathcal{I}\}$ **do**
      $\{\mathcal{P}, \mathcal{N}\} \Leftarrow$ ExtractPatches($\mathcal{S}_i, \mathcal{I}_j$) ▷ Extract positive and negative patches and add to set.
    end for
    $[C,$ has_converged$]$ = DoIterativeTraining($\mathcal{P}, \mathcal{N}$) ▷ Perform iterative training to generate a classifier.
    if has_converged **then**
      $\{C\} \Leftarrow C$ ▷ If converged, add to the set
    end if
  end for
  return $\{C\}$ ▷ The final set of classifiers.
end function

Algorithm 2 Iterative Training Algorithm

function C =DoIterativeTraining($\mathcal{P}, \mathcal{N}$)▷ $\mathcal{P}$ are positive, $\mathcal{N}$ are negative.
  $\mathcal{P} \Rightarrow \{\mathcal{P}_i\},\ \ \mathcal{N} \Rightarrow \{\mathcal{N}_i\}$ ▷ Divide $\mathcal{P}$ and $\mathcal{N}$ into equal sized disjoint sets.
  while not max_iterations & **not** has_converged **do**
    for $i = 1$: numel **do**
      $C \Leftarrow$ SVMTrain($\mathcal{P}_i, \mathcal{N}_i$) ▷ Train classifier on current partition.
      $\mathcal{P}_{i+1,new} \Leftarrow$ DetectTop($C, \mathcal{P}_{i+1}, M$) ▷ Detect and take top M. Also note that we need to wrap to i=1 when we reach the end.
      if $\mathcal{P}_{i+1,new} = \mathcal{P}_{i+1}$ **then** ▷ A success is when the cluster memberships do not change.
        success[i]=1
      else
        success[i]=0
      end if
      $\mathcal{P}_{i+1} = \mathcal{P}_{i+1,new}$ ▷ Set the new positives.
    end for
    has_converged = (sum(success) == numel) ▷ We have converged if the cluster memberships for each partition did not change.
  end while
  return [C, has_converged] ▷ The final classifier.
end function

ery image. This pattern was generated by taking various permutations of square groupings over a grid on the image. From this fixed template of shapes, the training algorithm selects the subset using the process described above. It would be ideal to use a larger set of shapes; however, this increases the training times, since the procedure is performed on every shape.

Note that, at present, we only use appearance information and do not try and estimate depth from the monocular sequences. It is not immediately obvious if this would help as the depth estimates would be difficult to capture for distance objects and there are many patches with homogenous texture (e.g., the road). Nonetheless, adding depth information into the feature vector is an interesting addition that will be examined in the future.

It is also worth mentioning that if the images are not well aligned, fewer scene signatures will be detected since the iterative training procedure will not converge. Due to inaccuracies in the INS, this was observed in places where the vehicle executed a sharp turn (e.g., a 90 degree turn). As a result, the system would typically fail to localise during turns, in which case, we would integrate the odometry to continue to provide a pose estimate.

The next stage in this process looks at the stability of the classifiers over a local window of images and optimises a motion-consistent landmark location for each scene signature.

5.3.1.2 Landmark Refinement

The final step serves two purposes: (i) to eliminate bad candidate classifiers and (ii) compute a landmark position for each scene signature to enable metric localisation.

Consider one of the images in Figure 5.3.1.1 and imagine taking a window of images forward and backward in time from the initial training image to test the

classifier on each image in the window[6]. As the vehicle moves smoothly over this image sequence, one would expect the feature detections to move smoothly over time as well. Figure 5.4 illustrates this process. We use 2-point RANSAC to compute a line of best fit for the temporal x-y locations and reject any candidate classifiers if the ratio of inliers is less than half of the samples. If the inlier set is over half, we take this set of feature detections to perform landmark adjustment.

In our original work (McManus et al., 2014a), we estimated the landmark position by performing left-to-right stereo matching. However, as these features represent large patches in the image, template matching to obtain a single point estimate is not very sensible due parallax effects. Instead, we first discretely sample a number of possible depths, $\{z_i\}_p$, from 0-100 m with a resolution of 1 m and compute the total reprojection error over the window for each depth. We then take the best depth and perform a non-linear refinement around this initial guess.

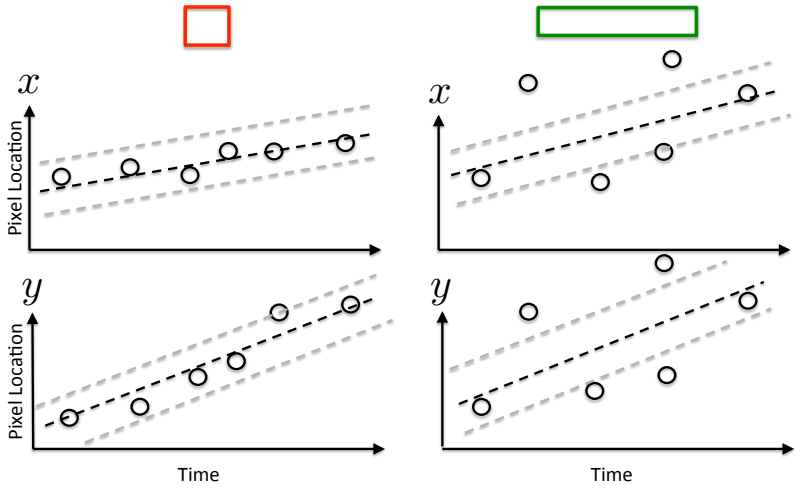
Assuming that we have odometry along with the training data, such as wheel odometry or VO, we can use the known incremental transformations, $\{\mathbf{T}_{1,0}, \ldots, \mathbf{T}_{k,k-1}\}$, between each image to reproject the landmark, $\mathbf{p}_p^i$, defined in $\underrightarrow{\mathcal{F}}_p$, into each frame in the window, $\underrightarrow{\mathcal{F}}_j$, according to our monocular camera model:

$$\boxed{\mathbf{y}_j^i := \mathbf{h}(\mathbf{T}_{j,p}\boldsymbol{p}_p^i) + \mathbf{v}_j^i = \frac{1}{z}\begin{bmatrix} x\ f_u + z\ c_u \\ y\ f_v + z\ c_v \end{bmatrix} + \mathbf{v}_j^i, \qquad \mathbf{v}_j^i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_j^i)} \tag{5.1}$$

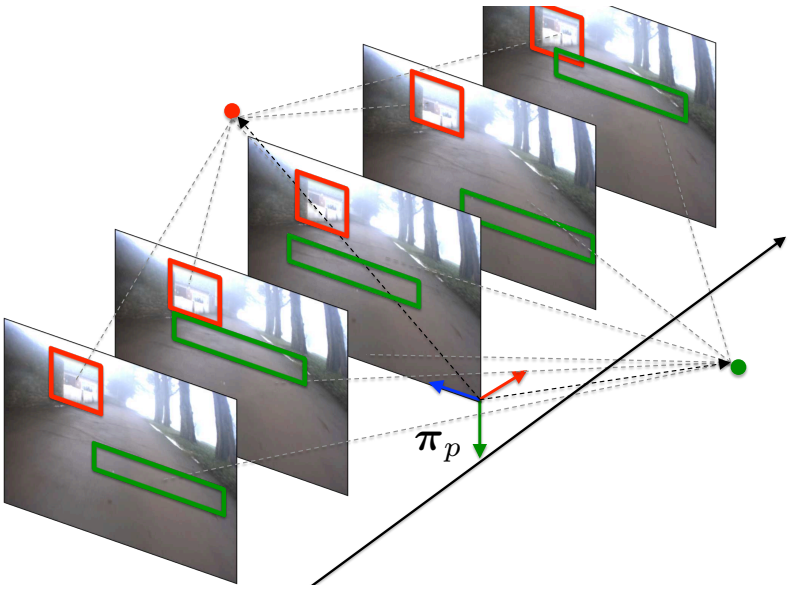
where $\mathbf{y}_j^i = [u, v]^T$ in the j^{th} image, $\mathbf{T}_{j,p}$ is the transformation from $\underrightarrow{\mathcal{F}}_p$ to $\underrightarrow{\mathcal{F}}_j$, $\boldsymbol{p}_p^i$ is the homogenous form of $\mathbf{p}_p^i$, and $\mathbf{R}_j^i$ is the noise covariance term. Note the similarities with the stereo model introduced in Section 2.2. We have simply omitted the disparity term.

Our objective function is then just an uncertainty-weighted squared difference

[6]In our experiments, the window was taken to be the distance between places, which is 10 m.



Temporal x-y pixel locations for the red and blue detections over the window of images shown below. As the vehicle moves smoothly through the scene, we expect the detections to vary smoothly as well (i.e., we do not expect the detections to jump from one side of the image to the other) . We use 2-point RANSAC to fit a line to the temporo-spatial data to asses the quality of the classifier. If more than half detections are outliers, the classifier is rejected. Otherwise, the inlier set are used for landmark refinement.



Landmark refinement using the inlier set of detections. In this example, the green candidate would likely not pass the smoothness check and would be culled.

Figure 5.4: Each candidate classifier generated by the training scheme described in Section 5.3.1.1 is subjected to a round of temporal checks during the landmark refinement stage. A window of frames is taken around each place and the classifier is fired on all frames in order to compute statistics on the stability of the classifier. Inlier detections are then used for landmark refinement.

between the observed location, $\mathbf{y}_j^i$, (given by the classifier) and the predicted location, $\mathbf{h}(\mathbf{T}_{j,p}\boldsymbol{p}_p^i)$, given by our reprojection function:

$$J(\mathbf{p}_p^i) := \frac{1}{2}\sum_{j=0}^{M}\left(\mathbf{y}_j^i - \mathbf{h}(\mathbf{T}_{j,p}\boldsymbol{p}_p^i)\right)^T(\mathbf{R}_j^i)^{-1}\left(\mathbf{y}_j^i - \mathbf{h}(\mathbf{T}_{j,p}\boldsymbol{p}_p^i)\right), \tag{5.2}$$

$$= \frac{1}{2}\left(\mathbf{y} - \mathbf{h}(\mathbf{p}_p^i)\right)^T\mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{h}(\mathbf{p}_p^i)\right), \tag{5.3}$$

where

$$\mathbf{y} := \begin{bmatrix}\mathbf{y}_0^i\\ \mathbf{y}_1^i\\ \vdots\\ \mathbf{y}_M^i\end{bmatrix}, \qquad \mathbf{h}(\mathbf{p}_p^i) := \begin{bmatrix}\mathbf{h}(\mathbf{T}_{0,p}\boldsymbol{p}_p^i)\\ \mathbf{h}(\mathbf{T}_{1,p}\boldsymbol{p}_p^i)\\ \vdots\\ \mathbf{h}(\mathbf{T}_{M,p}\boldsymbol{p}_p^i)\end{bmatrix}, \qquad \mathbf{R} = \mathrm{diag}(\mathbf{R}_0^i, \mathbf{R}_1^i, \ldots, \mathbf{R}_M^i). \tag{5.4}$$

Using the techniques described in Section 2.1.2, we perturb our design variable about some operating point to linearise the system:

$$J(\bar{\mathbf{p}}_p^i + \delta\mathbf{p}^i) = \left(\mathbf{y} - \mathbf{h}(\bar{\mathbf{p}}_p^i) - \frac{\partial\mathbf{h}}{\partial\delta\mathbf{p}^i}\delta\mathbf{p}^i\right)^T\mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{h}(\bar{\mathbf{p}}_p^i) - \frac{\partial\mathbf{h}}{\partial\delta\mathbf{p}^i}\delta\mathbf{p}^i\right). \tag{5.5}$$

Taking the derivative of $J(\cdot)$ with respect to the perturbation, setting it to zero and solving the system yields the set of normal equations:

$$\left(\mathbf{H}_\mathbf{p}^T\mathbf{R}^{-1}\mathbf{H}_\mathbf{p}\right)\delta\mathbf{p}^i = -\mathbf{H}_\mathbf{p}^T\mathbf{R}^{-1}\left(\mathbf{y} - \mathbf{h}(\bar{\mathbf{p}}_p^i)\right), \tag{5.6}$$

where,

$$\boxed{\mathbf{H}_\mathbf{p} := \frac{\partial\mathbf{h}}{\partial\mathbf{p}_p^i} = \begin{bmatrix}f_u/z & 0 & -f_u x/z^2\\ 0 & f_v/z & -f_v y/z^2\end{bmatrix}} \tag{5.7}$$

LM is used for the iterative optimisation.

To summarise, after finding a coarse initial guess from our discrete sampling we solve the above nonlinear least-squares problem using LM. This landmark position is then stored with the scene signature and used for pose estimation online. The end result of this procedure is a set of motion consistent scene signatures, $\{c_i\}_p$, defined for each place, $\boldsymbol{\pi}_p$. Example scene signatures can be seen in Figure 5.5.

It is worth commenting on the fact that we use a monocular camera model and not a stereo camera model as was done in our original work (McManus et al., 2014b). The reason for doing this is because in our original implementation, we did not optimise for the landmark position offline, and simply used left-to-right stereo matching to produce the landmark position. By optimising over a window, we are ensuring that the landmarks are motion consistent and engineering them to be better for localisation. This also has the added attraction that it can be done with a monocular camera and wheel odometry instead of stereo. However, for our experiments, we still use a stereo camera for VO, but perform the localisation using just the left camera.

5.3.2 Online Localisation

For the task of localisation, it is assumed that the vehicle is seeded with its initial pose by some external source such as GPS. The goal then becomes loading the bank of classifiers from the closest place and attempting to localise against that place. As the vehicle integrates its motion along the way, the system tracks where in the topometric map the vehicle is in order to cache the appropriate bank of classifiers. Note that this is not answering the kidnapped robot problem. This system is performing position tracking in its map and loading a set of scene signatures associated with the closest place. The system does not try to match within a window since the places are spaced apart by 10 m.

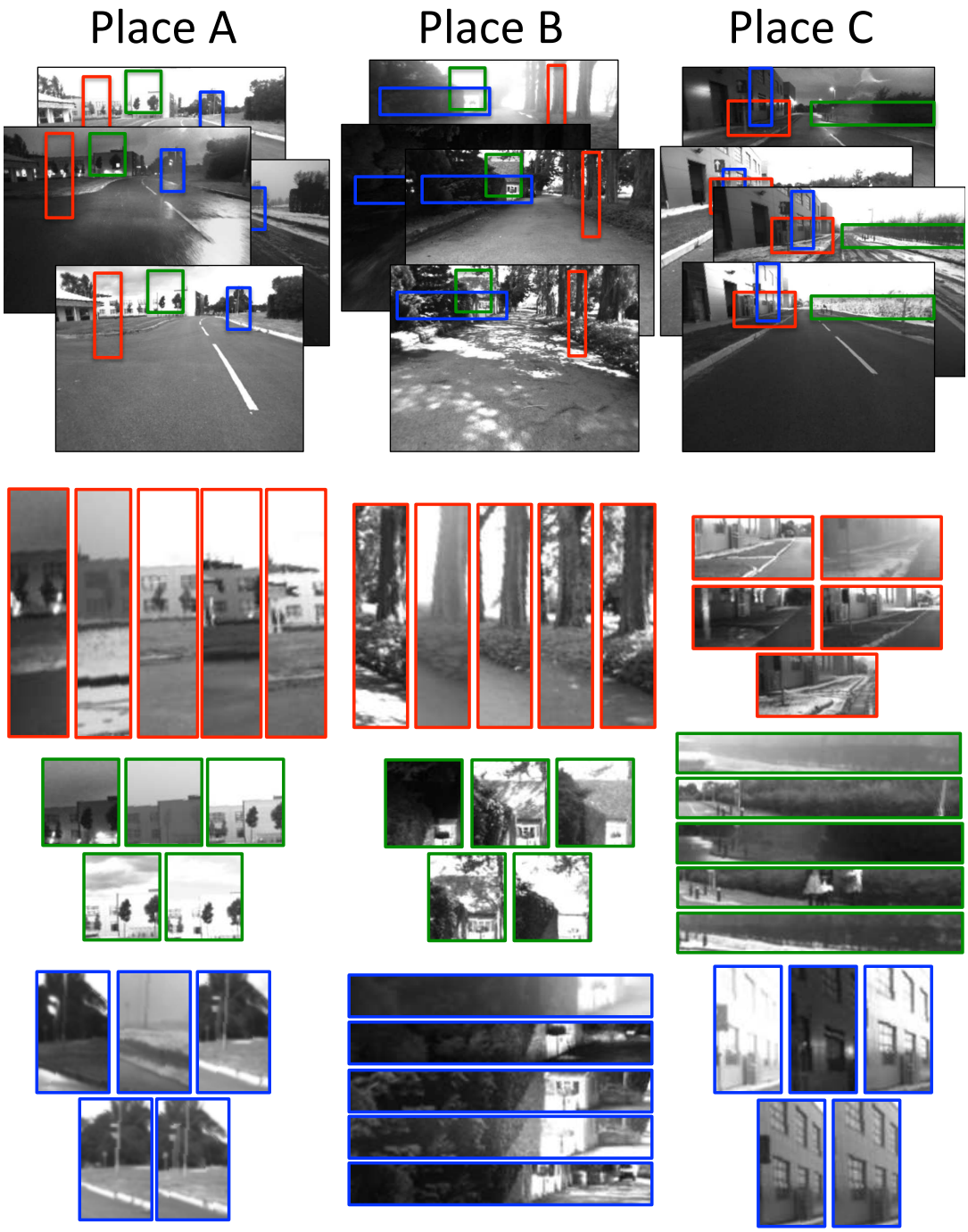


Figure 5.5: Example scene signatures learned by our algorithm. Image sets from the same place with varying appearances (represented by run-times in this figure) are used offline to learn these distinctive scene signatures. SVM classifiers are trained for each cluster of scene signatures and can be used at run-time on the live image stream to perform data association, followed by metric pose estimation. Note that the shapes vary in size and dimension and tend to pick up things like changes in structures, as these are very distinctive (e.g., from road, to grass, to building, to sky).

5.3.2.1 Weak Localisers

This section introduces the notion of a "weak localiser". As the scene signatures represent large image patches that are mostly located in the far field, the translational estimates from the nonlinear solve are not very accurate. Additionally, since the distribution of depths within a patch can have a large deviation and could be non symmetric (e.g., consider the case where a patch encompasses road, building, and sky) using a single value to define the depth may not accurately represent the 3D position of the feature. As a result, we use a strong motion prior from VO to bound the solution in translation.

At runtime, we load the bank of classifiers, $\{c_i\}_p$, associated with the closest place $\boldsymbol{\pi}_p$ and use them on the live image at time t_k to produce a set of measurements, $\mathbf{y}_k^i$. As each classifier in the map has an associated landmark position, $\mathbf{p}_p^i$, we can use our camera model (5.1) to predict the location of a landmark in the live frame, according to the transformation matrix, $\mathbf{T}_{k,p}$:

$$\mathbf{y}_k^i = \mathbf{h}(\mathbf{T}_{k,p}\boldsymbol{p}_p^i) + \mathbf{v}_k^i, \qquad \mathbf{v}_k^i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_\mathbf{k}^i). \tag{5.8}$$

As stated earlier, we use a strong motion prior to predict the transformation, $\hat{\mathbf{T}}_{k,p}$. Including the prior estimate, $\hat{\mathbf{T}}_{k,p}$, the final least-squares system we seek to optimise is given by the following:

$$J(\mathbf{T}_{k,p}) = \frac{1}{2}\begin{bmatrix} \mathbf{q}(\mathbf{T}_{k,p}\hat{\mathbf{T}}_{k,p}^{-1}) \\ \mathbf{y}_k - \mathbf{h}(\mathbf{T}_{k,p}, \mathbf{p}_p) \end{bmatrix}^T \begin{bmatrix} \mathbf{P}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{q}(\mathbf{T}_{k,p}\hat{\mathbf{T}}_{k,p}^{1}) \\ \mathbf{y}_k - \mathbf{h}(\mathbf{T}_{k,p}\boldsymbol{p}_p) \end{bmatrix}, \tag{5.9}$$

where

$$\mathbf{y}_k := \begin{bmatrix} \mathbf{y}_k^0 \\ \vdots \\ \mathbf{y}_k^M \end{bmatrix}, \quad \mathbf{p}_p := \begin{bmatrix} \mathbf{p}_p^0 \\ \vdots \\ \mathbf{p}_p^M \end{bmatrix}, \quad \mathbf{R} := \operatorname{diag}(\mathbf{R}_k^0, \ldots, \mathbf{R}_k^M), \tag{5.10}$$

and $\mathbf{q}(\cdot)$ is a function that takes two SE(3) transformation matrices and computes a 6×1 error vector. Recalling from Section 2.1.3, the orientation parameterisation is the rotation vector, $\boldsymbol{\phi}$. Thus, in our case, the orientation error is given by

$$\boldsymbol{\phi}^{\wedge} := \mathbf{R}_{k,p}\hat{\mathbf{R}}_{k,p}^{T} - \mathbf{1} \tag{5.11}$$

$$\implies \boldsymbol{\phi} := \left(\mathbf{R}_{k,p}\hat{\mathbf{R}}_{k,p}^{T} - \mathbf{1}\right)^{\vee}. \tag{5.12}$$

Thus, $\mathbf{q}(\cdot)$ is simply defined as

$$\mathbf{q}(\mathbf{T}_{k,p}\hat{\mathbf{T}}_{k,p}^{-1}) := \begin{bmatrix} \mathbf{t}_{k}^{p,k} - \hat{\mathbf{t}}_{k}^{p,k} \\ \left(\mathbf{R}_{k,p}\hat{\mathbf{R}}_{k,p}^{T} - \mathbf{1}\right)^{\vee} \end{bmatrix}, \tag{5.13}$$

where $(\cdot)^{\vee}$ is the inverse-cross operator (Barfoot and Furgale, 2014). The final linearised system of equations takes the following form:

$$\begin{bmatrix} \mathbf{Q}_{\mathbf{x}}^{T} & \mathbf{H}_{\mathbf{x}}^{T} \end{bmatrix} \begin{bmatrix} \mathbf{P}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{Q}_{\mathbf{x}} \\ \mathbf{H}_{\mathbf{x}} \end{bmatrix} \delta\mathbf{x} = - \begin{bmatrix} \mathbf{Q}_{\mathbf{x}}^{T} & \mathbf{H}_{\mathbf{x}}^{T} \end{bmatrix} \begin{bmatrix} \mathbf{P}^{-1} & 0 \\ \mathbf{0} & \mathbf{R}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{q}(\mathbf{T}_{k,p}\hat{\mathbf{T}}_{k,p}^{-1}) \\ \mathbf{y}_{k} - \mathbf{h}(\mathbf{T}_{k,p}\boldsymbol{p}_{p}) \end{bmatrix}, \tag{5.14}$$

where $\mathbf{Q}_{\mathbf{x}} := \partial\mathbf{q}/\partial\mathbf{x}_{c}$ and $\mathbf{H}_{\mathbf{x}} := \partial\mathbf{h}/\partial\mathbf{x}_{c}$, which can be computed using the techniques from Section 2.1. The first Jacobian is simply

$$\boxed{\mathbf{H}_{\mathbf{x}} = \frac{\partial\mathbf{h}}{\partial\mathbf{x}_{c}} = \frac{\partial\mathbf{h}}{\partial\mathbf{p}_{c}}\frac{\partial\mathbf{p}_{c}}{\partial\mathbf{x}_{c}} = \begin{bmatrix} f_{u}/z & 0 & -f_{u}x/z^{2} \\ 0 & f_{v}/z & -f_{v}y/z^{2} \end{bmatrix} \begin{bmatrix} \mathbf{1} & -\mathbf{p}_{c}^{\wedge} \end{bmatrix}} \tag{5.15}$$

Where we have made use of (5.7) and (2.66). The second Jacobian, $\mathbf{Q}_{\mathbf{x}}$, is a bit more involved. Consider the translation and rotational error terms:

$$\mathbf{e}_{k,\text{trans}} = \mathbf{t}_{k}^{p,k} - \hat{\mathbf{t}}_{k}^{p,k} \tag{5.16}$$

$$\mathbf{e}_{k,\text{rot}} = \left(\mathbf{R}_{k,p}\hat{\mathbf{R}}_{k,p}^{T} - \mathbf{1}\right)^{\vee} \tag{5.17}$$

Performing a first-order linearisation gives us:

$$\bar{\mathbf{e}}_{k,\text{trans}} + \left(\frac{\partial \mathbf{e}_{k,\text{trans}}}{\partial \mathbf{x}}\right)\delta\mathbf{x} \approx \bar{\mathbf{t}}_k^{p,k} + \delta\mathbf{t}_k - \hat{\mathbf{t}}_k^{p,k} \tag{5.18}$$

$$\bar{\mathbf{e}}_{k,\text{rot}} + \left(\frac{\partial \mathbf{e}_{k,\text{rot}}}{\partial \mathbf{x}}\right)\delta\mathbf{x} \approx \left((\mathbf{1} + \delta\boldsymbol{\phi}^\wedge)\bar{\mathbf{R}}_{k,p}\hat{\mathbf{R}}_{k,p}^T - \mathbf{1}\right)^\vee \tag{5.19}$$

Subtracting off the nominal solutions leaves us with

$$\left(\frac{\partial \mathbf{e}_{k,\text{trans}}}{\partial \mathbf{x}}\right)\delta\mathbf{x} = \delta\mathbf{t}_k \tag{5.20}$$

$$\left(\frac{\partial \mathbf{e}_{k,\text{rot}}}{\partial \mathbf{x}}\right)\delta\mathbf{x} = \left(\delta\boldsymbol{\phi}^\wedge\bar{\mathbf{R}}_{k,p}\hat{\mathbf{R}}_{k,p}^T\right)^\vee \tag{5.21}$$

Immediately we see that

$$\left(\frac{\partial \mathbf{e}_{k,\text{trans}}}{\partial \mathbf{x}}\right)\delta\mathbf{x} = \begin{bmatrix}\mathbf{1} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\delta\mathbf{t}_k \\ \delta\boldsymbol{\phi}\end{bmatrix} \Longrightarrow \left(\frac{\partial \mathbf{e}_{k,\text{trans}}}{\partial \mathbf{x}}\right) = \begin{bmatrix}\mathbf{1} & \mathbf{0}\end{bmatrix} \tag{5.22}$$

For the rotational term, by assumption, the product of $\bar{\mathbf{R}}_{k,p}\hat{\mathbf{R}}_{k,p}^T$ should represent a small rotation. Thus, we approximate this as $\bar{\mathbf{R}}_{k,p}\hat{\mathbf{R}}_{k,p}^T \approx \mathbf{1} + \boldsymbol{\psi}^\wedge$. From this, we arrive at the following:

$$\left(\frac{\partial \mathbf{e}_{k,\text{rot}}}{\partial \mathbf{x}}\right)\delta\mathbf{x} \approx \left(\delta\boldsymbol{\phi}^\wedge\left(\mathbf{1} + \delta\boldsymbol{\psi}^\wedge\right)\right)^\vee \tag{5.23}$$

$$= \left(\delta\boldsymbol{\phi}^\wedge + \underbrace{\delta\boldsymbol{\phi}^\wedge\delta\boldsymbol{\psi}^\wedge}_{\approx\mathbf{0}}\right)^\vee \tag{5.24}$$

$$= \delta\boldsymbol{\phi} \tag{5.25}$$

$$\Longrightarrow \left(\frac{\partial \mathbf{e}_{k,\text{rot}}}{\partial \mathbf{x}}\right) = \begin{bmatrix}\mathbf{0} & \mathbf{1}\end{bmatrix} \tag{5.26}$$

Thus, we have that the total Jacobian for the prior term is simply given by the identity; i.e., $\mathbf{Q}_x = \partial\mathbf{q}/\partial\mathbf{x}_c = \mathbf{1}$.

For outlier rejection, we use a similar technique as described in Chapter 4, which

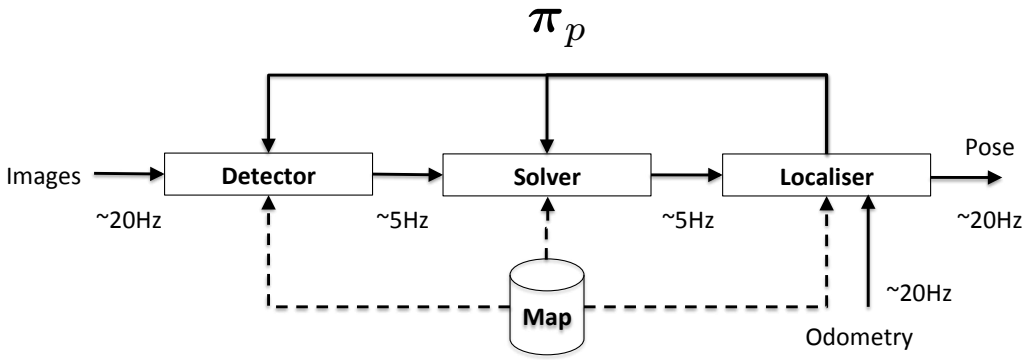


Figure 5.6: Block-flow diagram of our asynchronous localisation pipeline. In separate threads, the detection block and solver perform the scene-signature localisation as described in Section 5.3.2.1. The high-level localiser integrates the odometry measurements in between localisation updates and also performs posegraph relaxation to smooth the output.

is to predict where the feature in the map will reproject into the image plane. If the difference between the observed and predicted reprojections is beyond a threshold, then the feature match is labeled an outlier and culled. As was outlined in Section 2.1.2, we also incorporate a Huber cost function and perform LM for the nonlinear optimisation. Appendix B contains a summary of all the camera models and Jacobians derived up to this point.

5.3.2.2 Localisation Pipeline

To obtain realtime operation, an asynchronous pipeline design was used in order to fuse lower frequency localisation updates ($\sim 5\,\text{Hz}$) with high frequency VO measurements ($\sim 20\,\text{Hz}$). The pipeline is illustrated in Figure 5.6. The major processing blocks are described below.

- **Detector**: This block performs the HOG detection using a bank of classifiers, $\{c\}_p$, for the current place. Currently, OpenCV's OpenCL GPU HOG is being used for the detection. Depending on the graphics card and the number of models being used, this block runs anywhere from 2-5 Hz. Note that the detector block receives knowledge of $\boldsymbol{\pi}_p$ from the localiser block.

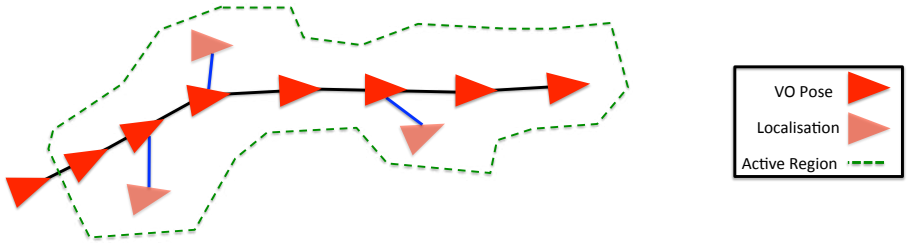


Figure 5.7: Illustration of our runtime localisation approach. Localisation updates occur at 2-5 Hz, while Visual Odometry updates occur at 15-20 Hz. After we receive a localisation update, we perform posegraph relaxation over a sliding window, indicated by the active region in green.

- **Solver**: This block performs the estimation detailed in Section 5.3.2.1 and also requires knowledge of the current place, $\boldsymbol{\pi}_p$, to load the associated landmarks for the pose solve. This block runs very fast but is limited by the rate at which the detections are given, and thus, runs at roughly the same rate as the detection block. Again, knowledge of the closest place is provided by the localiser.

- **Localiser**: This is the high-level localiser that outputs vehicle pose relative to current place, $\boldsymbol{\pi}_p$. The localiser block listens for high-rate odometry measurements that are used to predict the vehicle's pose in between the low-rate localisation updates. As the localisation updates occur slower, we run a sliding-window posegraph relaxation technique.

The posegraph relaxation technique takes into account the following constraints: (i) relative constraints from the VO, (ii) localisation constraints from the solver, and (iii) a prior constraint on the initial pose in the window. When localisation updates are not available, the system simply integrates the VO for an estimate (see Figure 5.7 for an illustration). For a derivation of the error terms and Jacobians involved in this posegraph relaxation, please see Appendix C.

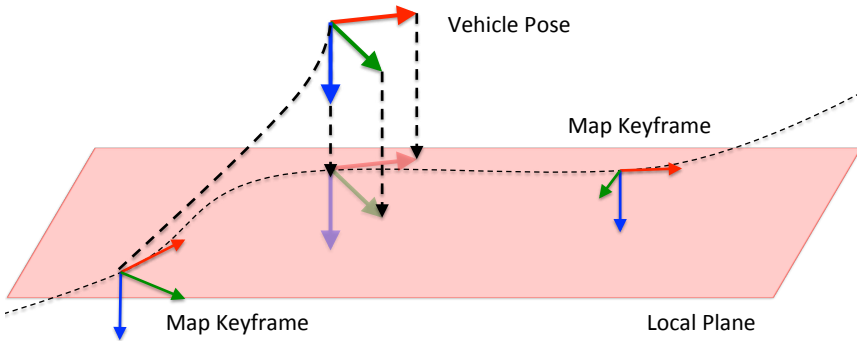


Figure 5.8: Using the two nearest keyframes in the map, we define a local plane and project the estimated vehicle pose onto this plane. This augmented pose solution is then used in the next localisation cycle. This trick proved to work well in preventing drift in the z-direction. We also adjusted the roll and pitch to lay within the plane.

5.3.2.3 Projecting to SE(2)

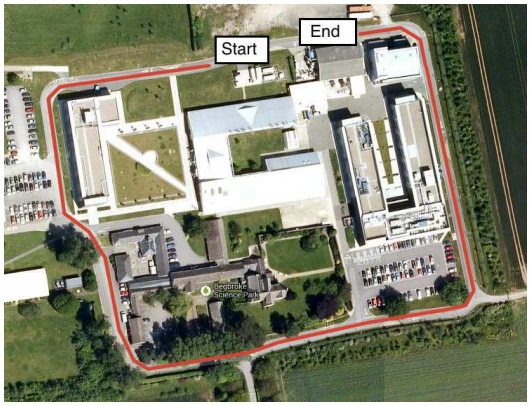
The map, VO poses, localisation solver, and posegraph relaxation all operate in SE(3). However, we found the following "trick" proved useful in improving the localisation performance by preventing significant z-drift.

Since the vehicle is driving on a road, we know that at any point in time it is reasonable to approximate the local vicinity as a plane. If we take the pose estimate in the map, we can project the pose down onto a local plane defined by the two closest keyframes. This is illustrated in Figure 5.8. Essentially, we allow for the full 6 DOF pose, but then snap the solution down to SE(2). Again, this turned out to be a helpful trick to prevent slow drift in the z-direction.

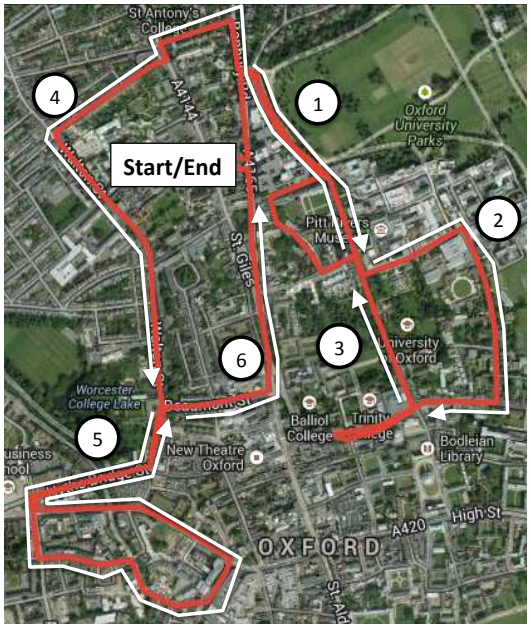
5.4 Experiments and Results

In this section we present two experiments. The first experiment analyses the matching performance of the scene signatures across extreme lighting and weather conditions (i.e., we focus purely on the front-end of the system). The second experiment looks at the performance of the complete end-to-end localisation system on kilometres of data collected from the Begbroke Science Park and central Oxford (see

Figure 5.9). Before proceeding with the experimental results, we first discuss the training and setup of the experiments.



Begbroke route (approx. 650 m).



Oxford route (approx. 8 km).

Figure 5.9: Dataset routes used in our localisation experiments. Note that for the Oxford route (right image), we only report errors relative to each segment indicated with white arrows. This was a consequence of not having enough training data due to poor GPS measurements (recall that the training images are gathered from GPS-tagged surveys).

5.4.1 Training and Setup

For Begbroke, we used 36 runs of a 650 m loop (see Figure 5.9), with places defined every 10 m along the respective routes according to our INS system. We note however, that places can be defined by other means, either manually or by using place recognition techniques. The only important factor is that the training images for a particular place have roughly the same viewpoint.

We trained our system with 31 datasets, which included images taken under different lighting and weather conditions. For our test data, we used 5 separate datasets with a wide range of visual variability, which included a sunny day run, an

Map image: clear and sunny; noon.

Test image: clear and sunny; morning.

Test image: clear; evening.

Test image: rainy; evening.

Test image: clear and snowy; morning.

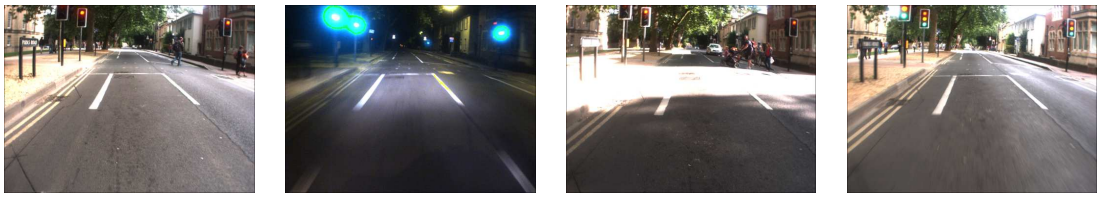
Test image: foggy and snowy; noon.

Figure 5.10: Example test images used in our Begbroke localisation experiments. These were chosen due to their large visual variability.

evening run, a rainy evening run, a snowy run, and a snowy and foggy run (example images are shown in Figure 5.10). After training the scene signatures, we picked one reference run from the training data, which will be denoted as "the map," and is the reference we localise against.

Similar to the Begbroke datasets, for Oxford, we defined places every 10 m along the 8 km route shown in Figure 5.9. Unfortunately, as our INS system is not reliable in urban environments, we were not able to automatically generate training data for certain sections of the route. We therefore only trained places on the 6 segments illustrated in the figure, which amounts to approximately 5.5 km. We used 15 datasets for training and 3 for testing. Example images of the map and live runs is shown in Figure 5.11. Again, we wish to stress the selection of the live runs was hand-chosen to be the most challenging datasets for localisation, owing to their extreme differences in appearance from the map.

The learning phase took approximately 120 minutes per place, but were run as 5



Map image: clear and sunny; noon. | Test image: dark, night. | Test image: sunny with shadows. | Test image: clear and sunny; noon.

Figure 5.11: Example test images used in our Oxford localisation experiments. These were chosen due to their large visual variability.

separate processes, reducing the effective training time to 24 minutes. As each place is represented by a collection of SVM classifiers, the memory footprint is quite low at approximately 5 MB per place.

5.4.2 Feature Matching Experiments

The purpose of this section is to contrast the matching performance of the scene signatures against our point-feature system, in the absence of any geometric checks or motion-consistency checks that take place in the localisation pipeline. In other words, we wish to isolate the front-end of the system to see what matching potential is possible across extreme lighting and weather conditions.

For each INS-defined place in our training data, we took the corresponding groundtruth location of each test image so that the viewpoints of all images for every place are as similar as possible. By ensuring that the viewpoints of the test images and map image are well aligned, we can define a successful match as one in which the feature locations in both the live image and map image are in approximately the same location in image space. In other words, if we have a feature defined in the map image, $\mathbf{y}_m^j$, we would expect that the corresponding measurement in the live frame, $\mathbf{y}_l^j$, would be close in image space: $||\mathbf{y}_m^j - \mathbf{y}_l^j||_2 < \delta$, since we know that the transformation between the live and map frame is close to identity, by construction. In the following experiments, we defined $\delta = 15$ pixels. The same criteria applied

to our point-feature system.

5.4.2.1 Begbroke

Figure 5.12 shows the number of feature matches per place around Begbroke for both our scene signatures and the point-feature system. As expected, we see that we are able to achieve correspondences across all appearance conditions with the scene signatures, but not with point features. In particular, point features fail to find enough matches for foggy and evening runs, due to motion blur, lack of texture, and environmental changes (e.g., snow on the road and buildings).

Figure 5.13 shows heat maps of the locations in the image where matches are most likely to occur. To generate these heat maps, we simply added a count of +1 to each pixel contained within one of the matched shapes and averaged over the five test images for each place. Thus, we can compute an average distribution in image space for each place, as well as an average over the entire dataset. As expected, most of the matches occur in the far field, where we typically see distinctive structure, such as buildings and trees. As will be shown shortly, the Oxford datasets present much more interesting structure in the heat maps, since there are more distinctive man-made objects in the environment, like traffic lights, stop signs, and road markings.

5.4.2.2 Oxford

Figure 5.14 show the feature matches for the Oxford datasets using scene signatures and point features. It should be reiterated that not all the live runs followed the same trajectory as the map, so matches are only shown against the segments in common with the map (see Figure 5.9).

Once again, the results show that using scene signatures is much more robust and we are able to match against all datasets for all places, while the point-feature approach fails over a number of places, especially for the nighttime dataset. This

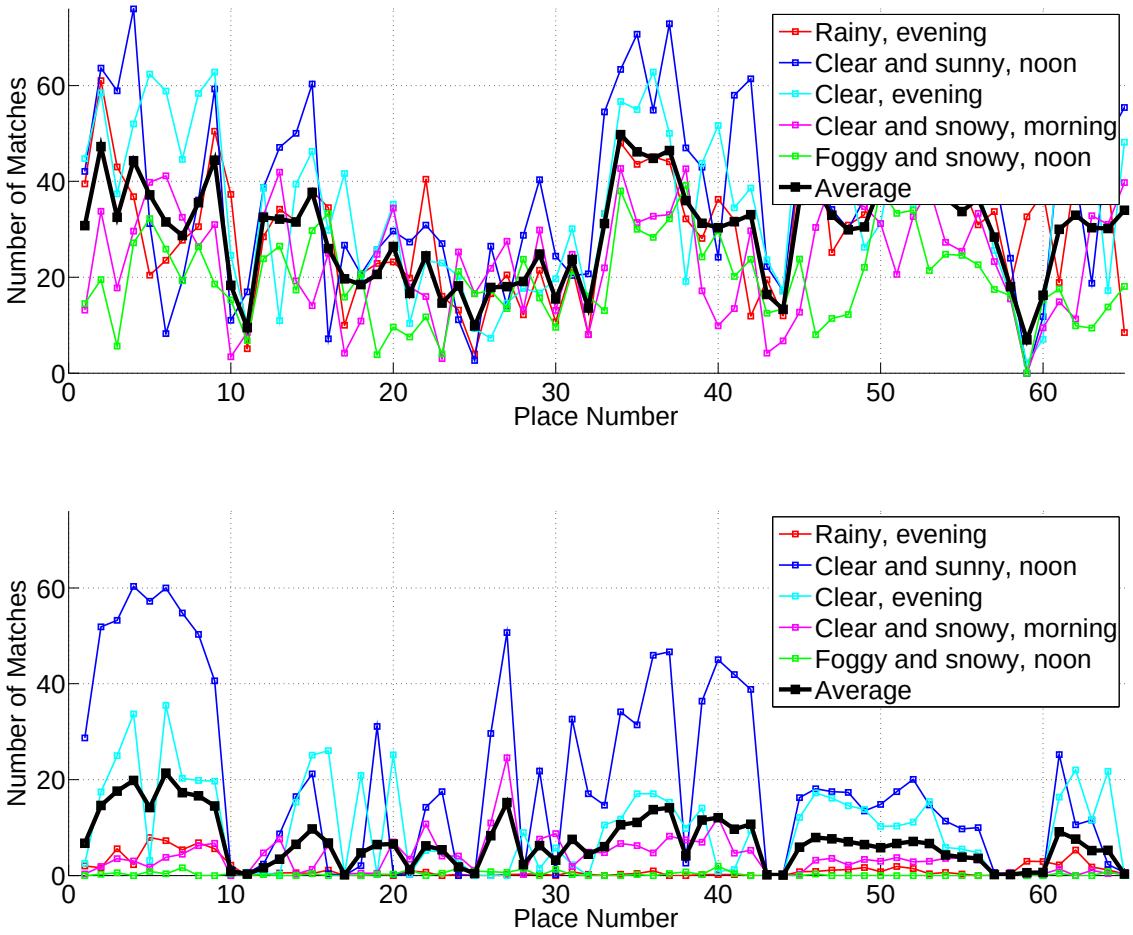
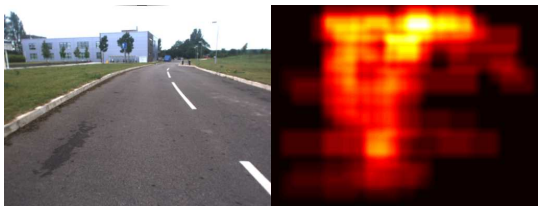
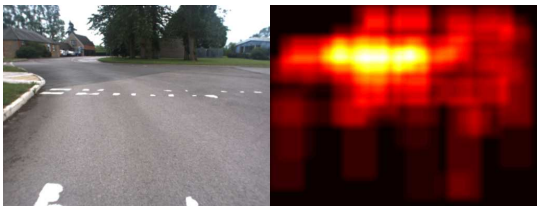


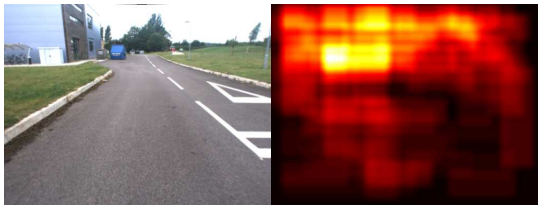
Figure 5.12: Feature matches for each place in our Begbroke dataset using scene signatures (top) and point features (bottom). Places were defined as 10 m segments along the reference trajectory shown in Figure 5.9. In this experiment, we used groundtruth aligned images at each place and performed feature matching against the map image for each respective live run. The results show that using scene signatures, we are able to match under all appearance conditions; not the case for the point-feature counterpart, which fails for the evening and foggy runs.



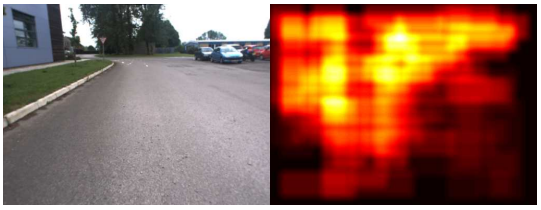
Map image (left) and feature distribution (right).



Map image (left) and feature distribution (right).

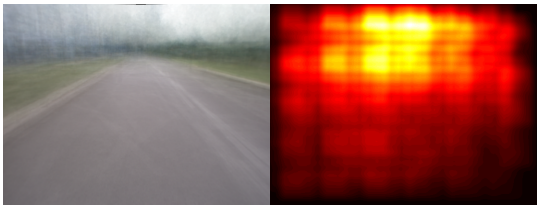


Map image (left) and feature distribution (right).



Map image (left) and feature distribution (right).

Map image (left) and feature distribution (right).



Average map image (left) and average feature distribution (right).

Figure 5.13: Sample feature distributions per place, represented by heat maps. Each pair of images has the map image at a particular place (left) and the average heat map at that place (right), which was computed by adding a count of +1 for each pixel within a detection box and averaging over all detections. The bottom right figure shows the average place and average heat map over the entire dataset. Note that most of the detections are made on the upper half of the image, which is where we typically see distant buildings, trees, and distinctive structure.

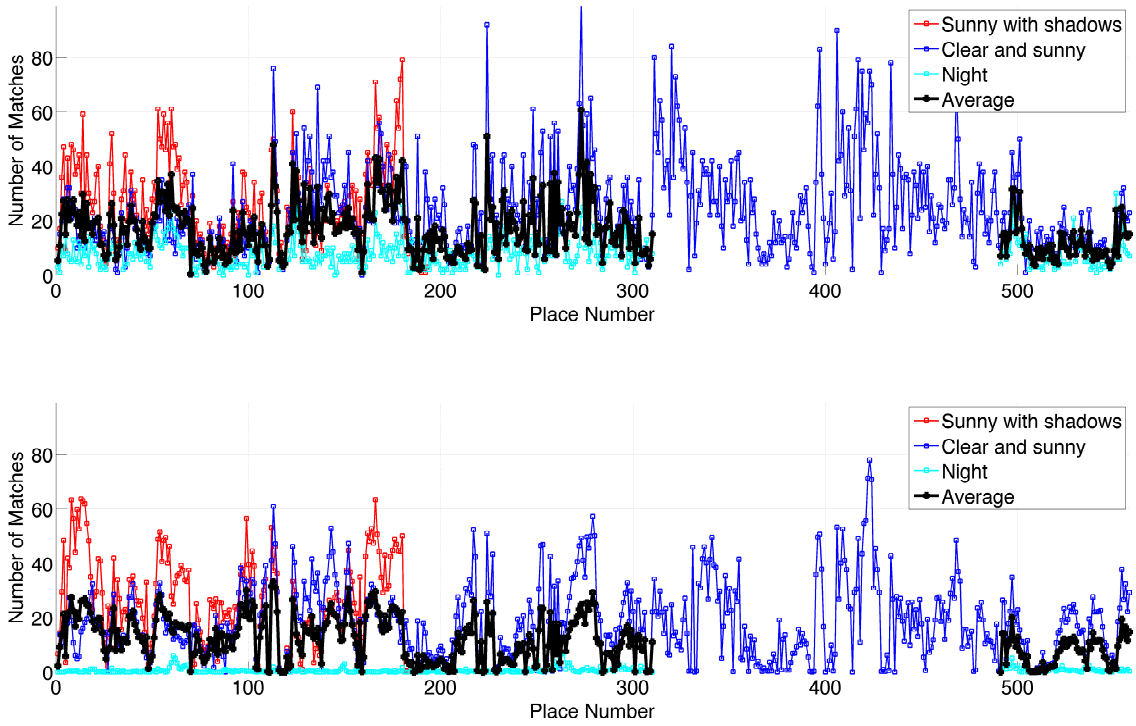
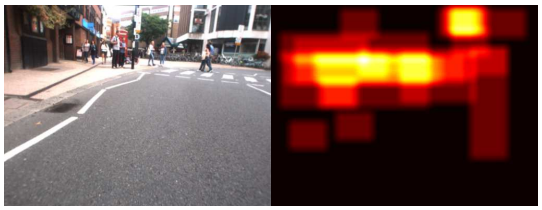


Figure 5.14: Feature matches for each place in our Oxford dataset using scene signatures (top) and our point features (bottom). Note that gaps in the data do not represent failures, they are simply regions where a live run did not intersect with the map. Failures are when the feature counts are at zero. Places were defined as 10 m segments along the reference trajectory shown in Figure 5.9. In this experiment, we used groundtruth aligned images at each place and performed feature matching against the map image for each respective live run. The results show that using scene signatures, we are able to match under all appearance conditions, which is not the case for the point-feature counterpart, which fails for the nighttime run (the flat cyan line for the point feature approach).

Map image (left) and feature distribution (right).

Map image (left) and feature distribution (right).



Map image (left) and feature distribution (right).

Map image (left) and feature distribution (right).

Map image (left) and feature distribution (right).

Average map image (left) and average feature distribution (right).

Figure 5.15: Sample feature distributions per place, represented by heat maps. Each pair of images has the map image at a particular place (left) and the average heat map at that place (right), which was computed by adding a count of +1 for each pixel within a detection box and averaging over all detections. The bottom right figure shows the average place and average heat map over the entire dataset. These heat maps show more interesting structure than the Begbroke datasets, as we see density around traffic lights and road markings. This is the reason the average heat map has a bias towards the left side of the image.

was the most challenging dataset as there is extreme motion blur and very little detail in the images.

5.4.3 Localisation Experiments

In this section, we compare our weak-localiser approach to a point-feature-based system for the task of localisation. The goal of these experiments is to show that we can use scene signatures and a weak localiser to produce metric estimates even with some of the most challenging appearance changes. In order to ensure repeatable results, the processing was done offline to control the rate of localisation updates. We gave the the system the opportunity to localise on every 4th image, which is equivalent to saying the localiser ran at 5 Hz for a 20 Hz camera feed.

5.4.3.1 Begbroke

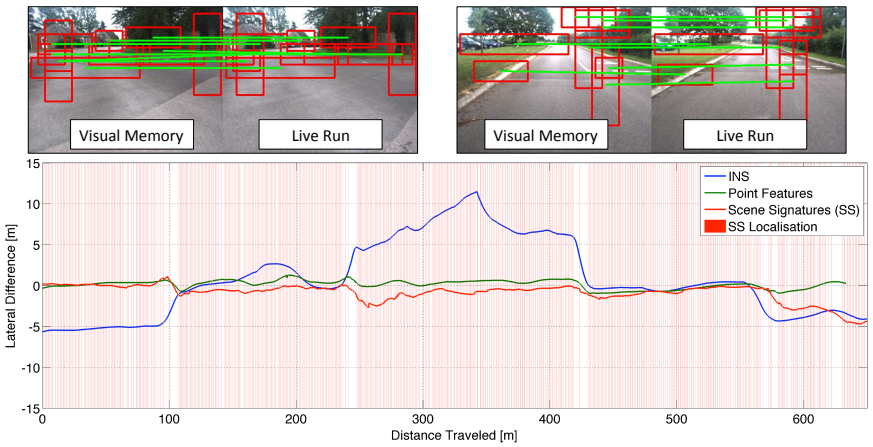
As our vehicle controller is only concerned with lateral and heading errors, we focus on these two metrics for assessing localisation performance. Figures 5.16, 5.17, 5.18, 5.19, 5.20 each show the following four plots for the 5 live runs: (i) lateral estimates, (ii) heading estimates, and (iii) speed estimates, and (iv) number of feature matches. The plots also show areas were our system was able to localise, represented by vertical red strips. Although these may appear sparse throughout the plots, recall that our system localises at a rate of about 5 Hz and integrates odometry in between the updates. Thus, there are typically gaps between each localisation update, but these are typically over small distances where the dead reckoning is quite accurate. Some sample images of the feature matches have also been provided to give a more visual interoperation of the system's performance (see Figure 5.23).

A localisation failure was defined as travelling blind on odometry for more than 20 m (i.e., 2 segments in our case). If a failure occurred, we would reset the system using the INS. This criteria applied to both systems. Also note that simply seeding

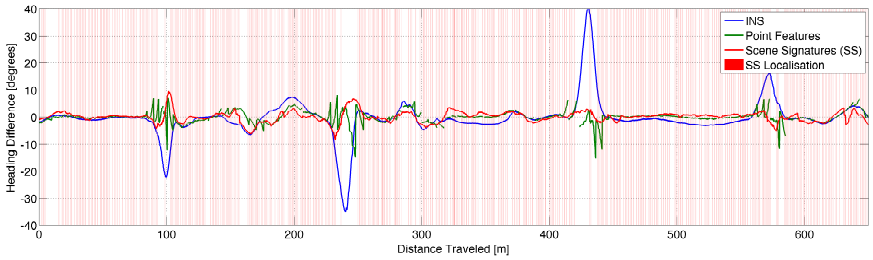
each respective system with the correct location in the map does not guarantee a successful localisation, as the appearance may still be too different, which was the case with the point-feature system in most runs.

As will become evident from the results, using the INS for groundtruth is not appropriate due to significant drift from one dataset to the next, as well as jumps that occur during the run. In order to aid the reader, it important to note that the "ideal" plot would be close to zero lateral and heading errors from the reference trajectory since we drove the same physical path (approximately). Thus, the reader should look at all plots with the expectation that the best system would have close to zero lateral/heading error. Additionally, as the number of figures can be overwhelming at first glance, summary plots of the performance are provided at the end of this section.

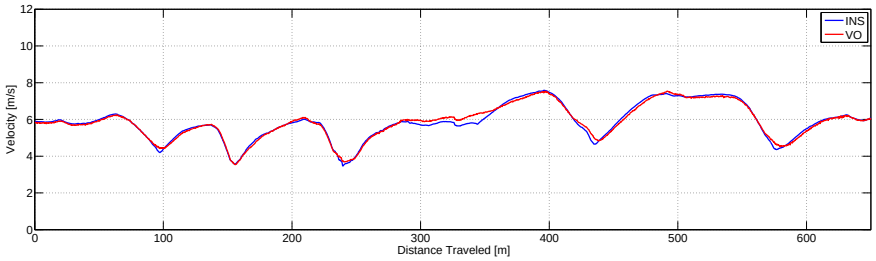
There are a number of interesting results from these plots. Firstly, one can see the scene-signature system works as well, if not better than our INS system and somewhat comparable to the point-feature system when it is working. As expected, the point-feature system was unable to localise in most of the cases where the appearance changes were drastic. Secondly, we see that there are two runs where the scene-signature system struggled: 5.17, 5.18. This is most likely due to a lower number of feature matches during those runs and, more significantly, poor dead reckoning. Referring to the VO outputs for each run (Figures 5.17 and 5.18), we see that the two runs where the system struggled correspond to the two runs where the VO output was very noisy. Since the localisation system depends heavily on a strong motion prior, the runs where the motion priors were noisy most likely corresponded to suboptimal localisation estimates. To test this hypothesis, we swapped the VO output with the INS incremental transformations. These plots are shown in Figure 5.21 and Figure 5.22 and confirm the hypothesis. Although the average number of feature matches during those runs were lower due to the extreme appearance



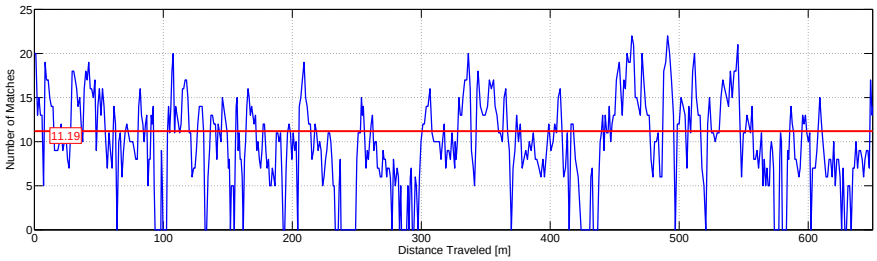
Lateral estimates for a sunny visual memory vs. a sunny run.



Heading estimates for a sunny visual memory vs. a sunny run

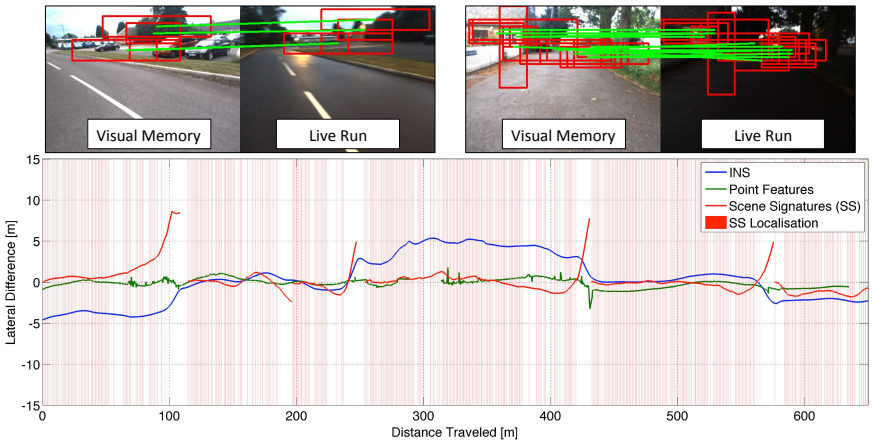


Live VO profile against groundtruth.

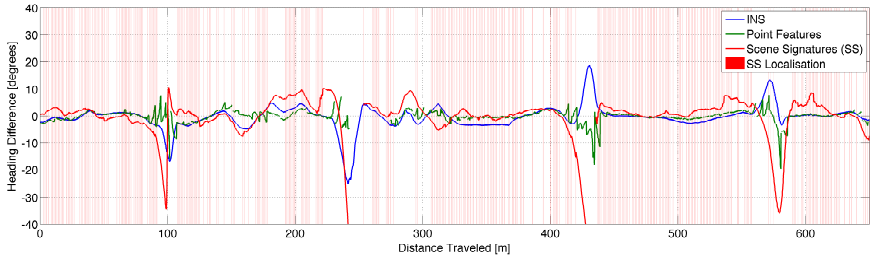


Number of feature matches.

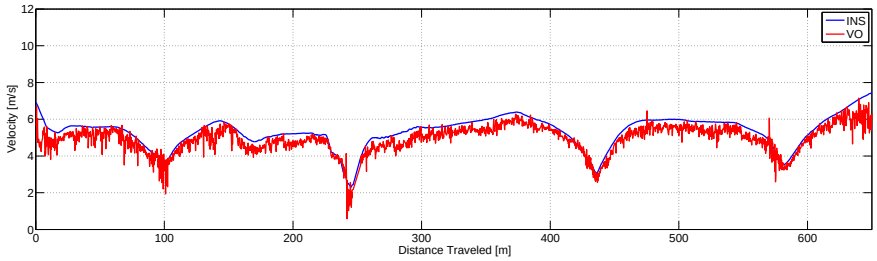
Figure 5.16: Localisation results for the sunny afternoon run (Begbroke). The scene-signature system (red) performed comparably with the point-feature system (green) and outperformed the INS (blue), which drifted quite significantly from one dataset to the next. The top two plots represent the key error terms fed into our vehicle controller.



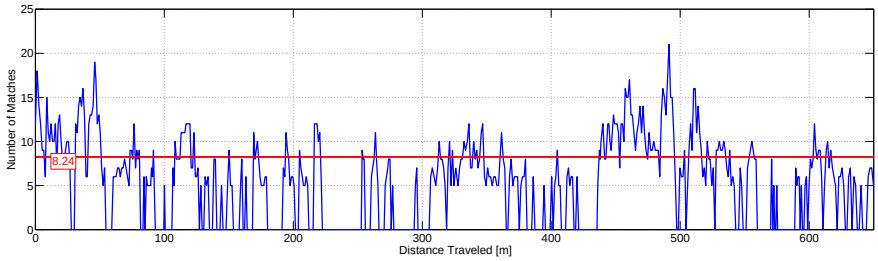
Lateral estimates for a sunny visual memory vs. a clear, evening run.



Heading estimates for a sunny visual memory vs. a clear, evening run

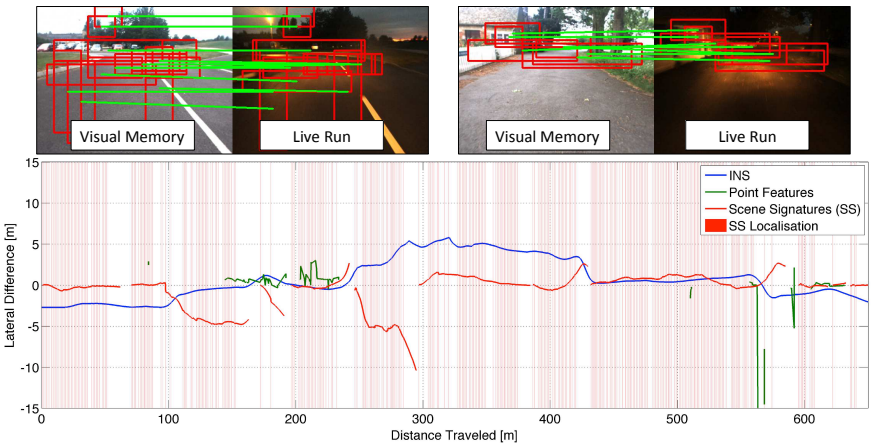


Live VO profile against groundtruth.

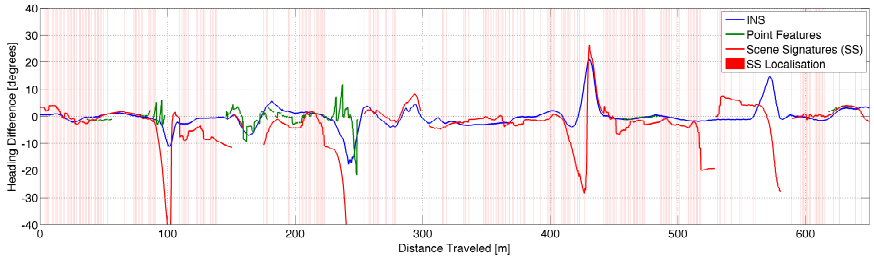


Number of feature matches.

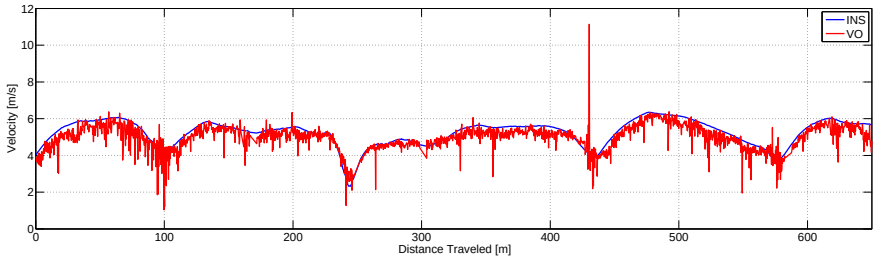
Figure 5.17: Localisation results for the clear, evening run (Begbroke). The scene-signature system experienced greater lateral deviations in this run due to the noisy VO output (third subfigure from the top). Note that when the VO was substituted for the INS relative poses, the accuracy significantly improved (shown later in Figure 5.21).



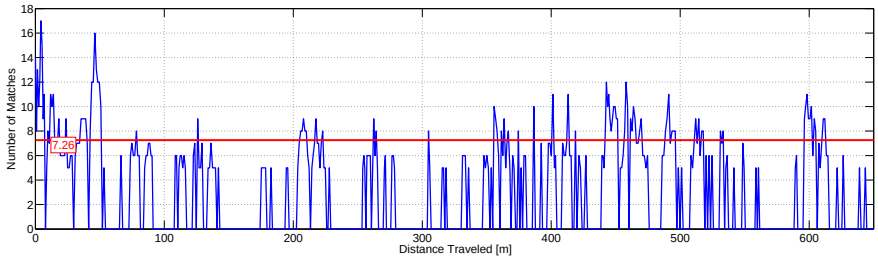
Lateral estimates for a sunny visual memory vs. a rainy, evening run.



Heading estimates for a sunny visual memory vs. a rainy, evening run

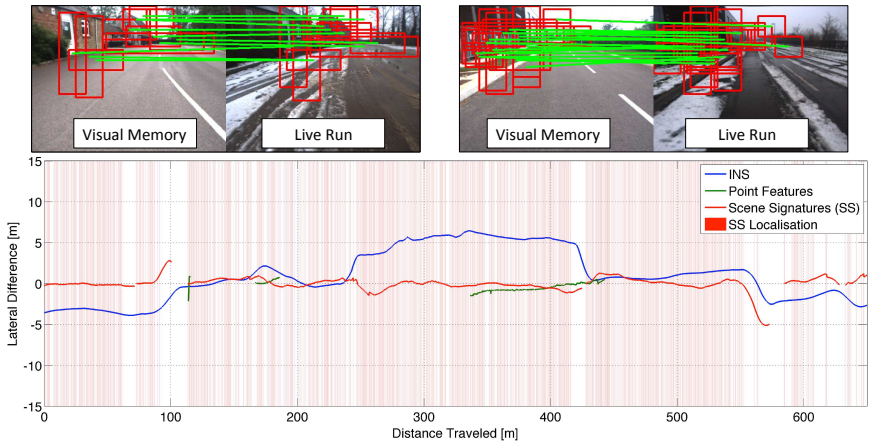


Live VO profile against groundtruth.

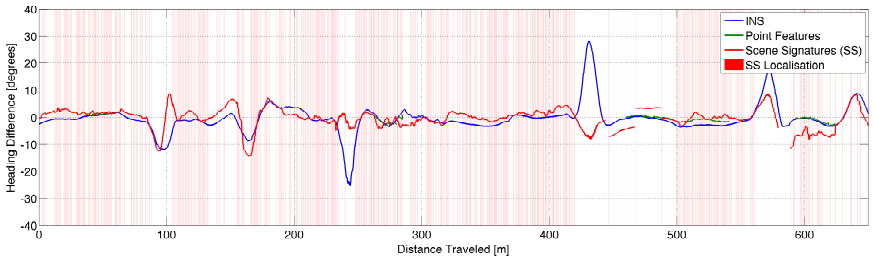


Number of feature matches.

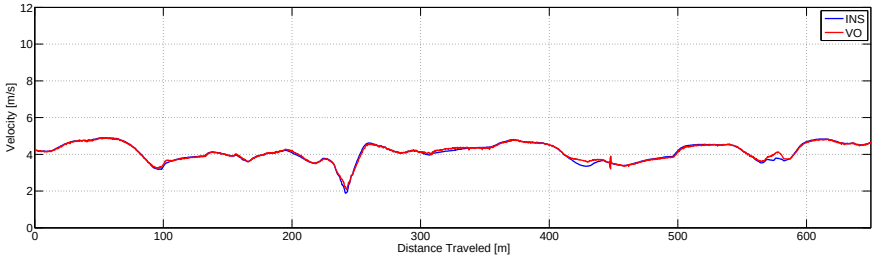
Figure 5.18: Localisation results for the rainy, evening run (Begbroke). Similar to the other evening run, the VO output was very noisy and as a result, the localisation performance suffered. However, when the VO was substituted fro the INS relative poses, we observed a significant improvement in accuracy (shown later in Figure 5.22). We did, however, outperform the point-feature system (green), which was unable to cope with such extreme appearance changes.



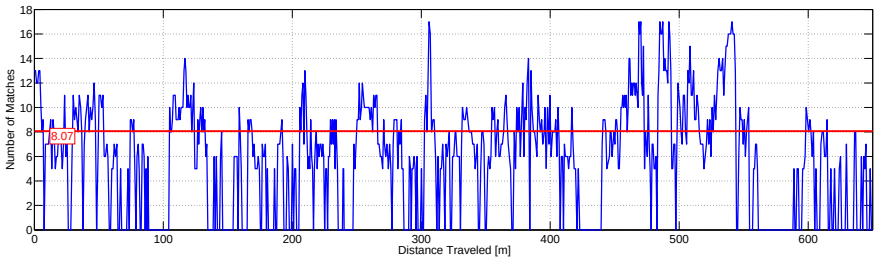
Lateral estimates for a sunny visual memory vs. a clear, snowy run.



Heading estimates for a sunny visual memory vs. a clear, snowy run

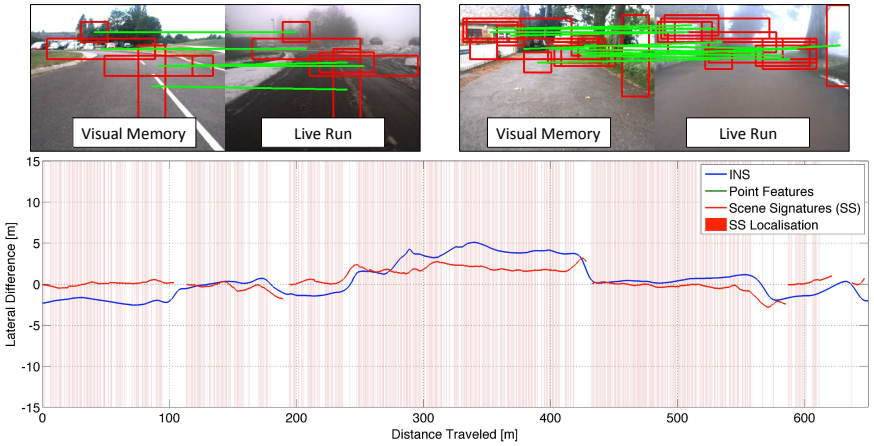


Live VO profile against groundtruth.

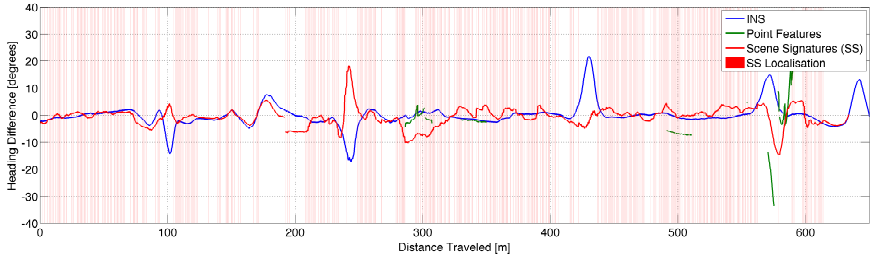


Number of feature matches.

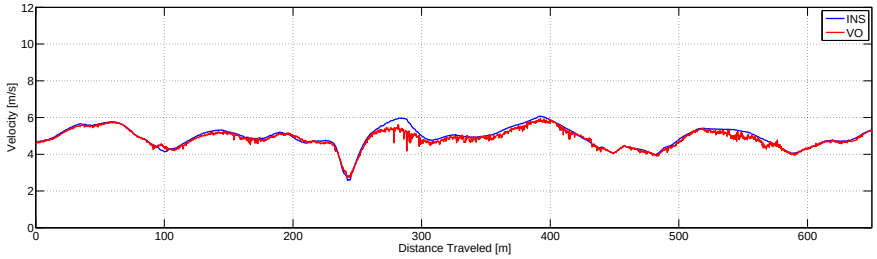
Figure 5.19: Localisation results for the a clear, snowy run (Begbroke). As the VO output was quite good for this run, even with significant appearance differences to the map, the scene-signature system (red) was able to successfully localise, whereas the point-feature system failed over most of the run (green).



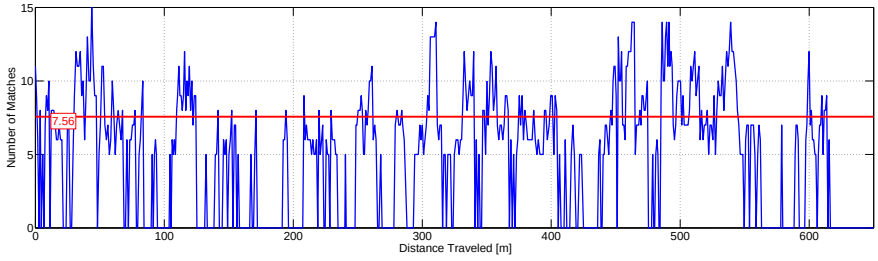
Lateral estimates for a sunny visual memory vs. a misty, snow run.



Heading estimates for a sunny visual memory vs. a misty, snow run

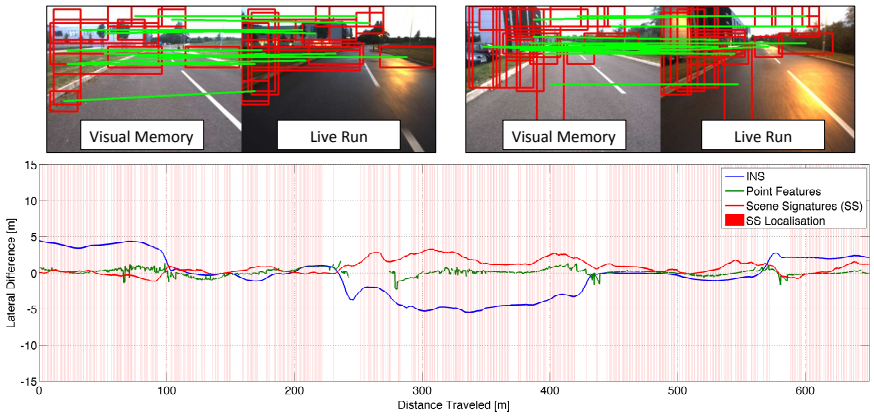


Live VO profile against groundtruth.

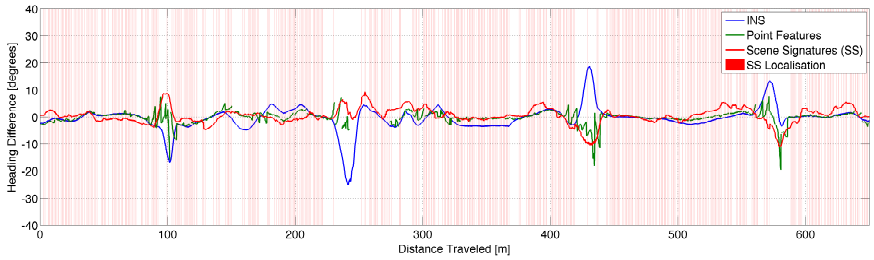


Number of feature matches.

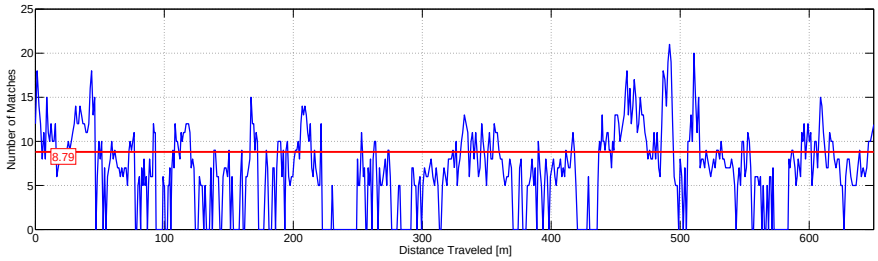
Figure 5.20: Localisation results for the misty, snow run (Begbroke). Another example where the scene-signature system (red) was able to localise over the entire run despite significant appearance differences, while the point-feature system failed (green).



Lateral estimates for a sunny visual memory vs. a clear, evening run.

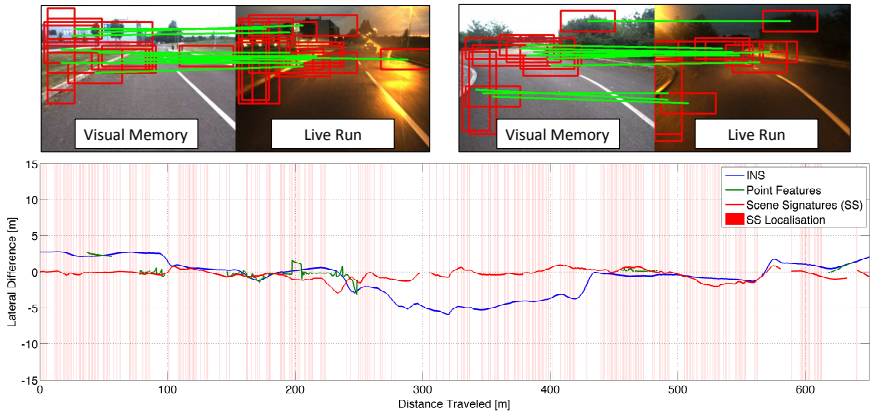


Heading estimates for a sunny visual memory vs. a clear, evening run

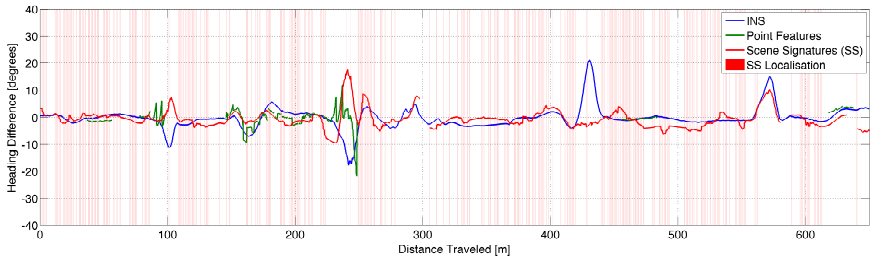


Number of feature matches.

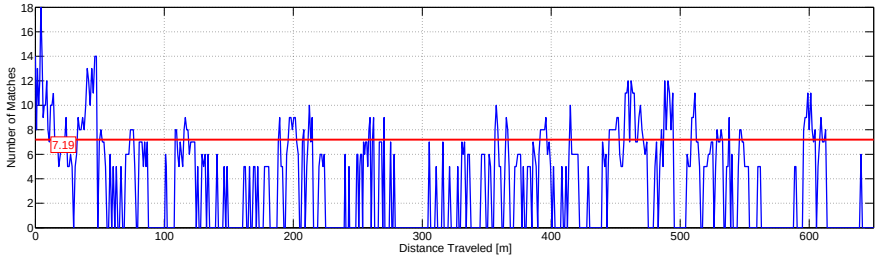
Figure 5.21: Localisation results for the clear, evening run using the INS for dead reckoning (Begbroke). By replacing the noisy VO relative poses for this run with the smoother INS measurements, we see that the scene-signature system is able to localise with a comparable accuracy to both systems. Again, this is due to the fact that the system relies on a strong motion prior for localisation, so if this motion prior is noisy, the estimates will suffer.



Lateral estimates for a sunny visual memory vs. a rainy, evening run.

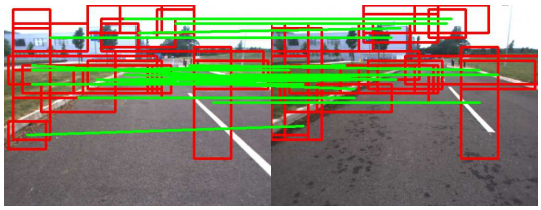


Heading estimates for a sunny visual memory vs. a rainy, evening run

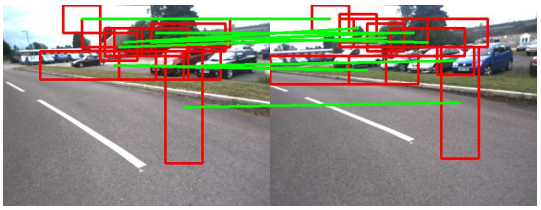


Number of feature matches.

Figure 5.22: Localisation results for the rainy, evening run using the INS for dead reckoning (Begbroke). Another example where accurate dead reckoning improved the system's localisation performance despite drastic differences in appearance. The point-feature system was unable to cope and failed over a majority of the run.

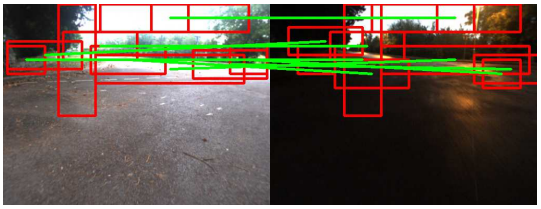


Sunny visual memory and a sunny live run.

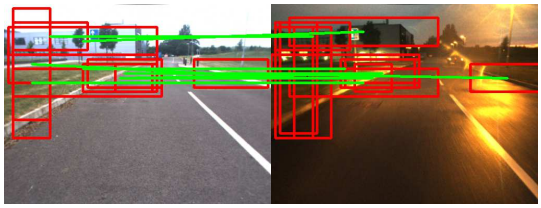


Sunny visual memory and a sunny live run.

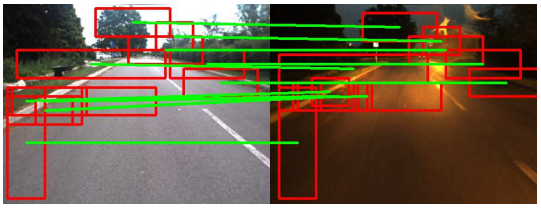
Sunny visual memory and a clear, evening run.



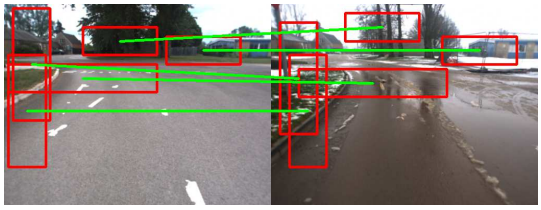
Sunny visual memory and a clear, evening run.



Sunny visual memory and a rainy, evening run.



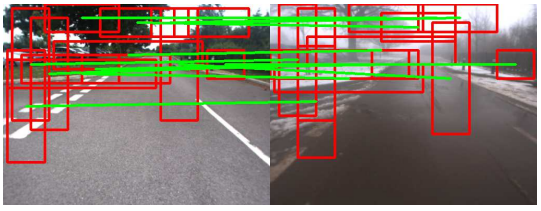
Sunny visual memory and a rainy, evening run.



Sunny visual memory and a snowy run.

Sunny visual memory and a snowy run.

Sunny visual memory and a misty, snow run.



Sunny visual memory and a misty, snow run.

Figure 5.23: Scene signature feature matches for all Begbroke runs. Each row represents a particular dataset. Top row is the sunny run, the second row is the evening run, the third row is the rainy evening run, the fourth row is the snow run, and the last row is the misty run. The left image in each pair of images is from the sunny visual memory and the right image is the live run.

changes between the map image and the live run, we see that it was the poor dead reckoning estimates that contributed to the significant error.

This highlights the importance of accurate egomotion estimation for the task of localisation, which was the goal with distraction suppression in Chapter 3. In the next chapter, we will show how distraction suppression can improve localisation estimates in urban environments.

5.4.3.2 Oxford

Figures 5.24, 5.26, and 5.27 show the lateral errors, heading errors, velocity profile, and number of feature matches for the three Oxford runs, which included a nighttime run, a sunny and shadowy run, and a clear sunny run. To reiterate, as each live run took a different path from the reference route, or because there were areas where we did not have training data, errors are only reported on the segments indicated in Figure 5.9.

The scene-signature system struggled during the nighttime run (Figure 5.24) because of the extreme lack of any texture or detail in the images (some sample feature matches are shown in Figure 5.28). This resulted in poor VO estimates and noisy feature matches. As a result, the system drifts in a number of locations, as indicated in the plots. Figure 5.25 shows the localisation results using the INS for a pose source instead of the VO. However, as the INS is quite noisy for this run, no improvement was observed. We wish to stress that the point-feature system was unable to work under these conditions, further demonstrating the robustness of the scene-signature approach.

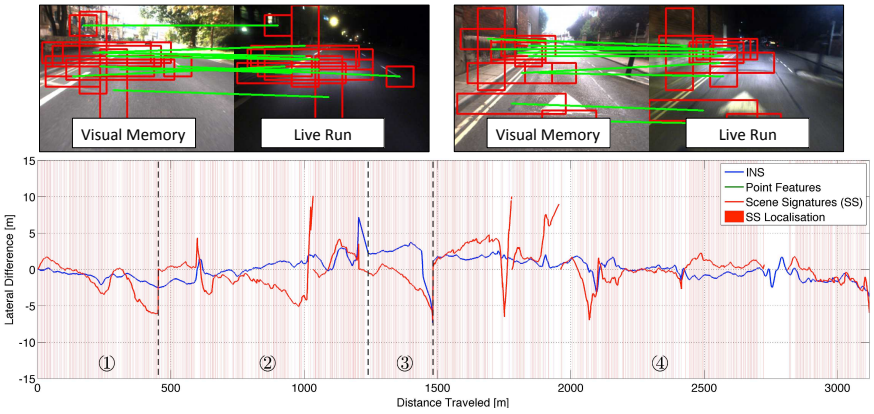
The shadowy daytime run (Figure 5.26) went well, producing estimates commensurate to the point-feature system and better than the INS. As mentioned earlier, the point-feature system works well when the appearance conditions are similar, meaning that it serves as better groundtruth for the like-against-like runs, owing

to the unreliability of the INS. The clear daytime run (Figure 5.27) shows similar performance again to the point-feature system. As before, we provide some sample images of the feature matches during each run, which is provided in Figure 5.28.

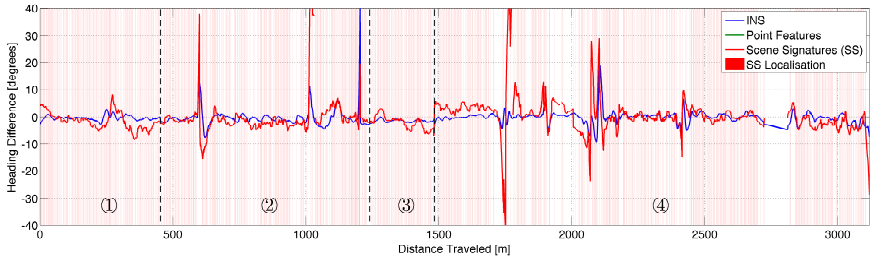
5.5 Summary

Borrowing from Chapter 4, Figure 5.29 provide a nice summary of the performance over all datasets and show the likelihood of traveling on dead reckoning for our approach and the point-feature approach. It is clear that although the proposed system may not be as accurate as a point-feature system in all cases, it is significantly more robust to motion blur, lighting changes, and weather conditions. Additionally, for like-against-like runs (e.g., sunny live run against sunny visual memory), the results indicate the scene-signature approach does perform comparably to the point-feature system. For other runs involving drastic appearance changes, the point-feature system simply failed, whereas the scene-signature approach showed comparable performance to the INS.

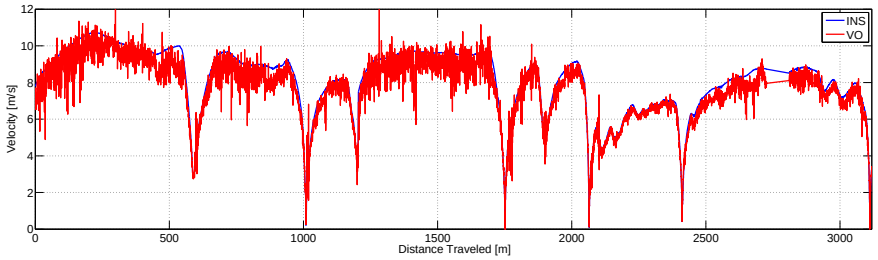
It should be emphasised that the conclusion of these results is not that point features are bad. On the contrary, they work very well for a number of tasks, such as VO. It is simply for the task of metric localisation admits significant appearance change that they struggle, which is to be expected by their very design. Point-feature systems attempt to match low-level primitives, which can look completely different from winter to summer. In contrast, scene signatures are designed to find mid-level patches that represent distinctive visual elements that can be associated across a range of appearances.



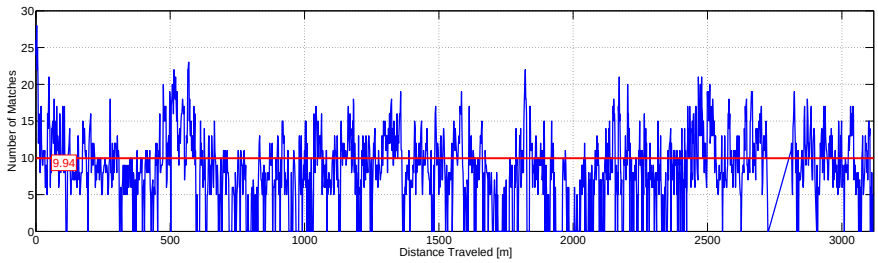
Lateral estimates for a sunny visual memory vs. a dark, night run.



Heading estimates for a sunny visual memory vs. a dark, night run.

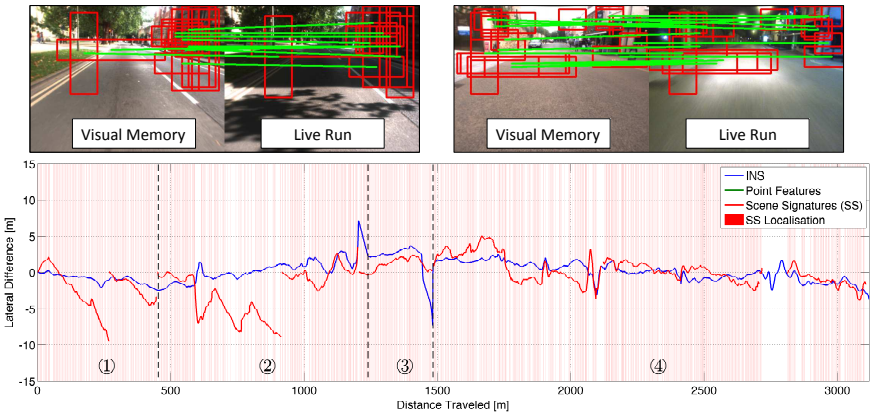


Live VO profile against groundtruth.

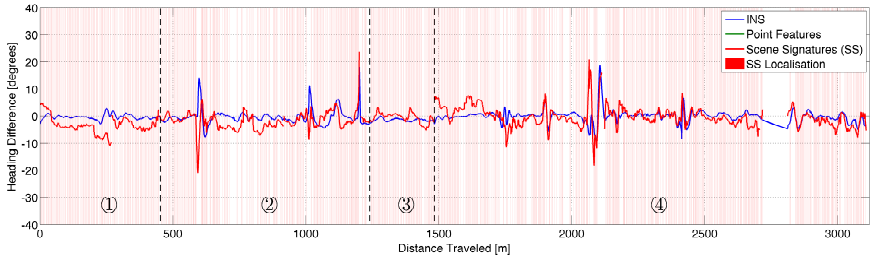


Number of feature matches.

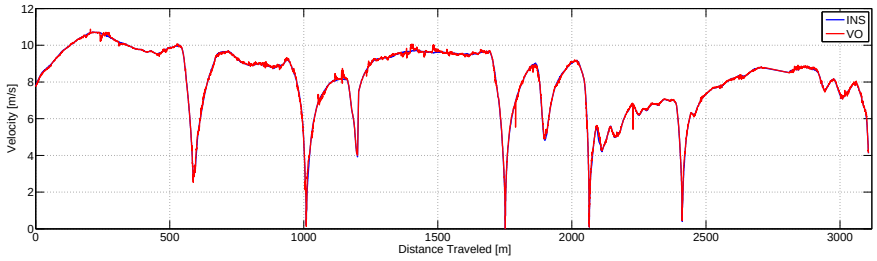
Figure 5.24: Localisation results for the night run through segments 1-4 (Oxford). The localisation performance for this run was poor due to extremely low-light conditions and the lack of texture in the images. However, we note that the estimates seem commensurate with the INS and that the point-feature system failed on this run.



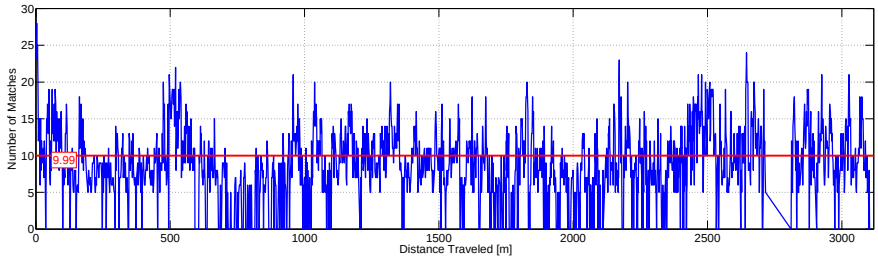
Lateral estimates for a sunny visual memory vs. a dark, night run.



Heading estimates for a sunny visual memory vs. a dark, night run.

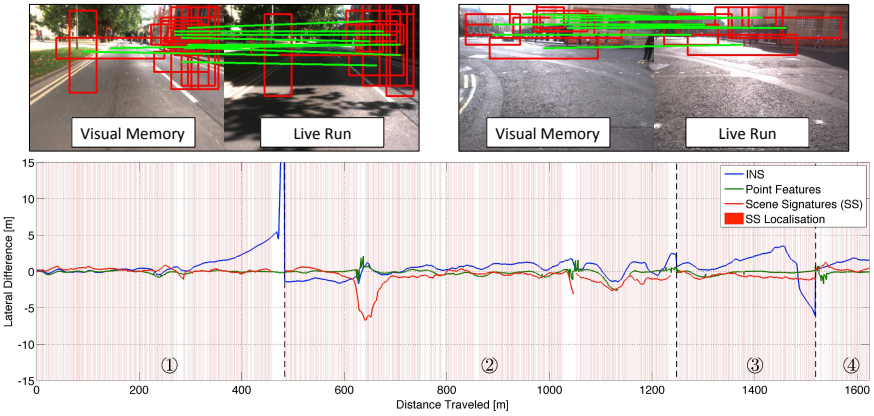


Live VO profile against groundtruth.

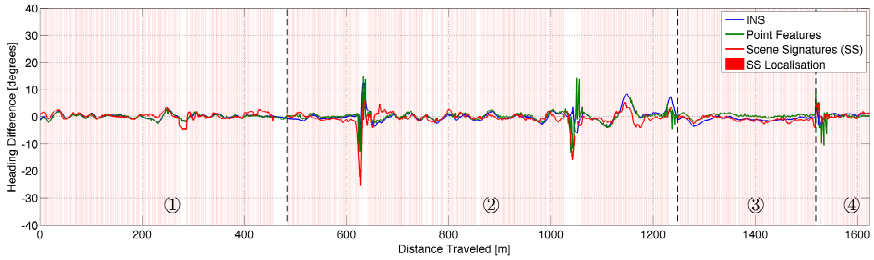


Number of feature matches.

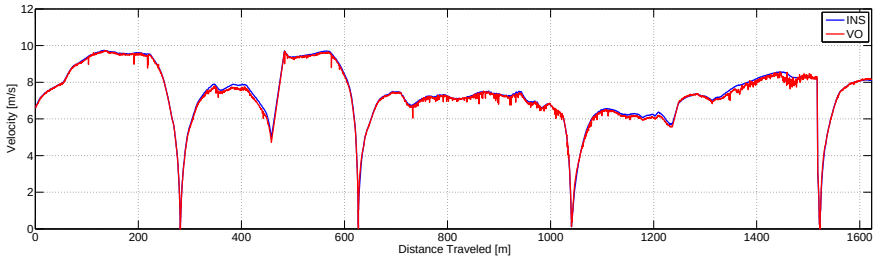
Figure 5.25: Localisation results for the night run through segments 1-4 using the INS (Oxford). Unfortunately, we found that swapping out the VO with the INS offered no improvement in performance as the INS drifted quite substantially during this run. Additionally, the number of feature matches was quite low owing to the severe, low-light conditions.



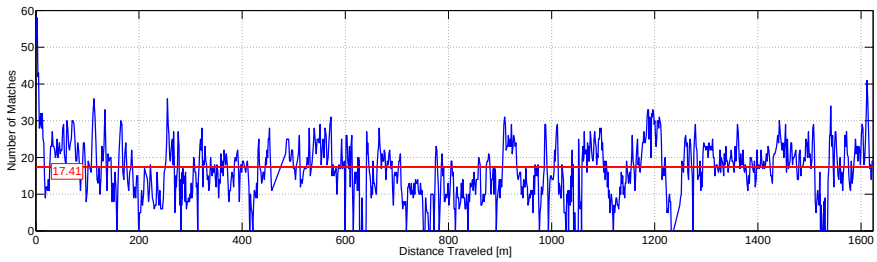
Lateral estimates for a sunny visual memory vs. a sunny run.



Heading estimates for a sunny visual memory vs. a sunny run.

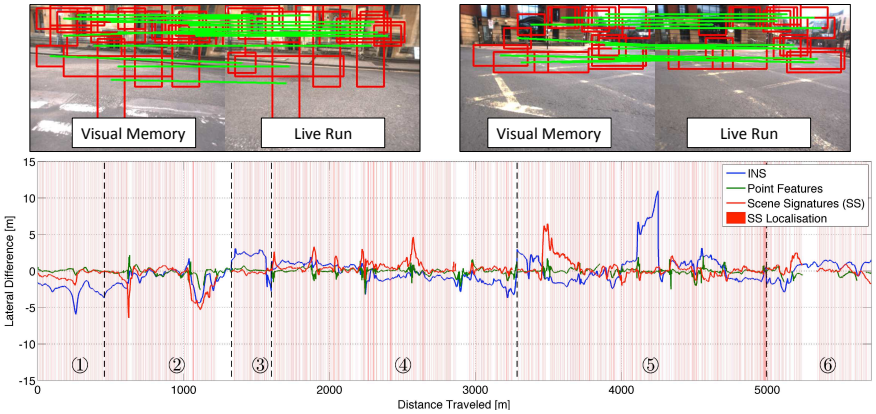


Live VO profile against groundtruth.

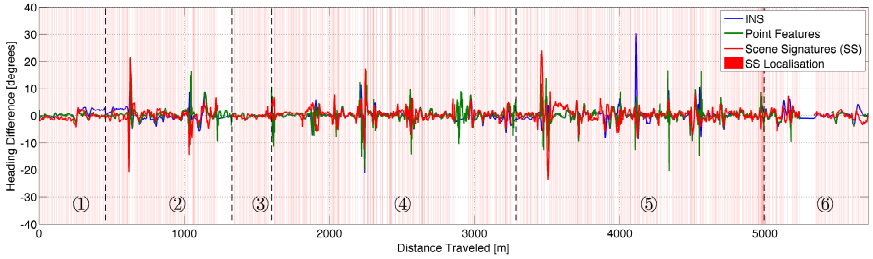


Number of feature matches.

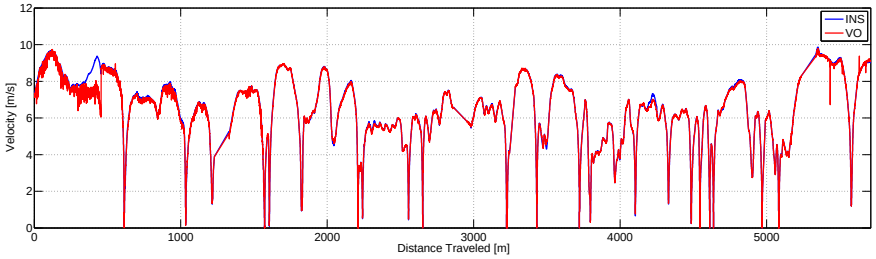
Figure 5.26: Localisation results for a sunny, shadowy run through segments 1-4 (Oxford). The scene-signature system performed well on this run, with one dip around the 600 m mark, which corresponded to a localisation drift during a turn. The rest of the run followed the point-feature estimate closely, which is more trustworthy as groundtruth than the INS in these runs, since the conditions were similar.



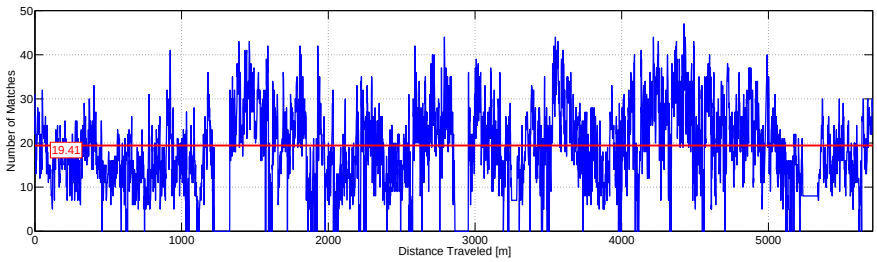
Lateral estimates for a sunny visual memory vs. a sunny run.



Heading estimates for a sunny visual memory vs. a sunny run.

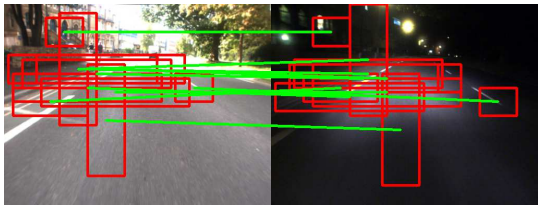


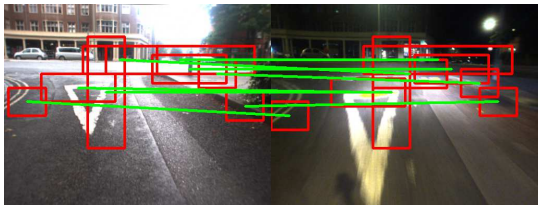
Live VO profile against groundtruth.

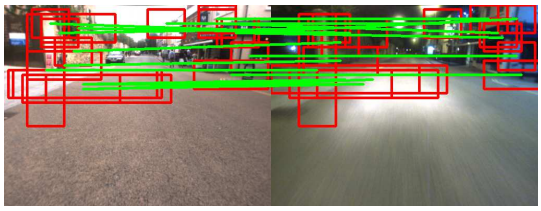


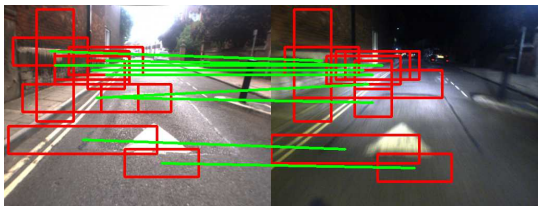
Number of feature matches.

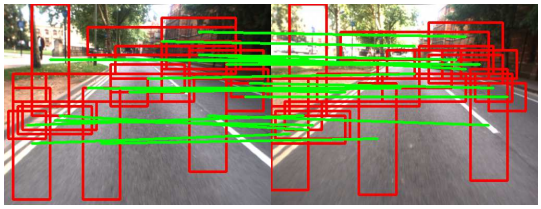
Figure 5.27: Localisation results for a sunny run through segments 1-6 (Oxford). This was a longer run and again, we see that the scene-signature system performs as well, if not better than the INS, and commensurate in a number of locations with the point-feature system, which performs well because the appearance conditions are similar.


Sunny visual memory and a nighttime live run.

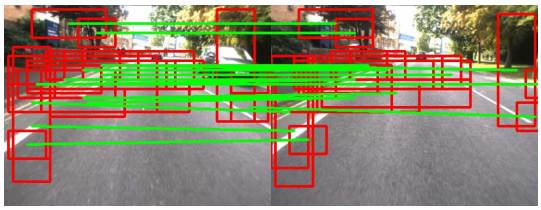

Sunny visual memory and a nighttime live run.


Sunny visual memory and a nighttime live run.

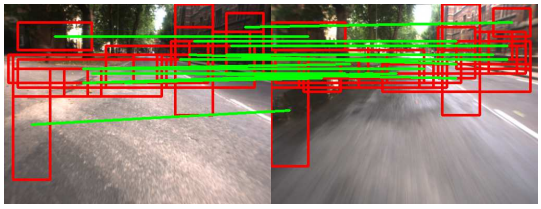

Sunny visual memory and a nighttime live run.

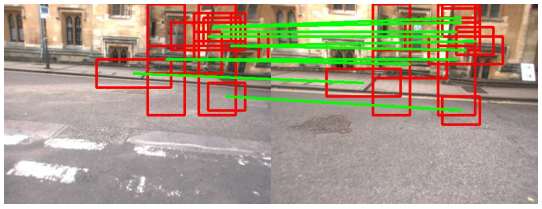

Sunny visual memory and a shadowy live run.

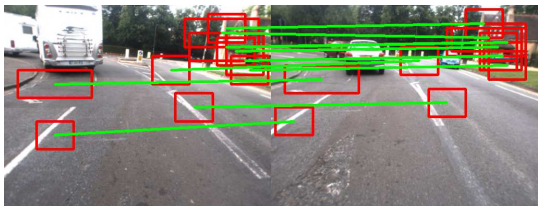

Sunny visual memory and a shadowy live run.


Sunny visual memory and a shadowy live run.


Sunny visual memory and a shadowy live run.


Sunny visual memory and a sunny live run.

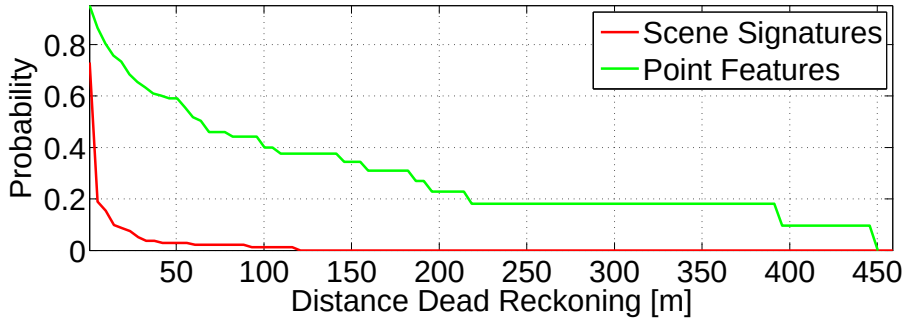

Sunny visual memory and a sunny live run.


Sunny visual memory and a sunny live run.


Sunny visual memory and a sunny live run.

Figure 5.28: Scene signature feature matches for all Oxford runs.

Begbroke



Oxford

Figure 5.29: This figure shows the likelihood of driving using only dead reckoning (i.e., the likelihood of traveling a certain distance while failing to localise). This metric provides a measure of how bad things can go when the system fails. As can be seen, the likelihood of traveling blind on odometry is significantly less using scene signatures than point features.

Chapter 6

Combined System

Before proceeding, it is worth reviewing the techniques and results that have been presented up to this point.

First, we looked at how 3D priors can be leveraged to produce distraction masks, which are useful for ignoring ephemeral objects that could degrade the performance of VO. This is important for localisation tasks because the frame-to-frame VO estimates are used for motion prediction, which can either be fused in the final output, or at the very least, used to predict the closest keyframe in the map.

Next, we examined the fragility of a point-feature-based system in environments with illumination changes and offered a simple, yet remarkably effective solution that uses a model of a black-body radiator to reduce the effects of shadows for robust localisation. We also introduced an important performance metric for a localisation system, which considers the distance traveled in between localisation failures. This is an important property because it means that even if a localiser fails infrequently, we should still be concerned with what happens during that failure. As was shown, incorporating an illumination-invariant stream into our pipeline significantly reduced the likelihood of traveling on odometry over large distances.

Although this offered an improvement to dealing with shadows, it did not address more long-term appearance changes caused by different seasons or weather condi-

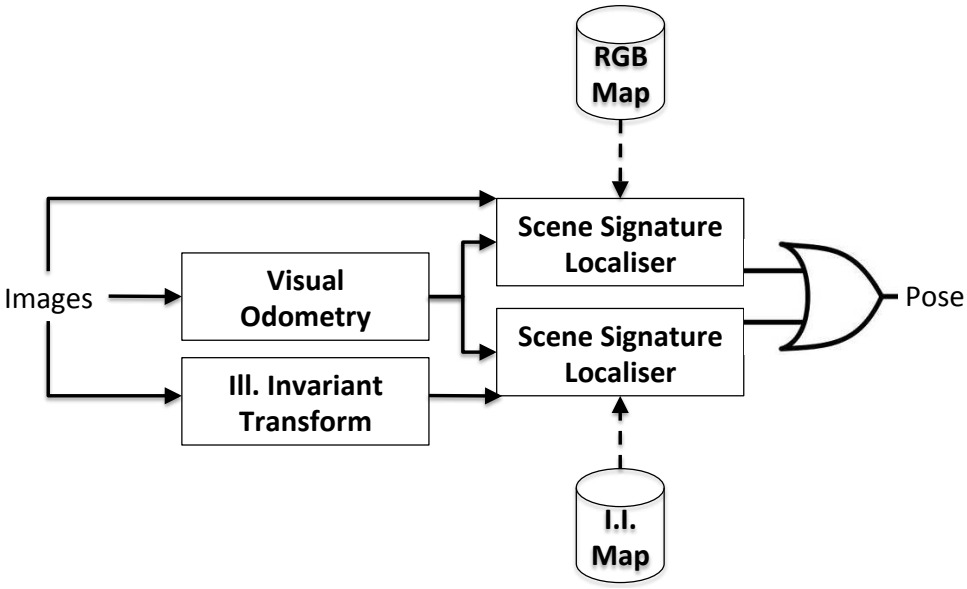
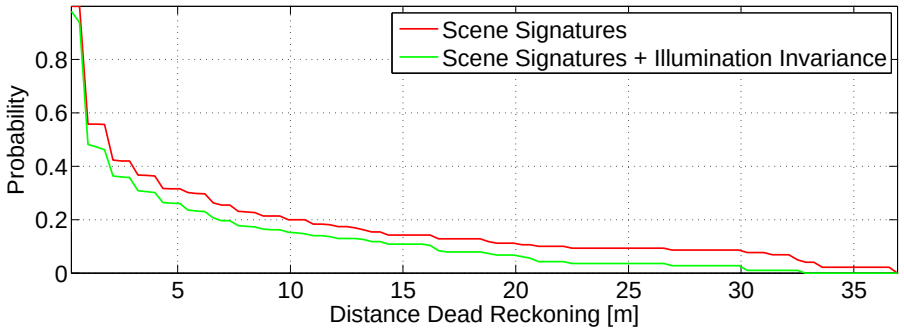


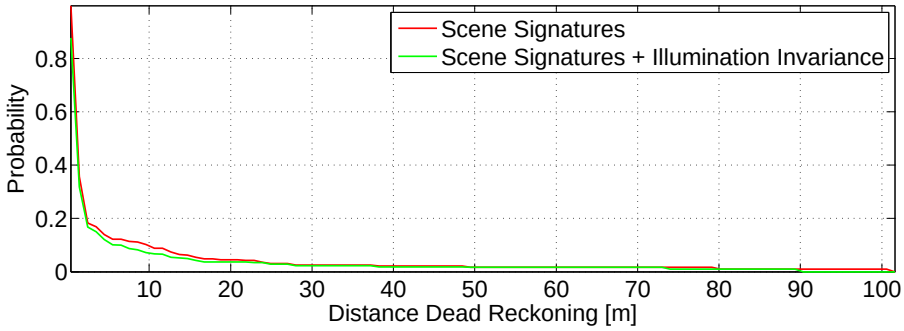
Figure 6.1: Illustration of the localisation pipeline combining two scene-signature localisers: one trained in an RGB colour space and the other trained in an illumination-invariant colour space. Note how the live image stream is fed into the VO block to provide odometry for both localisers and is also transformed to an illumination-invariant colour space for the second localiser. As was done in Chapter 4, we OR the output, with a default policy of choosing the RGB stream when available.

tions. For this, we introduced a technique that leverages image priors to learn place-dependent feature detectors that can identify distinctive, visual elements, which can be associated across extreme appearance differences. It was shown that this technique on average outperformed our INS system in terms of lateral/heading error accuracy and was more robust than our point-feature system.

In this Chapter, we draw upon previous work and analyse system performance when we combine each element (i.e., illumination invariance and distraction suppression) into our localisation framework.



Begbroke



Oxford

Figure 6.2: Probability of travelling on open-loop odometry with/without an illumination-invariant colour space.

6.1 Illumination Invariance

In Chapter 4, we presented an illumination-invariant image transformation (4.11):

$$\mathcal{I} = \log(R_G) - \alpha \log(R_R) - \beta \log(R_B), \tag{6.1}$$

where $\{R_i\}$ are the colour responses in each channel, $\{\alpha, \beta\}$ are channel coefficients, and $\{\lambda_R, \lambda_G, \lambda_B\}$ are the peak sensitivity wavelengths for each image sensor. This was a relatively inexpensive, pixel-wise image transform that provided significant improvements for our point-feature-based system. We now explore what would happen if we trained scene signatures in an illumination-invariant colour space instead of an RGB colour space. Similar to Chapter 4, we can then combine a parallel localisation output in a switching framework (see Figure 6.1).

For the sake of brevity, we omit the lateral/heading error plots from this section and focus on the metric introduced in Chapter 4 as this provides a nice summary of the data. The lateral/heading error plots can be seen in Appendix D and E for the interested reader. Figure 6.2 shows the results for the combined system (RGB + illumination invariance) versus the baseline (RGB). The results show a marginal improvement for the Begbroke datasets, but very little difference in the Oxford datasets. The most likely reason we only see a marginal improvement is that by construction, we trained the scene signatures to learn distinctive patches across a variety of appearance conditions. Thus, it is not an entirely surprising result that the illumination-invariant space offers only a slight benefit in conjunction with the RGB space.

Figure 6.3 shows some of the cases where the combined system succeeded and the baseline system failed. Note that a localisation failure does not necessarily mean that the number of matches was below the minimum required for a pose solve, but could also result when the iterative solve fails to converge within a set number of iterations due to outliers. It is interesting to note how noisy the illumination-invariant images are for the night runs. This is because the predominant external illuminates (e.g., street lights) cannot be modelled as black-body radiators, thus invalidating the technique's primary assumption. Maddern et al. (2014b) report similar findings at night using the same illumination-invariant colour space.

Although we see an improvement in terms of a reduction in distance traveled on odometry, this reduction is very small (on the order of a meter or two). Additionally, the localisation estimates from the illumination-invariant stream are very noisy and would not be very useful in practice (see Figures 6.4, and 6.5 for representative examples). These reasons seem to suggest that training scene signatures on an illumination-invariant colour space is not warranted, since the estimates are very noisy and provide little benefit.

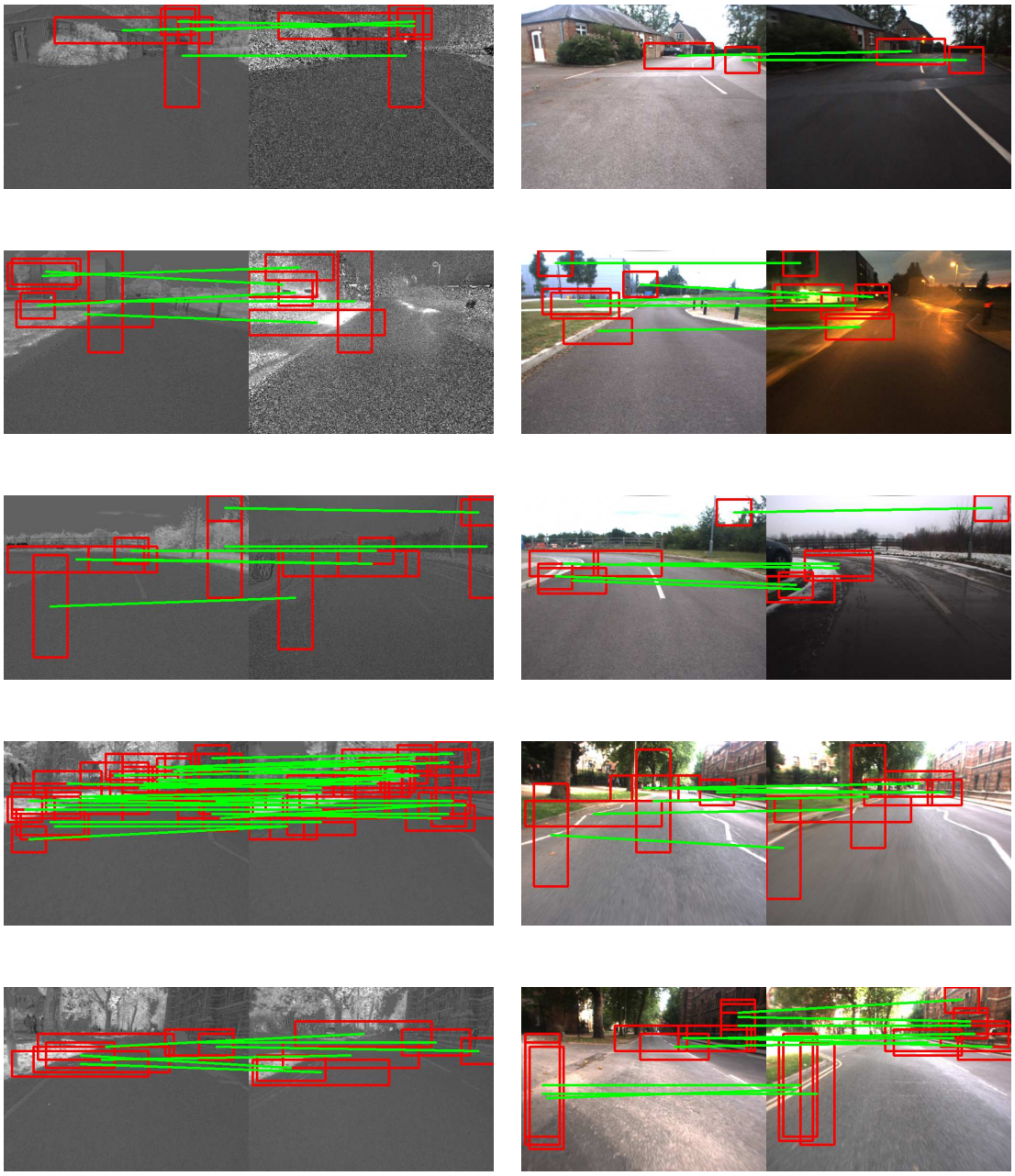
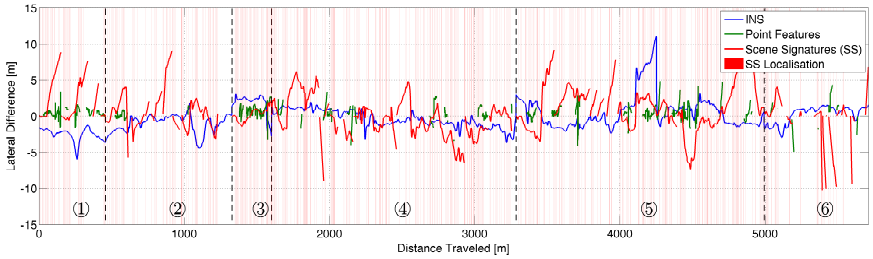
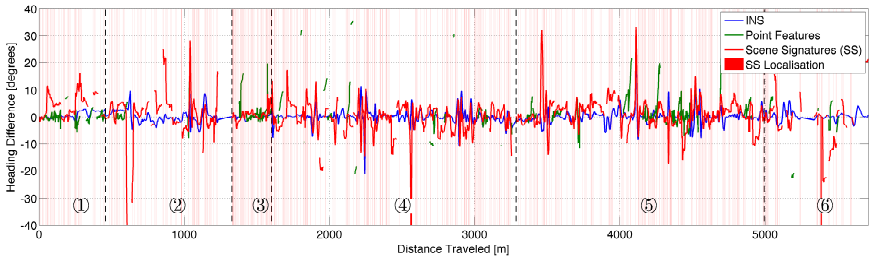


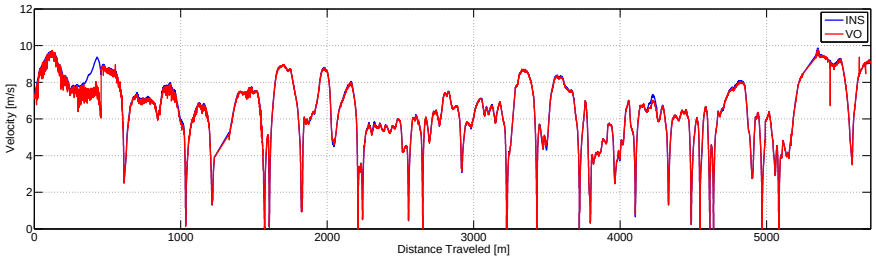
Figure 6.3: Frames where our illumination-invariant system was able to localise and our RGB system failed to localise. Each row contains the illumination-invariant and RGB counterpart, where the left image in each pair is the map and the right image in each pair is the live run. Localisation failures are flagged as either an insufficient number of matches or if the pose solve fails to converge within a set number of iterations. Note how the illumination-invariant transform adds a significant amount of noise to the images taken at night (top two rows). This is because the black-body assumption is violated at night, as street lights and car lights do not obey this law. Despite additional localised frames, we found overall that the combined system offered very little improvement over the baseline.



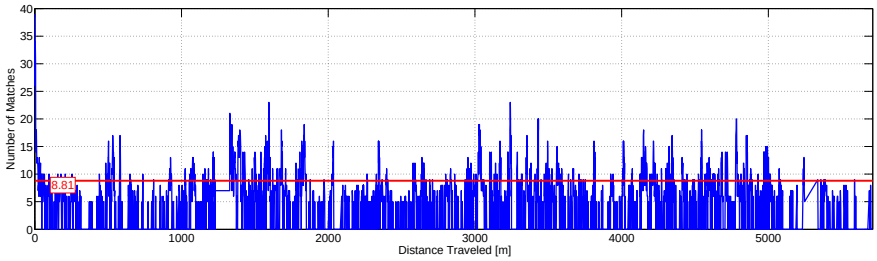
Lateral estimates for a sunny visual memory vs. a sunny run.



Heading estimates for a sunny visual memory vs. a sunny run.

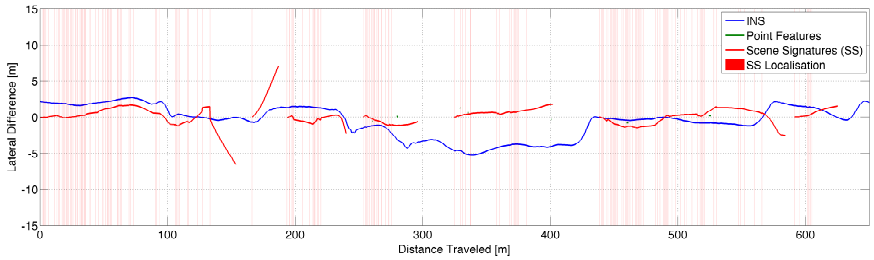


Live VO profile against groundtruth.

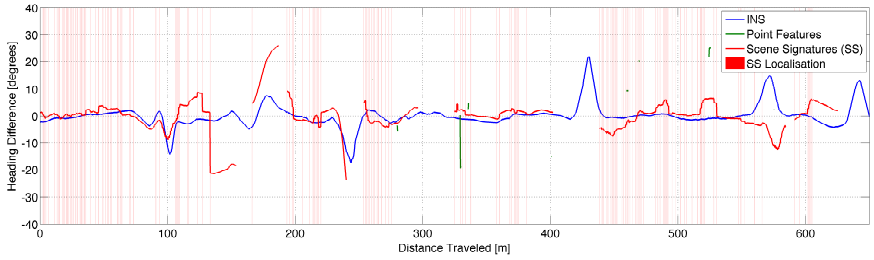


Number of feature matches.

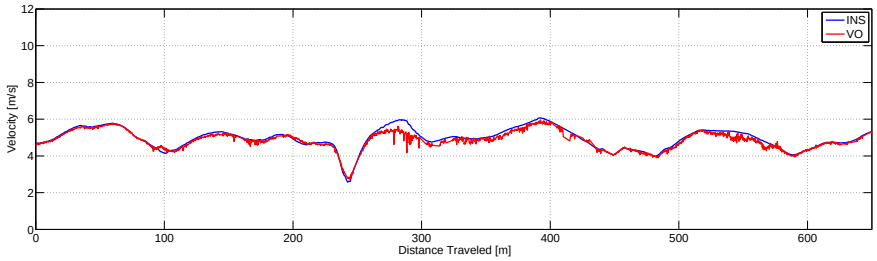
Figure 6.4: Localisation results for a sunny run through segments 1-6 (Oxford) using an illumination-invariant colour space. Note how noisy the estimates are and the low number of matches when compared to the RGB counterpart shown in Chapter 5.



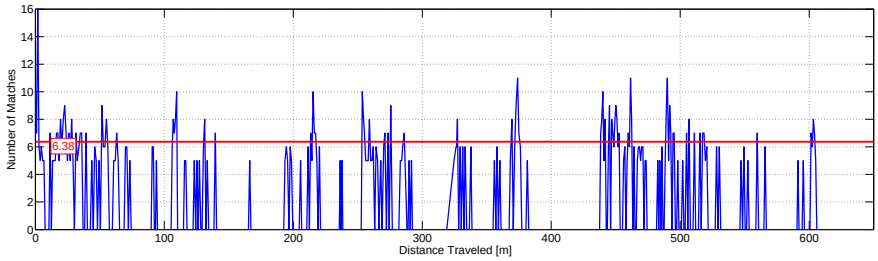
Lateral estimates for a sunny visual memory vs. a misty, snow run.



Heading estimates for a sunny visual memory vs. a misty, snow run



Live VO profile against groundtruth.



Number of feature matches.

Figure 6.5: Localisation results for the misty, snow run (Begbroke) using an illumination-invariant colour space. Note how noisy the estimates are and the low number of matches when compared to the RGB counterpart shown in Chapter 5.

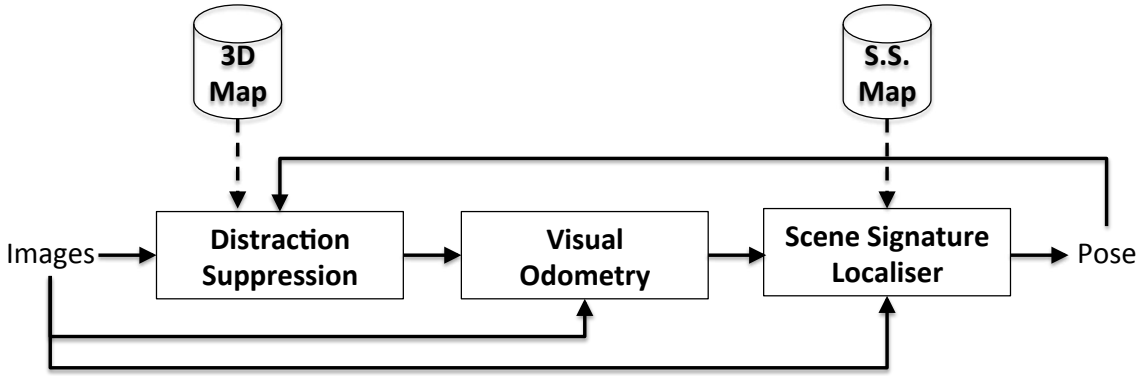


Figure 6.6: Combining distraction suppression with scene signatures. The distraction suppression module loads 3D point clouds from disk and uses an estimated pose to compute the distraction masks, which are then fed into the VO block. The frame-to-frame motion estimate is then fed into the scene-signature localiser as discussed in Chapter 5.

6.2 Distraction Suppression

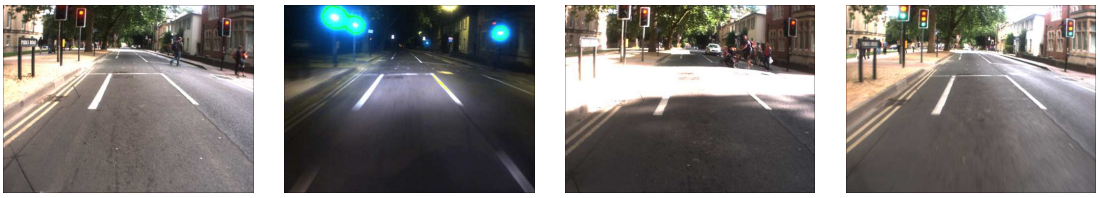
As illustrated from the results in the previous chapter, accurate dead reckoning is important to constrain the translational component of the estimate. Earlier, in Chapter 3 we introduced a technique for generating distraction masks for robust egomotion estimation. In this section we incoporate distraction suppression into our pipeline and analyse the performance of the combined system. Figure 6.6 illustrates how this pipeline works. Distraction suppression sits at the front and uses the previous pose estimate and previous motion estimate to predict the future pose of the vehicle in order to generate a distraction mask. Since distraction suppression receives the pose from the previous timestep, $\mathbf{T}_{k-1,m}$, we use the previous frame-to-frame VO measurement as the prediction for the current time (i.e., assume $\hat{\mathbf{T}}_{k,k-1} \approx \mathbf{T}_{k-1,k-2}$ to compute $\hat{\mathbf{T}}_{k,m} = \hat{\mathbf{T}}_{k,k-1}\mathbf{T}_{k-1,m}$). This constant-velocity motion model has its limitations, which will be discussed later. Using odometry or an Inertial Measurement Unit (IMU) could help by providing an accurate realtime prediction of the pose; however, in this work we restrict ourselves to using a single monocular camera. The output of this process yields a distraction mask, which is then fed

Figure 6.7: Comparison of synthetic images generated using the image-interpolation method described in Chapter 3 versus the new depth-buffered approach. Although the new approach results in a sparser image in the near field, it is significantly faster and allows for 5-10 Hz operation. Additionally, it more accurate in depth since the points are depth ordered. Looking at the interpolated images (far right column), one can see mixed pixel values from the interpolation of the nearest neighbours.

into the VO block in order to mask features extracted on ephemeral objects. The frame-to-frame motion estimate is then output into the localiser.

As the system described in Chapter 3 did not run in realtime and was implemented in Matlab, some modifications have been made to port it to C++. Instead of performing interpolation in image space to generate the synthetic images, we now reproject the sparse points into the image with depth buffering. Not only is this faster, but it correctly captures the depth of each reprojected point, which was not the case in the original implementation. The only downside is that there are gaps in the image owing to the fact that we do not use a mesh. However, as can be seen in Figure 6.7, the resulting synthetic images look very similar and due to the filtering that is applied to the final distraction mask, the gaps become less noticeable.

The other change required for faster operation has to do with the Jacobian term



Map image: clear and sunny; noon. | Test image: dark, night. | Test image: sunny with shadows. | Test image: clear and sunny; noon.

Figure 6.8: Example test images used in our Oxford localisation experiments. These were chosen due to their large visual variability.

introduced in Section 3.3.3:

$$Z_x := \sqrt{\left(\frac{\partial z^s}{\partial \mathbf{x}}\right) \mathbf{P}_x \left(\frac{\partial z^s}{\partial \mathbf{x}}\right)^T}, \tag{6.2}$$

which represented the change in depth at a particular pixel location given the localisation uncertainty, $\mathbf{P}_x$. Originally, we had computed an averaged depth-Jacobian image from a separate training set to approximate this Jacobian. Although this worked well, it is limiting in the sense that it is a static image that does not change based on scene geometry. Experimenting with different approaches, we borrowed the technique used for the optical flow differencing, which was to scale the difference by the associated depth. This provides the behaviour we desired, which is to down-weight errors in the near-field relative to the far-field. At present, the system runs at 5 Hz, with the main bottleneck being the stereo matching. In future work, we wish to run the stereo matching and depth buffering on a GPU for faster operation.

Recall that the localisation updates occur at approximately 5 Hz as well, but since odometry can be used in between localisation updates, the system can still report pose estimates at frame rate. However, to produce repeatable results that are independent of graphics card performance, the results are generated offline assuming the the localisation updates occur at 5 Hz and that the masks are provided at 5 Hz.

As the Begbroke datasets did not have any dynamic objects, we focus on the

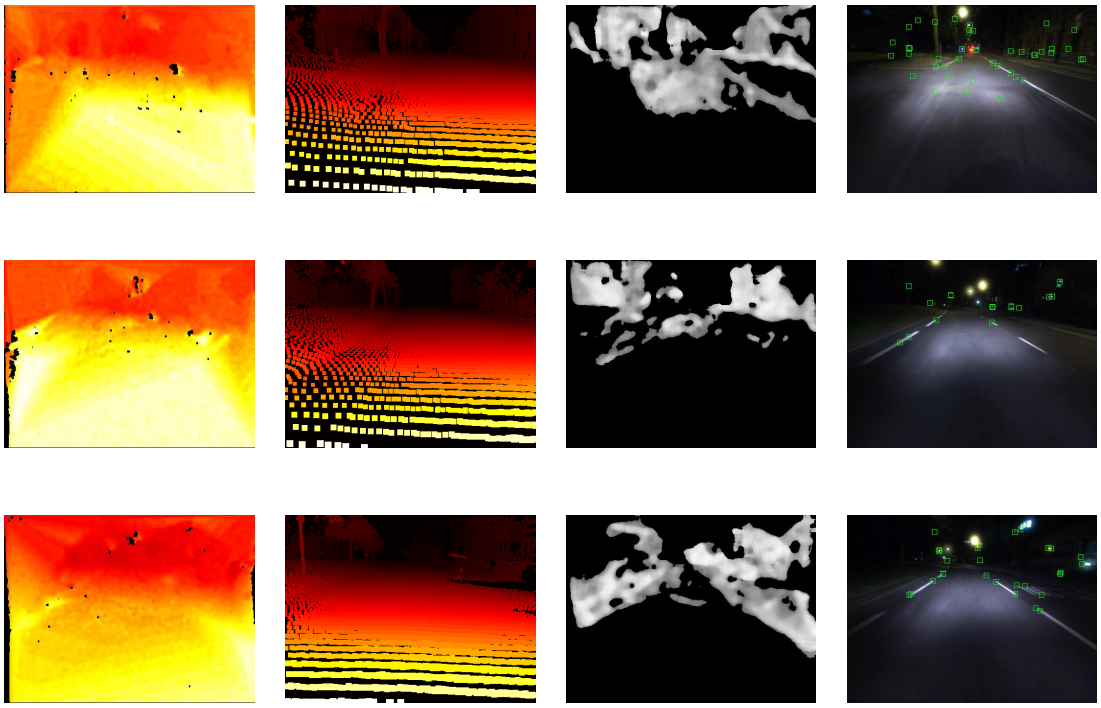
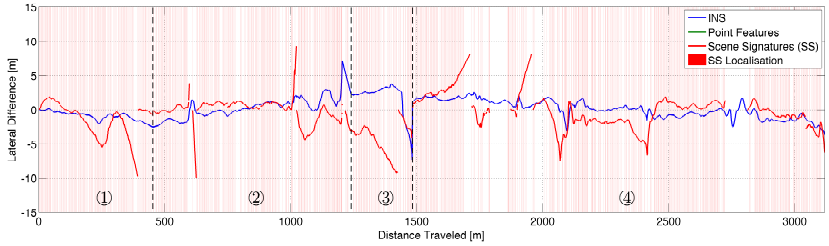


Figure 6.9: Images taken from the nighttime run in Oxford. Each row contains: (i) the measured disparity, (ii) the predicted disparity using the 3D scene prior, (iii) the depth-weighted distraction mask, and (iv) the VO output with active features shown as green squares. As is evident, the measured disparity images are completely erroneous for this dataset owing to the lack of sufficient texture. As a result, the distraction mask restricts the allowable search region for the VO system, which results in poor motion estimates.

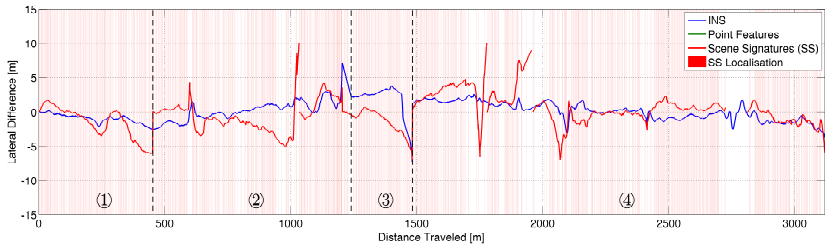
Oxford datasets in this section. For clarity, we include Figure 6.8 again, which shows a representative map image and live image used in these experiments.

6.2.0.3 Nighttime Run

For the nighttime dataset, distraction suppression made the system, on average, worse that the baseline because the stereo matching was unable to cope with completely textureless regions. As a result, the distraction masks encompassed nearly the entire image, leading to erroneous motion estimates. Figure 6.9 show examples of the synthetic disparity image, the live disparity image, and the resulting distraction masks. Figure 6.10 show the lateral errors with and without distraction



With distraction suppression.



Without distraction suppression.

Figure 6.10: Lateral localisation results for the Oxford night run with/without distraction suppression. Including distraction suppression in this run resulted in poorer performance when compared with the baseline system. This is because the stereo matching technique suffered in the low-light conditions, resulting in large discrepancies with the predicted disparity images.

suppression for a comparison. For all plots on all of the estimation errors the reader is referred to Appendix F.

The nighttime run proved to be the most challenging of all the datasets (including Begbroke) because of a lack of illumination and texture throughout most of the run. This was even observed in the last chapter with the scene signature results. Most matches were very noisy as there was not enough detail in the images to match against. The streets were poorly lit and with the longer exposure times, the only visible sections encountered significant motion blur.

6.2.0.4 Shadow Run

For the live run with shadows, our results indicated that distraction suppression was able to improve localisation in a number of areas and suffered in some other areas. Figure 6.11 provides a side-by-side comparison of the lateral error profiles

with some key locations labeled for clarity. There were many instances were the pose estimates were improved when passing oncoming traffic, which, as shown in Chapter 3, can degrade the quality of VO. Since our scene-signature localiser relies on a strong motion prior, this resulted in more accurate lateral estimates according to both groundtruth and the point-feature-based system, which we note works well when matching scenes of similar appearance. When passing oncoming traffic, the VO solution tends to be gently "tugged" towards the direction of traffic if enough outlier feature matches are accepted, which was the case in the labeled regions. Unfortunately, there were not any examples of where most of the image was obscured by moving vehicles, which was the case in the London datasets. Nonetheless, we still observed situations where the techniques improved the pose estimate.

The primary failure mode (shown in case B in the figure) is due to an interesting feedback loop that naturally arises with the inclusion of distraction suppression. Referring back to the pipeline diagram in Figure 6.6, we see that the localisation estimates directly influence the quality of the distraction mask, which in turn, affects the VO output. Since the localisation system uses a strong motion prior, if the odometry output is corrupted, the final pose estimate will suffer. This cyclical pattern means that if the pose estimate begins to drift, the system is likely to produce erroneous distraction masks and fail to localise.

6.2.0.5 Sunny Run

As with the previous run, we found that there were regions where distraction suppression helped and regions where it did not (see Figure 6.12). As before, the typical failure mode involved a feedback loop whereby poor pose estimates resulted in poor VO estimates through the distraction masks, which worsened the pose estimates further. Typical success cases involved situations where the vehicle drove behind busses or passed oncoming traffic, which degraded the baseline VO system without

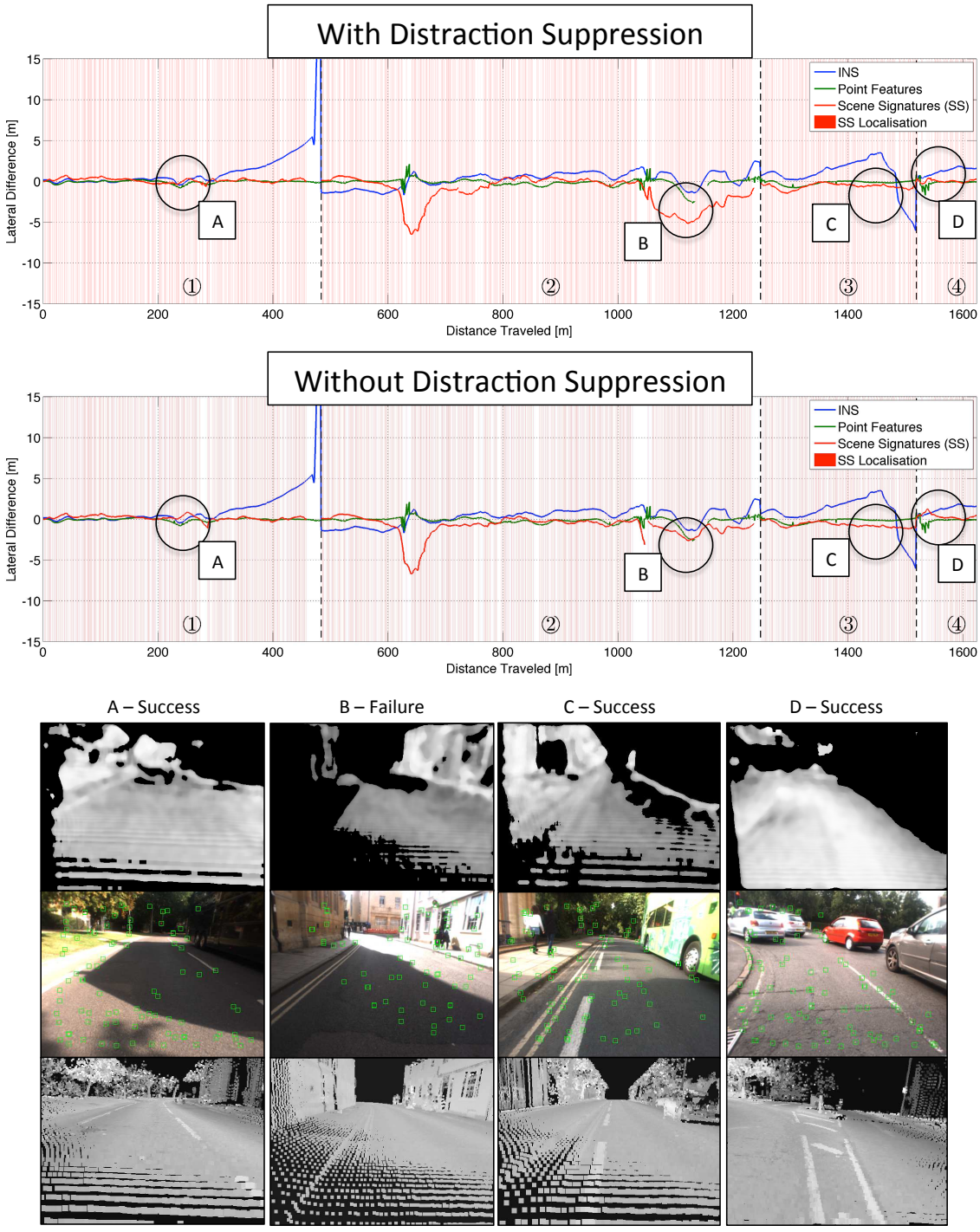


Figure 6.11: Lateral localisation results for the Oxford shadow run with/without distraction suppression. Some examples where distraction suppression improved and worsened the pose estimate have been labeled. For each case, we show the distraction mask, the frame-to-frame matched VO features indicated with green squares, and the synthetic reflectance image, which provides a qualitative measure on the quality of estimate in the map. Note that the localisation estimate for case D is off by several meters, but we are still able to identity large structural inconsistencies.

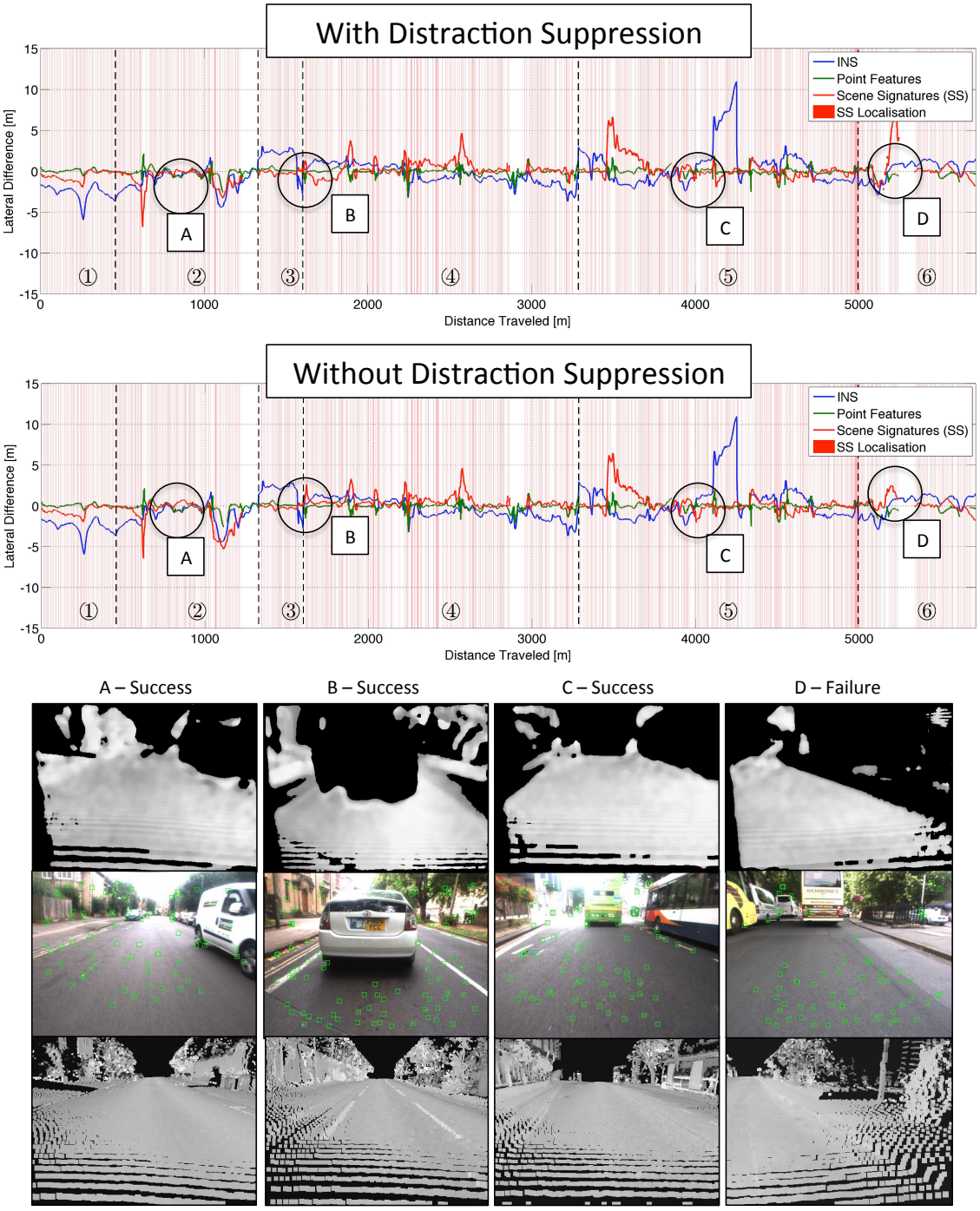
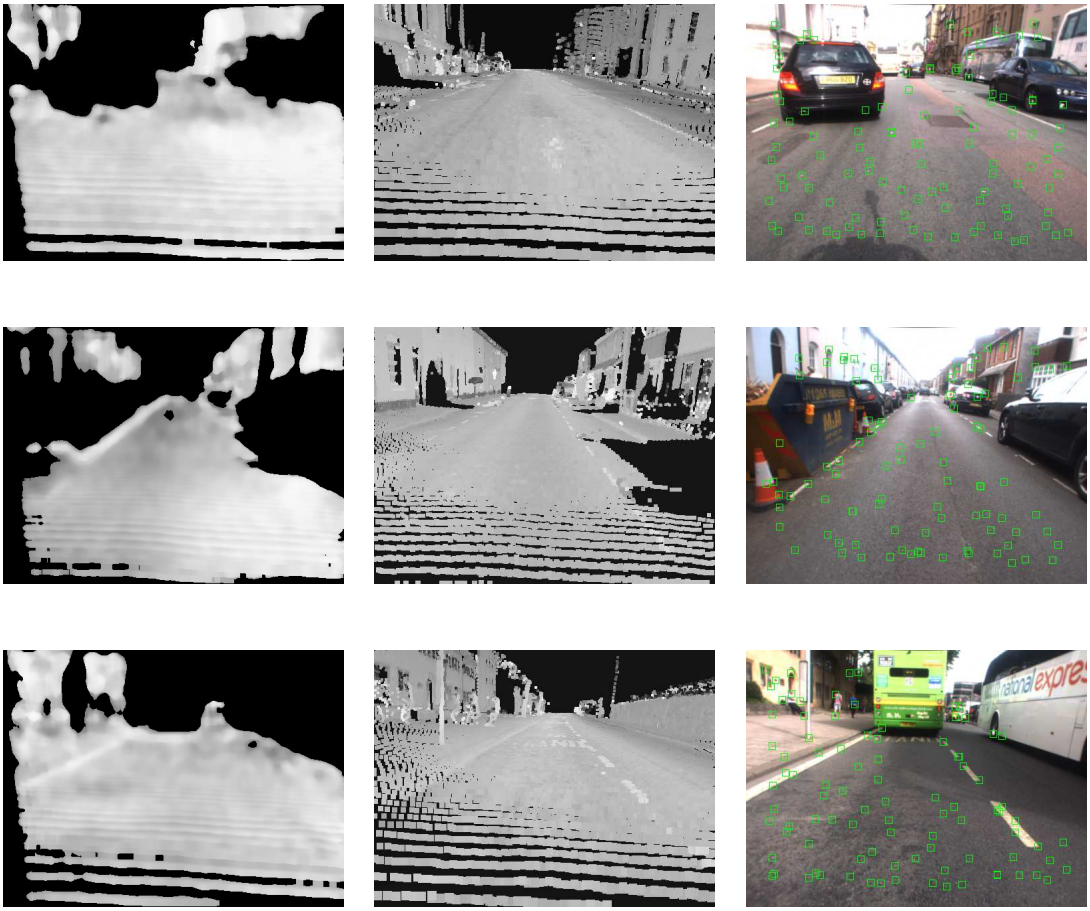


Figure 6.12: Lateral localisation results for the Oxford sunny run with/without distraction suppression. Some examples were distraction suppression improved and worsened the pose estimate have been labeled. For each case, we show the distraction mask, the frame-to-frame matched VO features indicated with green squares, and the synthetic reflectance image, which provides a qualitative measure on the quality of estimate in the map.

distraction suppression. Figures 6.13 and 6.14 show other representative examples of when distraction suppression improved/degraded the system's performance.

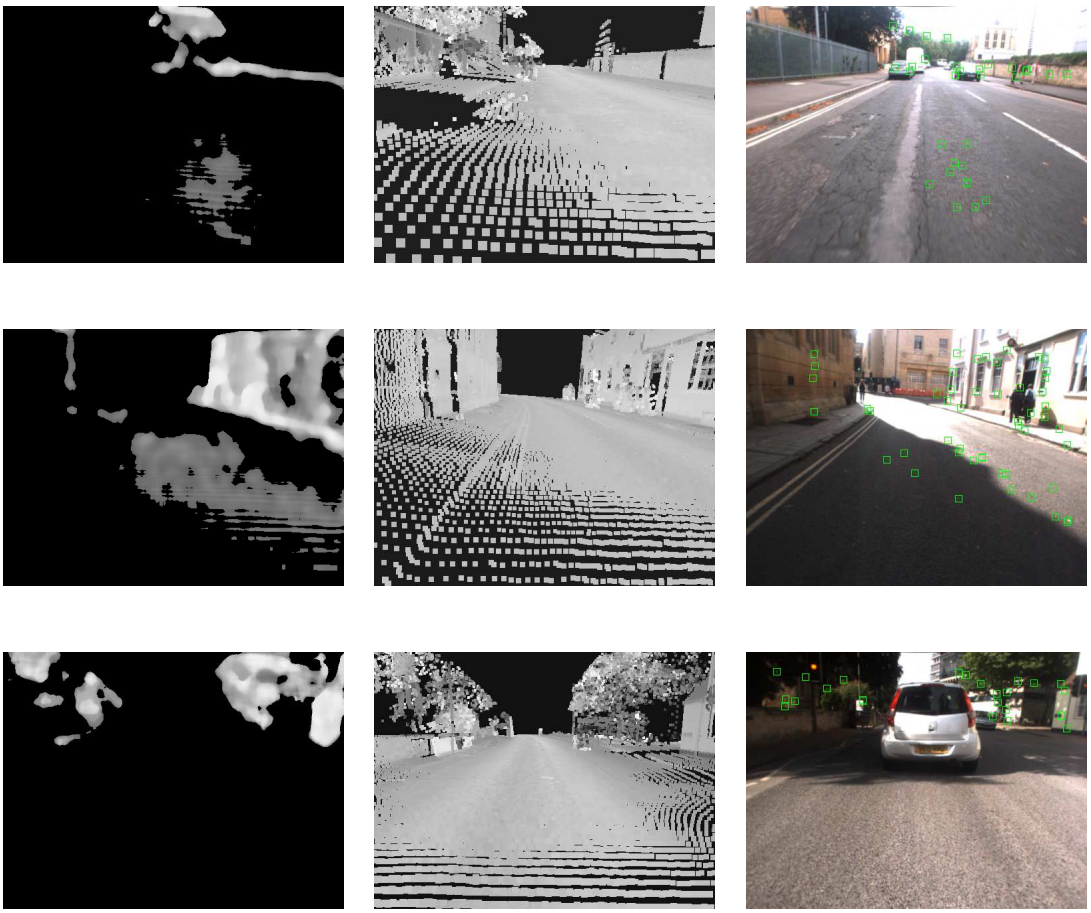
One of the interesting failure modes included cases where the system encountered sudden changes in orientation, which is not accurately accounted for with a constant-velocity motion model. Figure 6.15 shows an example where the vehicle went over a bump and pitched upwards suddenly. Since the estimated pose in the 3D scene prior did not predict this pitch, the predicted disparity disagreed quite significantly in the near field, leading to an erroneous distraction mask.

Thus, we have observed both positive and negative results with the inclusion of distraction suppression in the scene-signature localisation pipeline. This is not entirely surprising since the localisation estimates are accurate to the meter and not centimetre. As a result, the synthetic disparity images sometimes disagree with the measured disparity images. However, there were several cases when passing large vehicles where distraction suppression improved the VO estimates and, in turn, the localisation estimates. In more cluttered environments, such as central London, these types of situations would occur more often, as observed in Chapter 3.



Distraction Mask. Synthetic Image. VO Features.

Figure 6.13: Some examples where distraction suppression helped improve the localisation estimates. Each row shows the following images: (i) the distraction mask, (ii) the synthetic image, and (iii) the features used for VO after applying the mask. Note that the localisation estimate for the top row is off by several meters, but we are still able to identity large structural inconsistencies.



Distraction Mask. Synthetic Image. VO Features.

Figure 6.14: Some examples where distraction suppression failed. Each row shows the following images: (i) the distraction mask, (ii) the synthetic image, and (iii) the features used for VO after applying the mask. In the top two rows, we see examples of where the localisation estimate drifted substantially to the point where the predicted structure of the scene significantly disagreed with the observed structure of the scene. The bottom row represents an interesting case where the vehicle pitched upwards very suddenly. This pitch was not predicted since we use a constant-velocity motion model. As a result, the predicted disparity disagreed with the observed disparity and produced an erroneous mask.

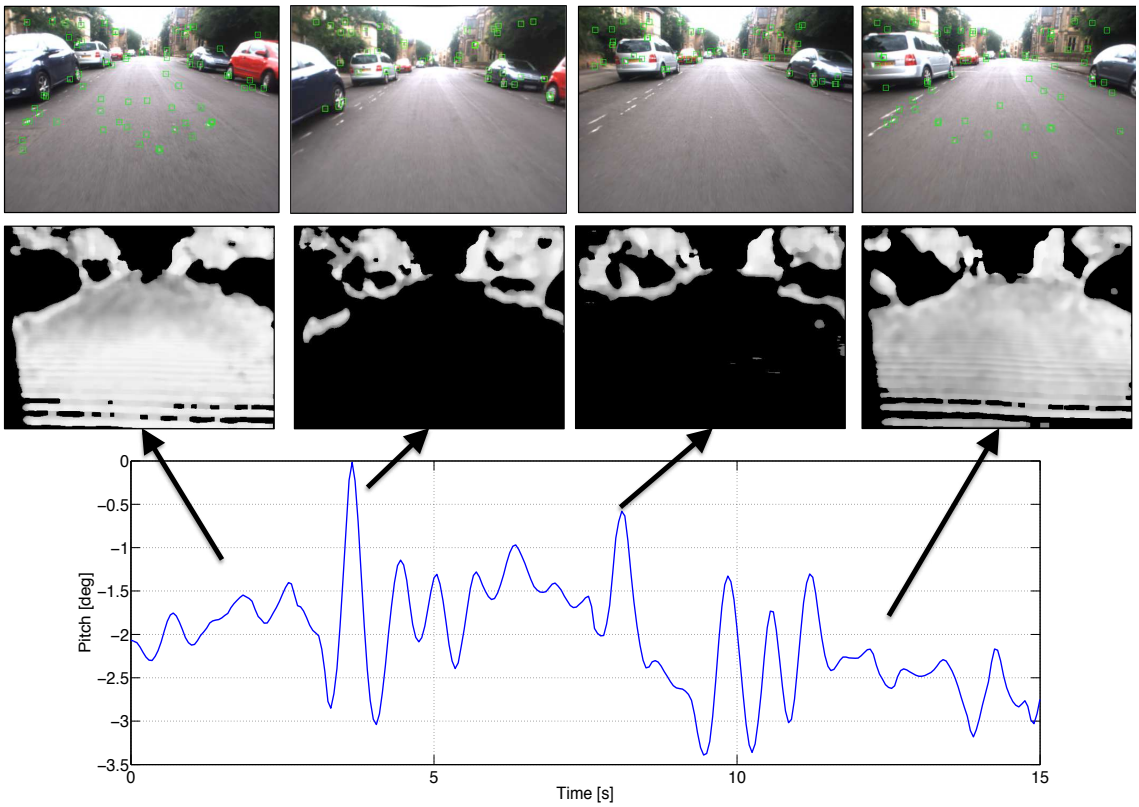


Figure 6.15: An example of when the constant-velocity motion assumption was violated, leading to a poor distraction mask. In this case we see that a sudden pitch that was not captured by the motion model leads to a large disagreement in the near-field predicted disparity (middle two images). Pitch and roll errors were the most significant contributors to large errors in the distraction masks. The far left and far right images show successful distraction masks since the constant-velocity motion model proved to be a good approximation to the true motion.

Chapter 7

Conclusion

This chapter restates the motivation for the thesis and summaries the major ideas, results, and contributions. It concludes with a discussion on the future directions of the work, as well as some closing remarks.

7.0.1 Summary

The goal of this thesis was to develop techniques to help address the problem of long-term, vision-based localisation outdoors, using only a single stereo camera and leveraging laser and appearance priors. More specifically, this thesis examined the problem of navigating amidst significant visual change, both sudden and gradual. Coping with extreme appearance changes caused by dynamic objects, external illuminates, weather, and seasonal changes is a necessary requirement for any autonomous agent operating outdoors over extended periods of time.

In Chapter 3, we introduced a technique called distraction suppression (McManus et al., 2013), which leverages 3D scene priors to mask objects that are not part of the static scene. This is important for relative motion estimation techniques like VO, as standard outlier rejection schemes will not work when the majority of matched features belong to moving objects (e.g., a large bus). This technique helps address

the problem of dealing with moving objects.

In Chapter 4, we presented a simple, yet effective model of black-body illuminates to account for effects of shadows, which cause issues for point-feature-based systems (McManus et al., 2014a). It was shown that the performance of combining a parallel localiser working in an illumination-invariant colour space drastically reduced the distance traveled in between localisation failures. However, this does not address the issue of navigation during night when the black-body radiator assumption is not valid, nor does it address problems associated with different weather and seasons. Ultimately, we found that for the task of outdoor localisation over long periods, a different approach to point-features was needed. Using one fixed detection scheme to find associations of corners or edges did not seem like the way forward.

In Chapter 5 a new approach to metric localisation was presented, which leverages appearance priors to learn place-dependent feature detectors. These feature detectors are designed to identify distinctive visual elements that can be associated across extreme lighting and weather conditions (McManus et al., 2014b, 2015). We used over 100 km of data for training and showed how these place-dependent feature detectors can be integrated in a standard keyframe-based localisation pipeline for robust localisation. Our results showed that the scene-signature approach outperformed the INS in many situations in terms of accuracy and was considerably more reliable than the point-feature system.

Chapter 6 was interested in analysing the respective performance differences resulting from combining illumination invariance and distraction suppression into the scene-signature pipeline. The results suggest that training scene signatures on an illumination invariant colour space and combining a parallel localiser as was done in Chapter 4 provided little to no benefit. This is because, by construction, illumination-invariant features are learned during the training procedure. Incorporating distraction suppression into the pipeline offered improvements in some areas

but led to failures in others due to a feedback loop that would result when poor pose estimates led to erroneous distraction masks, which worsen the pose estimates further. This is a natural consequence of the fact that the localiser is only accurate to within meters and thus, there are some cases where the predicted and measured structure differ significantly. Nonetheless, there were several cases where distraction suppression improved the lateral/heading estimates of the localiser.

7.0.2 Future Work

There are a number of future directions for the work presented in this thesis. In particular, we focus on Chapter 3 and Chapter 5.

- **Aiding Object Detection**: Distraction suppression is not only a useful tool for navigation systems, but could also be useful for perception tasks since it naturally segments the scene into foreground and background elements. Recall from Chapter 3 that after producing the uncertainty-weighted disparity difference, $\tilde{e}_d$, and normalising, we defined a background likelihood score as: $p(\text{background}) = 1 - \tilde{e}_d/\tilde{e}_{d,max}$. If we instead just considered the compliment of this, we obtain the likelihood of a pixel belonging to the foreground, which would be of significant use for an object detection system as it provides a strong cue for where to search.

- **Distraction Mask Quality Metrics**: Incorporating some form of quality checking seems crucial in order to prevent the failure cycle observed in Chapter 6. This is one of the dangers of the feedback design, since a drifting pose estimate can lead to suboptimal distraction masks, which can worsen VO and in turn, affect the localiser further. One approach might be to look at the uncertainty of the localisation estimate to judge whether or not to accept it as a valid guess. However, if the estimator is overconfident, the same problem

may arise. Another approach might be to learn what places in the environment are unreliable for localisation. For example, if there are certain sections where historically the vehicle struggled to localise, it might be possible to learn a place-depedent classifier on lateral and longitudinal estimates.

- **Extrapolating Distraction Masks**: As mentioned in Chapter 3, the bottleneck in terms of speed is the stereo matching, which runs at 5-10 Hz. It would be useful to explore methods of extrapolating in time to produce a predicted mask based on a sliding window of distraction masks. This would allow the system to effectively operate at framerate. An optical flow approach would be useful, but this is an expensive procedure. It is not yet clear how this mask prediction can be accomplished.

- **Reducing Scene-Signature Training Times**: The training times are currently quite long at approximately 2 hours per place. Although this happens offline and could be done on a cluster of high-performance computers, if we were to scale up the distances to thousands of kilometres, the training times need to be reduced. Instead of densely sampling the entire image at each place and retraining when new data are obtained, it may be possible to bootstrap the sampling process by using the preprocessed heat maps shown in Section 5.4.2, which represent the most likely locations in the image to find scene signatures. Each time the system retrains on more data, the per-place feature distributions could be updated, which will hopefully improve the sampling for any future retraining.

- **Scene Signatures for Place Recognition**: Extending the scene-signature approach to place recognition tasks is another possible direction. Recall that the training procedure was focused on finding distinctive elements specific to a particular place. This does not mean that the elements are necessarily distinct

from other places. One could imagine extending this training framework to consider all locations in the map, in order to find features that are unique across all places. Having a layer that sits above our coarse metric localiser that could report which place the vehicle is currently close to would be crucial for when the system becomes lost.

7.0.3 Closing Remarks

The message of this thesis was to show that a one-size-fits-all approach to vision is not optimal for life-long, persistent navigation. Instead, we wish to look at leveraging prior knowledge to develop place-specific policies for a given task. This idea is not exclusive to just localisation, but could extend to many other domains, such as planning, object detection, sensor selection, etc. By taking this stance, we are more likely to develop robust systems that will work in real world applications.

Appendix A

Acronyms

SVM Support Vector Machine

VO Visual Odometry

SLAM Simultaneous Localization and Mapping

RANSAC Random Sample and Consensus

GPS Global Positioning System

DARPA Defense Advanced Research Projects Agency

SURF Speeded-Up Robust Features

HOG Histogram of Oriented Gradients

SIFT Scale Invariant Feature Transform

pdfs probability density functions

pose position and orientation

LM Levenberg Marquardt

PF Particle Filter

UKF Unscented Kalman Filter

KF Kalman Filter

EKF Extended Kalman Filter

MoG Mixture of Gaussians

INS Inertial Navigation System

FAST Features from Accelerated Segment Test

BRIEF Binary Robust Independent Elementary Features

ORB Oriented FAST and Rotated BRIEF

SfM Structure from Motion

RFS Random Finite Sets

DOF Degree of Freedom

IMU Inertial Measurement Unit

Appendix B

Camera Geometry

B.0.4 Stereo Model

Given a point expressed in the camera frame, $\mathbf{p}_c$, the stereo model, $\mathbf{y}_c := \mathbf{g}(\mathbf{p}_c)$, reprojects the point into left and right image planes, according to

$$\mathbf{y}_c := \mathbf{h}(\mathbf{T}_{c,w}\boldsymbol{p}_w) = \begin{bmatrix} u_l \\ v_l \\ d \end{bmatrix} = \frac{1}{z} \begin{bmatrix} x\ f_u + z\ c_u \\ y\ f_v + z\ c_v \\ f_u\ b \end{bmatrix} \tag{B.1}$$

where $\{f_u, f_v\}$ are the horizontal and vertical focal lengths, b is the baseline and $\{c_u, c_v\}$ are the horizontal and vertical positions of the optical centre.

B.0.5 Stereo Jacobians

The Jacobian with respect to the point, $\mathbf{p}_c$, is

$$\frac{\partial \mathbf{h}}{\partial \mathbf{p}_c} = \begin{bmatrix} f_u/z & 0 & -f_u x/z^2 \\ 0 & f_v/z & -f_v y/z^2 \\ 0 & 0 & -f_u\ b/z^2 \end{bmatrix} \tag{B.2}$$

The Jacobian with respect to a perturbation in the state, $\mathbf{x}_c$, is

$$\boxed{\frac{\partial \mathbf{h}}{\partial \mathbf{x}_c} = \frac{\partial \mathbf{h}}{\partial \mathbf{p}_c}\frac{\partial \mathbf{p}_c}{\partial \mathbf{x}_c} = \begin{bmatrix} f_u/z & 0 & -f_u x/z^2 \\ 0 & f_v/z & -f_v y/z^2 \\ 0 & 0 & -f_u\ b/z^2 \end{bmatrix} \begin{bmatrix} \mathbf{1} & -\mathbf{p}_c^\wedge \end{bmatrix}} \tag{B.3}$$

B.0.6 Monocular Model

The monocular camera model is given by the following:

$$\boxed{\mathbf{y}_c := \mathbf{h}(\mathbf{T}_{c,w}\boldsymbol{p}_c) = \begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{z}\begin{bmatrix} x\ f_u + z\ c_u \\ y\ f_v + z\ c_v \end{bmatrix}} \tag{B.4}$$

B.0.7 Monocular Jacobians

Following a similar derivation as above, the Jacobians are

$$\boxed{\frac{\partial \mathbf{h}}{\partial \mathbf{p}_c} = \begin{bmatrix} f_u/z & 0 & -f_u x/z^2 \\ 0 & f_v/z & -f_v y/z^2 \end{bmatrix}} \tag{B.5}$$

$$\boxed{\frac{\partial \mathbf{h}}{\partial \mathbf{x}_c} = \begin{bmatrix} f_u/z & 0 & -f_u x/z^2 \\ 0 & f_v/z & -f_v y/z^2 \end{bmatrix} \begin{bmatrix} \mathbf{1} & -\mathbf{p}_c^\wedge \end{bmatrix}} \tag{B.6}$$

Appendix C

Posegraph Relaxation

Posegraph relaxation is a method for optimising over a chain of poses subject to a collection of constraints. These constraints can include relative constraints between poses in the form of odometry measurements, loop-closure constraints, or localisation constraints (i.e., a constraint on the position and orientation of a pose relative to some map). A toy example is presented in Figure C.1.

C.0.8 Problem Definition

There are three types of constraints being considered,

1. Relative constraints (e.g., from visual odometry),
2. Loop-closure constraints (e.g., FABMAP),
3. Localisation constraints (e.g., from experience-based navigation).

Using Figure C.1 as a guide, this section will derive the error terms and Jacobians for each type of constraint. We formulate this problem slightly differently and instead of using transformation matrices, we perturb the translation and rotation equations directly; however, one can easily workout the analogous set of Jacobians for using transformation matrices.

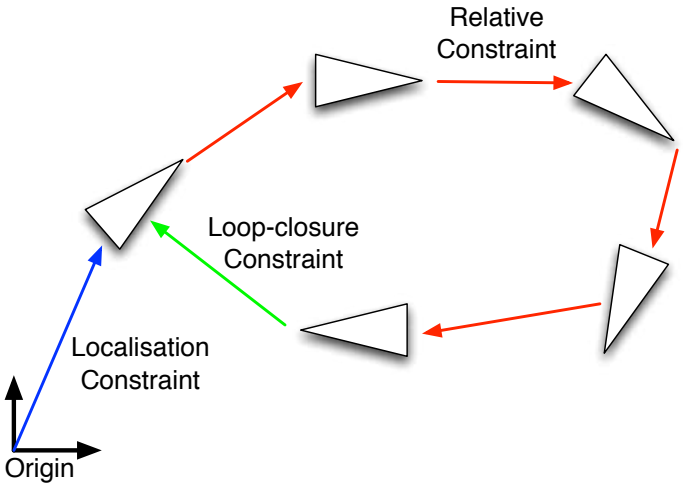


Figure C.1: Example setup of a pose graph relaxation problem, where red links represent relative constraints (e.g., from odometry), green links are loop-closure constraints (e.g., from a place-recognition system), and blue links are localisation constraints. Note that the optimisation is done in a global frame and at least one localisation constraint is required in order to anchor the graph to a common coordinate frame.

C.0.9 Relative Constraints

Defining the pose at time k as $\mathbf{x}_k := \{\mathbf{t}_0^{k,0}, \mathbf{R}_{0,k}\}$ and time $k-1$ as $\mathbf{x}_{k-1} := \{\mathbf{t}_0^{k-1,0}, \mathbf{R}_{0,k-1}\}$, a relative constraint between the two poses is defined by a translation and transformation, $\mathbf{u}_k := \{\mathbf{d}_{k-1}^{k,k-1}, \mathbf{\Psi}_{k-1,k}\}$, with an associated inverse covariance $\mathbf{Q}_{k-1,k}^{-1}$ (from odometry for example). The motion model takes the following form,

$$\mathbf{t}_0^{k,0} = \mathbf{t}_0^{k-1,0} + \mathbf{R}_{0,k-1}\mathbf{d}_{k-1}^{k,k-1} \tag{C.1}$$

$$\mathbf{R}_{0,k} = \mathbf{R}_{0,k-1}\mathbf{\Psi}_{k-1,k} \tag{C.2}$$

The error term for this relative constraint is defined as,

$$\boxed{\mathbf{e}_k := \begin{bmatrix} \mathbf{e}_{k,\text{trans}} \\ \mathbf{e}_{k,\text{rot}} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_0^{k,0} - \left(\mathbf{t}_0^{k-1,0} + \mathbf{R}_{0,k-1}\mathbf{d}_{k-1}^{k-1,k}\right) \\ \left(\mathbf{R}_{0,k}\mathbf{\Psi}_{k-1,k}^T\mathbf{R}_{0,k-1}^T - \mathbf{1}\right)^\vee \end{bmatrix}} \tag{C.3}$$

The object function is a Mahalanobis distance,

$$J_{\text{rel}} := \frac{1}{2}\sum_{k}^{K} \mathbf{e}_k^T \boldsymbol{\Sigma}_k^{-1} \mathbf{e}_k \tag{C.4}$$

where $\boldsymbol{\Sigma}_k^{-1} := \mathbf{E}_{u,k}^T \mathbf{Q}_{k-1,k}^{-1} \mathbf{E}_{u,k}$ and $\mathbf{E}_{u,k}$ is the Jacobian of the error function with respect to the odometry noise. As this is a nonlinear equation, we perform a frist-order Taylor series expansion on the error terms,

$$J_{\text{rel}} \approx \frac{1}{2}\sum_{k}^{K} \left(\bar{\mathbf{e}}_k + \mathbf{E}_{x,k}\delta\mathbf{x}_k + \mathbf{E}_{x,k-1}\delta\mathbf{x}_{k-1}\right)^T \boldsymbol{\Sigma}_{k-1,k}^{-1} \left(\bar{\mathbf{e}}_k + \mathbf{E}_{x,k}\delta\mathbf{x}_k + \mathbf{E}_{x,k-1}\delta\mathbf{x}_{k-1}\right) \tag{C.5}$$

where $\mathbf{E}_{x,k}$ and $\mathbf{E}_{x,k-1}$ are the Jacobians of the error function with respect to the state at time k and $k-1$. This is the linearised system of equations, for which we can take the derivative with respect to the perturbed variables and set the derivative to zero. This leads to the following system (considering just one error term for now):

$$\begin{bmatrix} \mathbf{E}_{x,k-1}^T \\ \mathbf{E}_{x,k}^T \end{bmatrix} \boldsymbol{\Sigma}_k^{-1} \begin{bmatrix} \mathbf{E}_{x,k-1} & \mathbf{E}_{x,k} \end{bmatrix} \begin{bmatrix} \delta\mathbf{x}_{k-1} \\ \delta\mathbf{x}_k \end{bmatrix} = - \begin{bmatrix} \mathbf{E}_{x,k-1}^T \\ \mathbf{E}_{x,k}^T \end{bmatrix} \boldsymbol{\Sigma}_k^{-1} \bar{\mathbf{e}}_k \tag{C.6}$$

We will now turn our attention to the Jacobian terms, which can be derived using the perturbation technique.

$$\begin{aligned} \bar{\mathbf{e}}_{k,\text{trans}} + \delta\mathbf{e}_{k,\text{trans}} \approx \bar{\mathbf{t}}_0^{k,0} + \delta\mathbf{t}_0^k - \\ \left(\bar{\mathbf{t}}_0^{k-1,0} + \delta\mathbf{t}_0^{k-1} + \bar{\mathbf{R}}_{0,k-1}\left(\mathbf{1} + \delta\boldsymbol{\phi}_{k-1}^{\times}\right)\left(\mathbf{d}_{k-1} + \delta\mathbf{d}_{k-1}\right)\right) \end{aligned} \tag{C.7}$$

$$\bar{\mathbf{e}}_{k,\text{rot}}^{\wedge} + \delta\mathbf{e}_{k,\text{rot}}^{\wedge} \approx \bar{\mathbf{R}}_{0,k}\left(\mathbf{1} + \delta\boldsymbol{\phi}_k^{\wedge}\right)\left(\mathbf{1} - \delta\boldsymbol{\psi}_{k-1}^{\wedge}\right)\bar{\boldsymbol{\Psi}}_{k-1,k}^T\left(\mathbf{1} - \delta\boldsymbol{\phi}_{k-1}^{\wedge}\right)\bar{\mathbf{R}}_{0,k-1}^T - \mathbf{1} \tag{C.8}$$

Note that for the rotational terms, we have used our approximation given in Section

(2.1.3), which relates a small rotation matrix to a rotation vector. Subtracting off the nominal solution from each side and dropping products of small terms gives us,

$$\delta\mathbf{e}_{k,\text{trans}} = \delta\mathbf{t}_0^k - \delta\mathbf{t}_0^{k-1} - \bar{\mathbf{R}}_{0,k-1}\delta\boldsymbol{\phi}_{k-1}^{\wedge}\mathbf{d}_{k-1} - \bar{\mathbf{R}}_{0,k-1}\delta\mathbf{d}_{k-1} \tag{C.9}$$

$$\delta\mathbf{e}_{k,\text{rot}}^{\wedge} = \bar{\mathbf{R}}_{0,k}\delta\boldsymbol{\phi}_k^{\wedge}\bar{\boldsymbol{\Psi}}_{k-1,k}^T\bar{\mathbf{R}}_{0,k-1}^T - \bar{\mathbf{R}}_{0,k}\bar{\boldsymbol{\Psi}}_{k-1,k}^T\delta\boldsymbol{\phi}_{k-1}^{\wedge}\bar{\mathbf{R}}_{0,k-1}^T - \bar{\mathbf{R}}_{0,k}\delta\boldsymbol{\psi}_{k-1}^{\wedge}\bar{\boldsymbol{\Psi}}_{k-1,k}^T\bar{\mathbf{R}}_{0,k-1}^T \tag{C.10}$$

Using the fact that $\mathbf{a}^{\wedge}\mathbf{b} = -\mathbf{b}^{\wedge}\mathbf{a}$, for any vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{3\times 1}$, we can rewrite the translational error term as

$$\delta\mathbf{e}_{k,\text{trans}} = \delta\mathbf{t}_0^k - \delta\mathbf{t}_0^{k-1} + \bar{\mathbf{R}}_{0,k-1}\mathbf{d}_{k-1}^{\times}\delta\boldsymbol{\phi}_{k-1} - \bar{\mathbf{R}}_{0,k-1}\delta\mathbf{d}_{k-1} \tag{C.11}$$

Turning to the rotational error term, we use the identity, $\mathbf{R}\mathbf{a}^{\wedge}\mathbf{R}^T = (\mathbf{R}\mathbf{a})^{\wedge}$ for any rotation matrix, $\mathbf{R}$, and any vector $\mathbf{a} \in \mathbb{R}^{3\times 1}$ and rewrite the error term as

$$\delta\mathbf{e}_{k,\text{rot}}^{\wedge} = \left(\bar{\mathbf{R}}_{0,k}\delta\boldsymbol{\phi}_k\right)^{\wedge} - \left(\bar{\mathbf{R}}_{0,k-1}\delta\boldsymbol{\phi}_{k-1}\right)^{\wedge} - \left(\bar{\mathbf{R}}_{0,k-1}\delta\boldsymbol{\psi}_{k-1}\right)^{\wedge} \tag{C.12}$$

using the inverse cross operator, $(\cdot)^{\vee}$, yields

$$\delta\mathbf{e}_{k,\text{rot}} = \bar{\mathbf{R}}_{0,k}\delta\boldsymbol{\phi}_k - \bar{\mathbf{R}}_{0,k-1}\delta\boldsymbol{\phi}_{k-1} - \bar{\mathbf{R}}_{0,k}\delta\boldsymbol{\psi}_{k-1} \tag{C.13}$$

Thus, our final set of linearised equations is given by

$$\delta\mathbf{e}_k = \underbrace{\begin{bmatrix}\mathbf{1} & 0\\ 0 & \bar{\mathbf{R}}_{0,k}\end{bmatrix}}_{\mathbf{E}_{x,k}}\underbrace{\begin{bmatrix}\delta\mathbf{t}_k\\ \delta\boldsymbol{\phi}_k\end{bmatrix}}_{\delta\mathbf{x}_k} + \underbrace{\begin{bmatrix}-\mathbf{1} & \bar{\mathbf{R}}_{0,k-1}\mathbf{d}_{k-1}^{\wedge}\\ \mathbf{0} & -\bar{\mathbf{R}}_{0,k-1}\end{bmatrix}}_{\mathbf{E}_{x,k-1}}\underbrace{\begin{bmatrix}\delta\mathbf{t}_{k-1}\\ \delta\boldsymbol{\phi}_{k-1}\end{bmatrix}}_{\delta\mathbf{x}_{k-1}} + \underbrace{\begin{bmatrix}-\bar{\mathbf{R}}_{0,k-1} & \mathbf{0}\\ \mathbf{0} & -\bar{\mathbf{R}}_{0,k}\end{bmatrix}}_{\mathbf{E}_{u,k}}\underbrace{\begin{bmatrix}\delta\mathbf{d}_{k-1}\\ \delta\boldsymbol{\psi}_{k-1}\end{bmatrix}}_{\delta\mathbf{u}_k} \tag{C.14}$$

To summarise, we have taken our state representations of translation/rotation and derived a linearised set of equations in terms of the perturbations of these state

variables. Stacking everything together, we arrive at the final system of equations:

$$\mathbf{E}^T\boldsymbol{\Sigma}^{-1}\mathbf{E}\delta\mathbf{x} = -\mathbf{E}^T\boldsymbol{\Sigma}^{-1}\bar{\mathbf{e}} \tag{C.15}$$

where

$$\delta\mathbf{x} := \begin{bmatrix} \delta\mathbf{t}_0 \\ \delta\boldsymbol{\phi}_0 \\ \vdots \\ \delta\mathbf{t}_K \\ \delta\boldsymbol{\phi}_K \end{bmatrix}, \quad \bar{\mathbf{e}} := \begin{bmatrix} \bar{\mathbf{e}}_1 \\ \vdots \\ \bar{\mathbf{e}}_K \end{bmatrix}, \tag{C.16}$$

$$\boldsymbol{\Sigma}^{-1} := \operatorname{diag}\left(\mathbf{E}_{u,1}^T\mathbf{Q}_{0,1}^{-1}\mathbf{E}_{u,1}, \ldots, \mathbf{E}_{u,K}^T\mathbf{Q}_{K-1,K}^{-1}\mathbf{E}_{u,K}\right), \tag{C.17}$$

$$\mathbf{E} := \begin{bmatrix} -\mathbf{E}_{x,1} & \mathbf{1} & \mathbf{0} & \mathbf{0} & \ldots \\ \mathbf{0} & -\mathbf{E}_{x,2} & \mathbf{1} & \mathbf{0} & \ldots \\ & & & \ddots & \\ \mathbf{0} & \ldots & \ldots & -\mathbf{E}_{x,K} & \mathbf{1} \end{bmatrix} \tag{C.18}$$

C.0.10 Loop-closure Constraints

A loop-closure constraint between pose i and j is specified by a translation and rotation, $\{\mathbf{d}_j^{i,j}, \boldsymbol{\Psi}_{j,i}\}$, an associated inverse covariance, $\mathbf{W}_{j,i}^{-1}$, which represents the level of certainty in the loop-closure measurement. Referring to the previous subsection, the reader should recognise that this is in fact just a relative constraint between two poses. Thus, the Jacobians take the exact same form as illustrated in the previous section, with the exception of their placement in the stacked Jacobian.

C.0.11 Localisation Constraints

A localisation constraint is again given as a translation and rotation, $\mathbf{m}_k := \{\mathbf{d}_0^{k,0}, \mathbf{\Psi}_{0,k}\}$, with measurement uncertainty given by $\mathbf{M}_k^{-1}$. Note that these are global vectors expressed in the same base frame as the state. The error term for a localisation constraint takes the form,

$$\boxed{\mathbf{e}_k := \begin{bmatrix} \mathbf{d}_0^{k,0} - \mathbf{t}_0^{k,0} \\ \left(\mathbf{\Psi}_{0,k}\mathbf{R}_{0,k}^T - \mathbf{1}\right)^\vee \end{bmatrix}} \tag{C.19}$$

We use the perturbation method as before to derive the Jacobians.

$$\bar{\mathbf{e}}_{k,\text{trans}} + \delta\mathbf{e}_{k,\text{trans}} \approx \bar{\mathbf{d}}_0^{k,0} + \delta\mathbf{d}_k - \bar{\mathbf{t}}_0^{k,0} - \delta\mathbf{t}_k \tag{C.20}$$

$$\bar{\mathbf{e}}_{k,\text{rot}}^\wedge + \delta\mathbf{e}_{k,\text{rot}}^\wedge \approx \bar{\mathbf{\Psi}}_{0,k}\left(\mathbf{1} + \delta\boldsymbol{\psi}_k^\wedge\right)\left(\mathbf{1} - \delta\boldsymbol{\phi}_k^\wedge\right)\bar{\mathbf{R}}_{0,k}^T - \mathbf{1} \tag{C.21}$$

The linearised system of equations is

$$\boxed{\delta\mathbf{e}_k = \underbrace{\begin{bmatrix} -\mathbf{1} & \mathbf{0} \\ \mathbf{0} & -\bar{\mathbf{R}}_{0,k} \end{bmatrix}}_{\mathbf{E}_{x,k}} \underbrace{\begin{bmatrix} \delta\mathbf{t}_k \\ \delta\boldsymbol{\phi}_k \end{bmatrix}}_{\delta\mathbf{x}_k} + \underbrace{\begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{\Psi}}_{0,k} \end{bmatrix}}_{\mathbf{E}_{m,k}} \underbrace{\begin{bmatrix} \delta\mathbf{d}_m \\ \delta\boldsymbol{\psi}_m \end{bmatrix}}_{\delta\mathbf{m}_k}} \tag{C.22}$$

C.0.12 Combined System

Stacking all of the error terms together, we have the following least-squares system

$$\mathbf{E}^T\mathbf{\Sigma}^{-1}\mathbf{E}\delta\mathbf{x} = -\mathbf{E}^T\mathbf{\Sigma}^{-1}\bar{\mathbf{e}} \tag{C.23}$$

where

$$\delta\mathbf{x} := \begin{bmatrix} \delta\mathbf{t}_0 \\ \delta\boldsymbol{\phi}_0 \\ \vdots \\ \delta\mathbf{t}_K \\ \delta\boldsymbol{\phi}_K \end{bmatrix}, \qquad \bar{\mathbf{e}} := \begin{bmatrix} \bar{\mathbf{e}}_{\text{rel}} \\ \bar{\mathbf{e}}_{\text{loop}} \\ \bar{\mathbf{e}}_{\text{loc}} \end{bmatrix}, \tag{C.24}$$

$$\boldsymbol{\Sigma}^{-1} := \begin{bmatrix} \boldsymbol{\Sigma}_{\text{rel}}^{-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\text{loop}}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Sigma}_{\text{loc}}^{-1} \end{bmatrix}, \qquad \mathbf{E} := \begin{bmatrix} \mathbf{E}_{\text{rel}} \\ \mathbf{E}_{\text{loop}} \\ \mathbf{E}_{\text{loc}} \end{bmatrix} \tag{C.25}$$

C.0.13 Optimisation

Once again we use Levenberg Marquardt (LM) for the nonlinear optimisation. To reiterate, the LM method works by adding a diagonal matrix $\lambda\mathbf{1}$, where $\lambda > 0$, to the coefficient matrix on the left-hand side in Equation (C.23). Depending on the change in the objective function, this damping parameter is adjusted after each iteration and adapts the convergence properties of LM to be similar to either gradient descent or the Gauss-Newton method. By adding this non-zero diagonal matrix, LM can guard against poorly conditioned Hessian approximations and prevent blow-ups during the optimisation. The algorithm is shown below.

1. Begin with an initial estimate for the state, $\bar{\mathbf{x}} = \left\{\bar{\mathbf{t}}_0, \bar{\mathbf{R}}_0, \ldots, \bar{\mathbf{t}}_K, \bar{\mathbf{R}}_K\right\}$.

2. Begin with an initial damping parameter, $\lambda = \lambda_0$.

3. Solve for the optimal step, $\delta\mathbf{x}^*$, given by

$$\left(\mathbf{E}^T\boldsymbol{\Sigma}^{-1}\mathbf{E} + \lambda\mathbf{1}\right)\delta\mathbf{x}^* = -\mathbf{E}^T\boldsymbol{\Sigma}^{-1}\mathbf{e}(\bar{\mathbf{x}}) \tag{C.26}$$

4. If $|\delta\mathbf{x}^*| <$ threshold, then stop. Else, continue with the next steps.

5. Update our estimate by applying the optimal step, $\delta\mathbf{x}^* = (\delta\mathbf{t}_0^*, \delta\boldsymbol{\phi}_0^*, \ldots, \delta\mathbf{t}_K^*, \delta\boldsymbol{\phi}_K^*)$, to our previous estimate according to

$$\bar{\mathbf{t}}_0^{k,0} \leftarrow \bar{\mathbf{t}}_0^{k,0} + \bar{\mathbf{R}}_{0,k}\delta\mathbf{t}_k^* \tag{C.27}$$

$$\bar{\mathbf{R}}_{0,k} \leftarrow \bar{\mathbf{R}}_{0,k}\boldsymbol{\Phi}_k^* \tag{C.28}$$

for all $k = 0 \ldots K$, and where

$$\boldsymbol{\Phi}_k^* = \cos(\delta\phi_k^*)\mathbf{1} + (1 - \cos(\delta\phi_k^*))\left(\frac{\delta\boldsymbol{\phi}_k^*}{\delta\phi_k^*}\right)\left(\frac{\delta\boldsymbol{\phi}_k^*}{\delta\phi_k^*}\right)^T + \sin(\delta\phi_k^*)\left(\frac{\delta\boldsymbol{\phi}_k^*}{\delta\phi_k^*}\right)^\wedge \tag{C.29}$$

and $\delta\phi_k^* := |\delta\boldsymbol{\phi}_k^*|$.

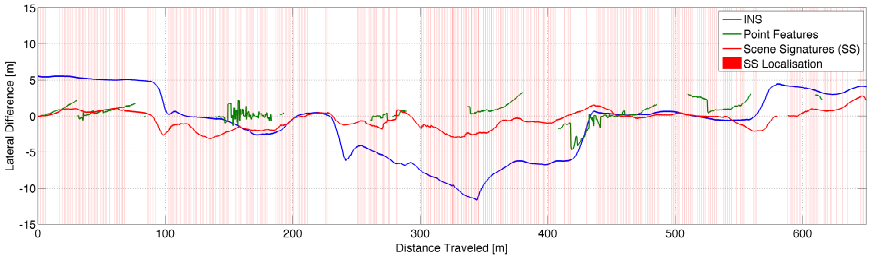
6. If $J_{\text{new}} - J_{\text{previous}} > 0$, $\lambda = \beta\lambda$, where $\beta > 1$. Else, $\lambda = \eta\lambda$, where $\eta < 1$.

7. Return to Step 3.

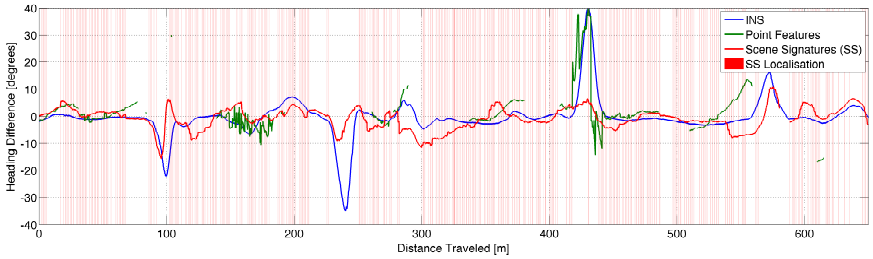
Appendix D

Begbroke Illumination-Invariant Results

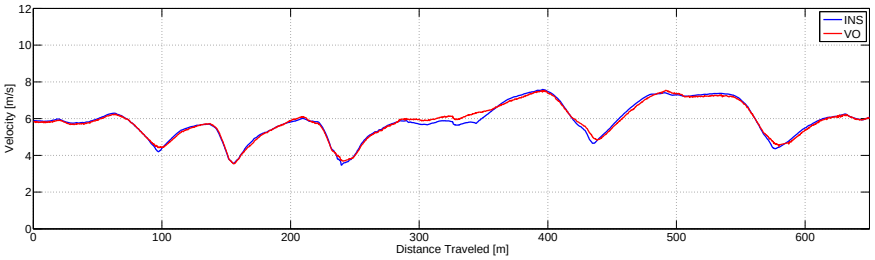
The following figures show the localisation results for Begbroke using the illumination-invariant colour space presented in Chapter 4.



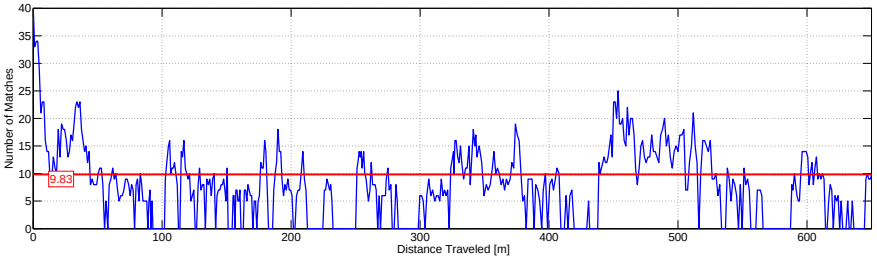
Lateral estimates for a sunny visual memory vs. a sunny run.



Heading estimates for a sunny visual memory vs. a sunny run

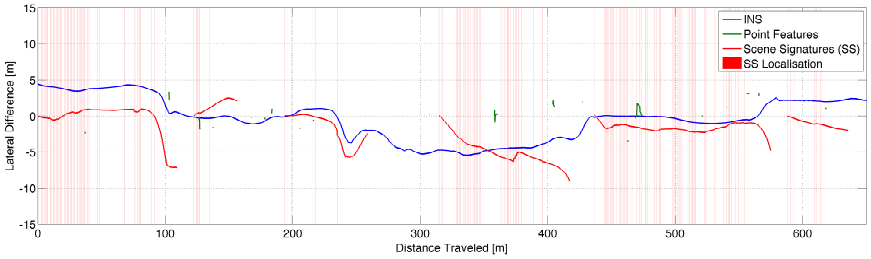


Live VO profile against groundtruth.

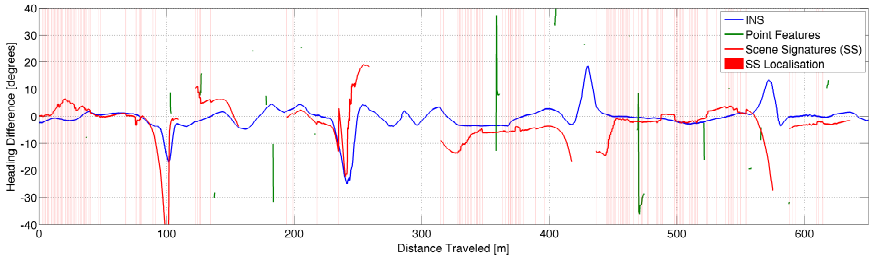


Number of feature matches.

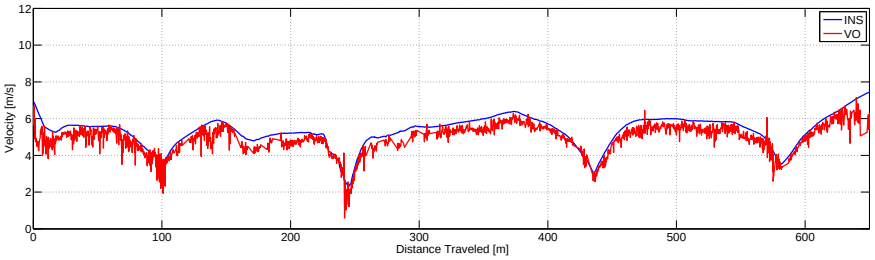
Figure D.1: Localisation results for the sunny afternoon run.



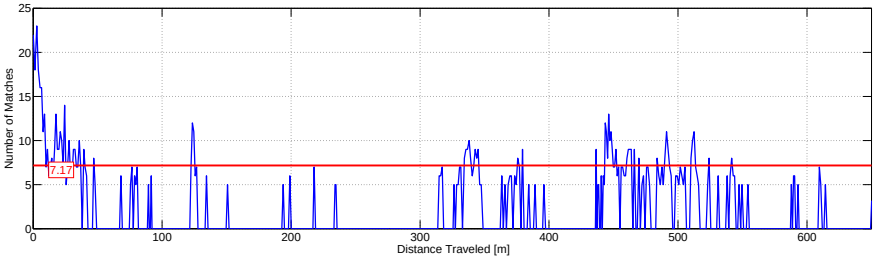
Lateral estimates for a sunny visual memory vs. a clear, evening run.



Heading estimates for a sunny visual memory vs. a clear, evening run

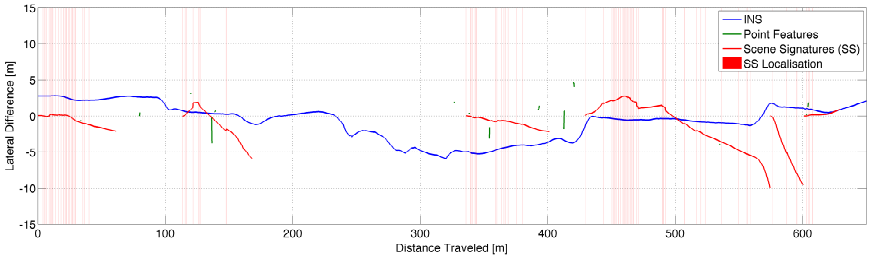


Live VO profile against groundtruth.

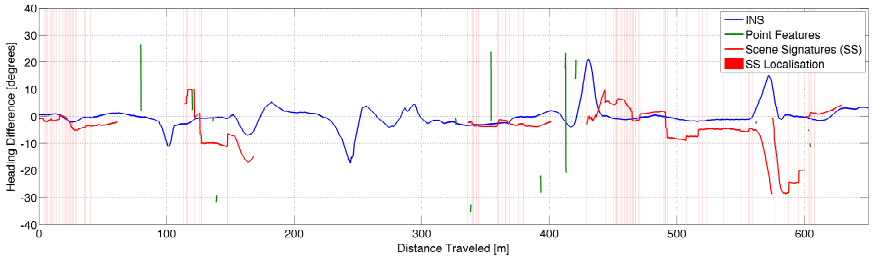


Number of feature matches.

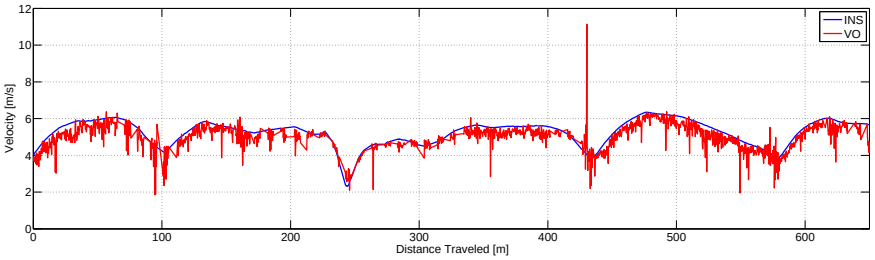
Figure D.2: Localisation results for the clear, evening run.



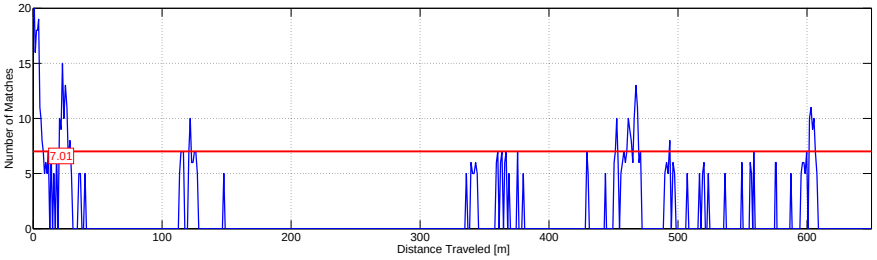
Lateral estimates for a sunny visual memory vs. a rainy, evening run.



Heading estimates for a sunny visual memory vs. a rainy, evening run

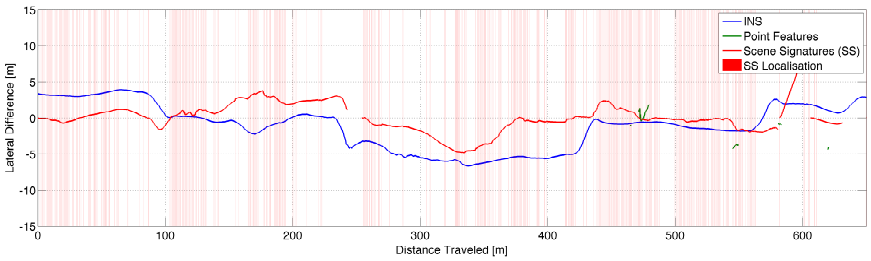


Live VO profile against groundtruth.

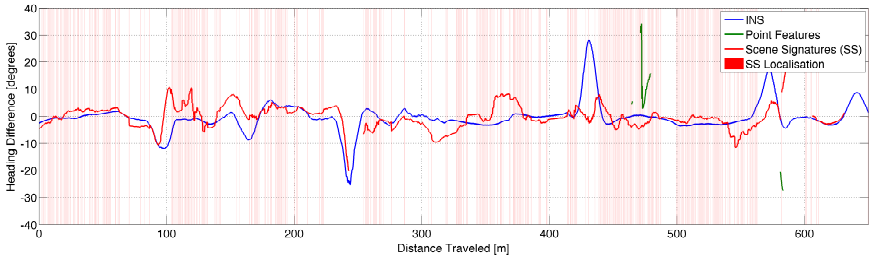


Number of feature matches.

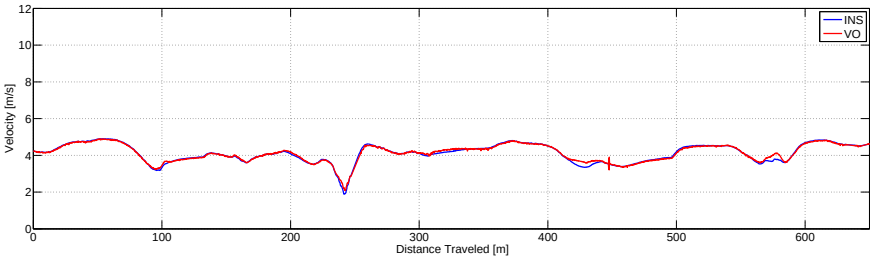
Figure D.3: Localisation results for the rainy, evening run.



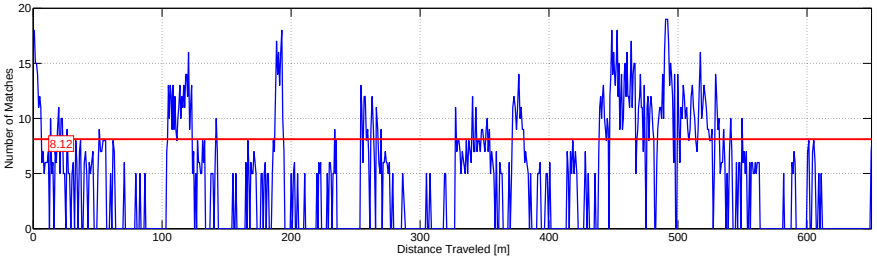
Lateral estimates for a sunny visual memory vs. a clear, snowy run.



Heading estimates for a sunny visual memory vs. a clear, snowy run

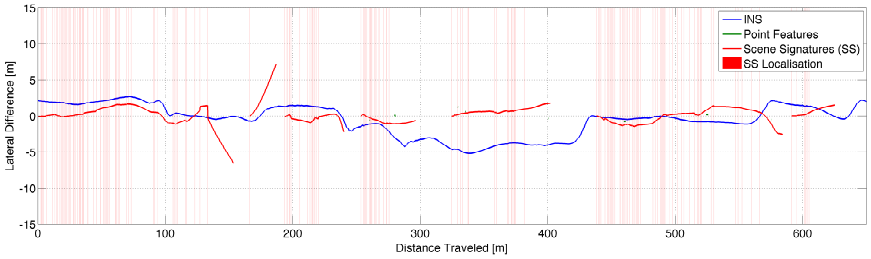


Live VO profile against groundtruth.

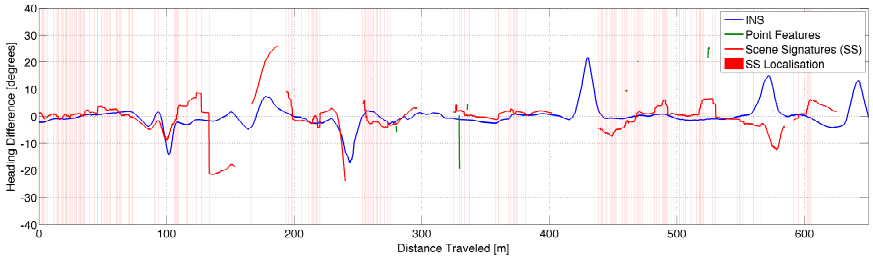


Number of feature matches.

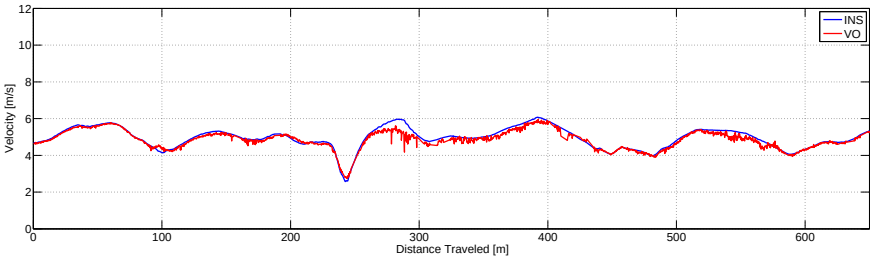
Figure D.4: Localisation results for the a clear, snowy run.



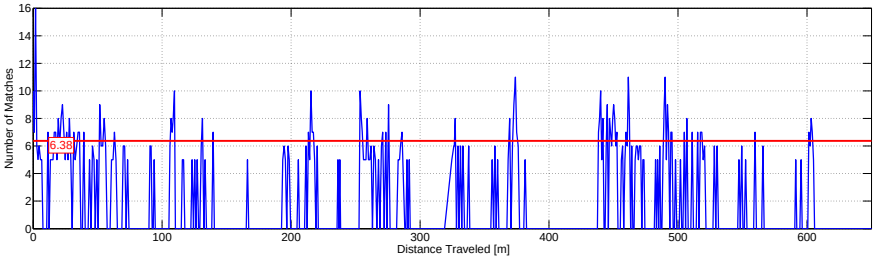
Lateral estimates for a sunny visual memory vs. a misty, snow run.



Heading estimates for a sunny visual memory vs. a misty, snow run



Live VO profile against groundtruth.



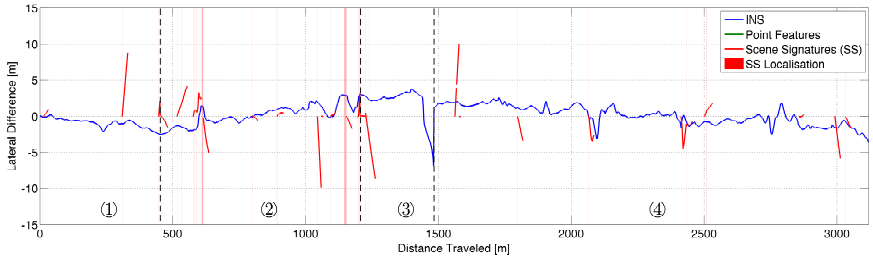
Number of feature matches.

Figure D.5: Localisation results for the misty, snow run.

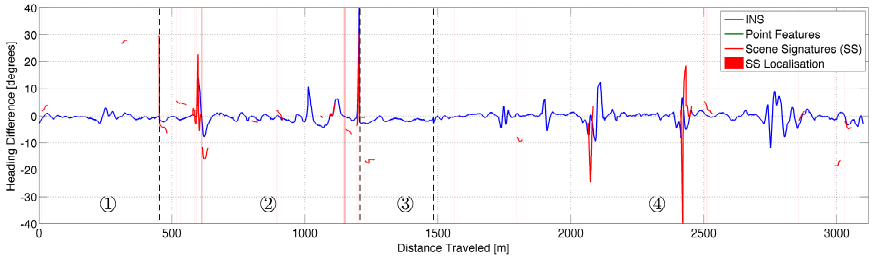
Appendix E

Oxford Illumination-Invariant Results

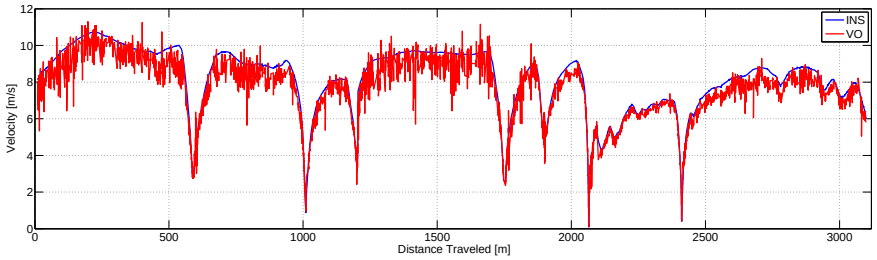
The following figures show the localisation results for Oxford using the illumination-invariant colour space presented in Chapter 4.



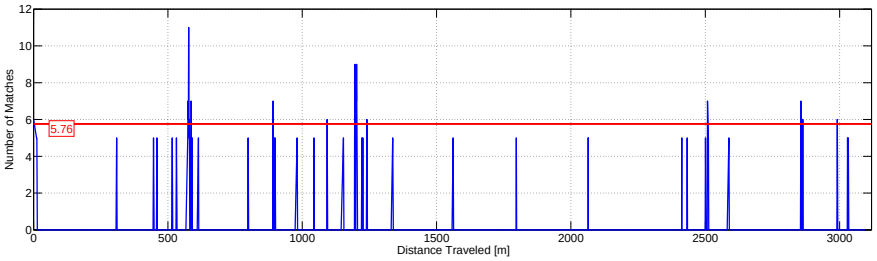
Lateral estimates for a sunny visual memory vs. a dark, night run.



Heading estimates for a sunny visual memory vs. a dark, night run.

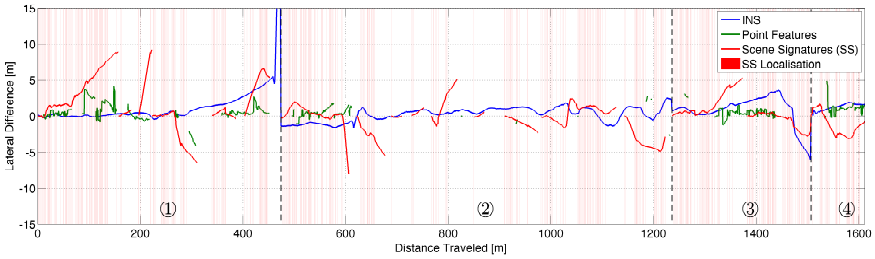


Live VO profile against groundtruth.

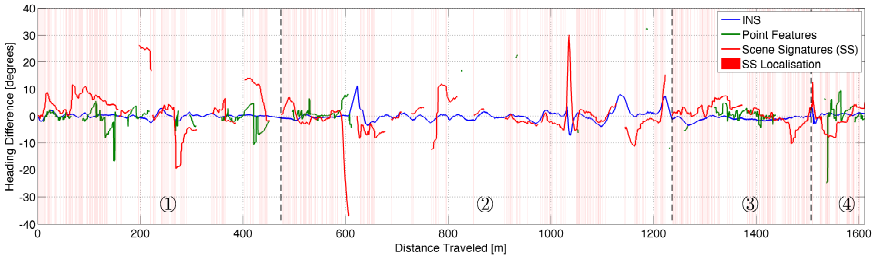


Number of feature matches.

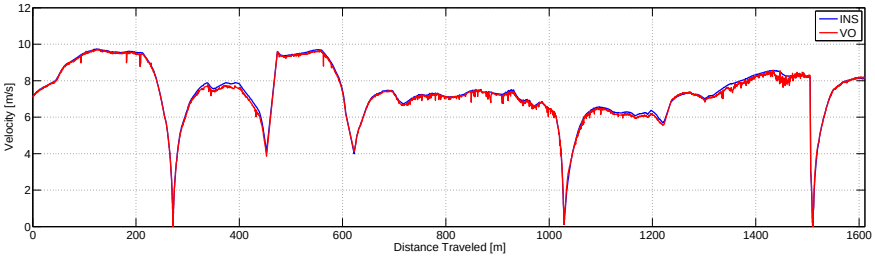
Figure E.1: Localisation results for the night run through segments 1-4 (Oxford).



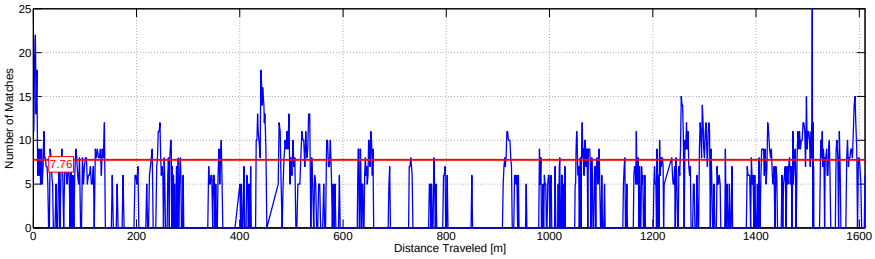
Lateral estimates for a sunny visual memory vs. a sunny run.



Heading estimates for a sunny visual memory vs. a sunny run.

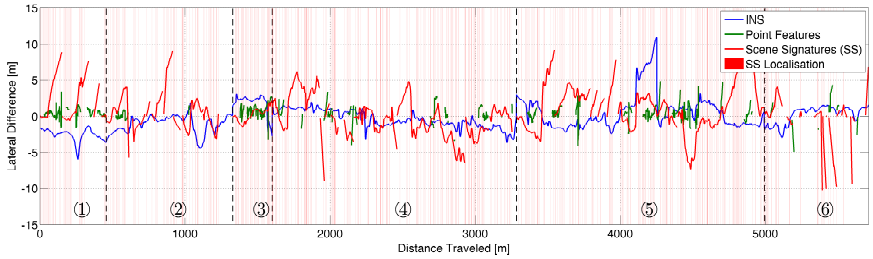


Live VO profile against groundtruth.

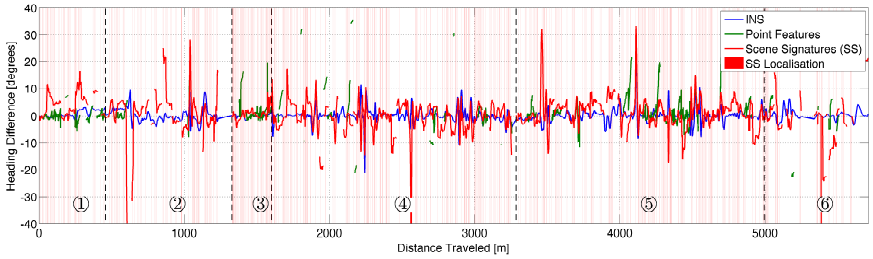


Number of feature matches.

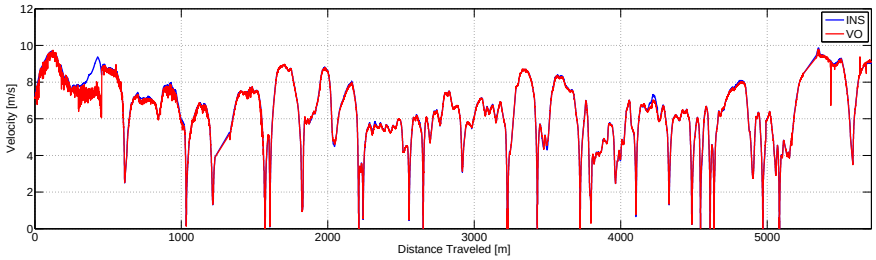
Figure E.2: Localisation results for a sunny run through segments 1-4 (Oxford).



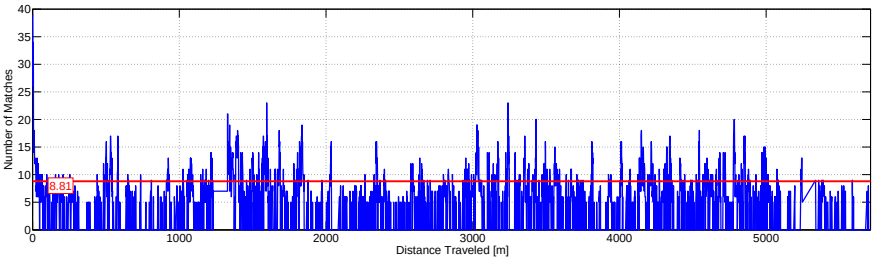
Lateral estimates for a sunny visual memory vs. a sunny run.



Heading estimates for a sunny visual memory vs. a sunny run.



Live VO profile against groundtruth.



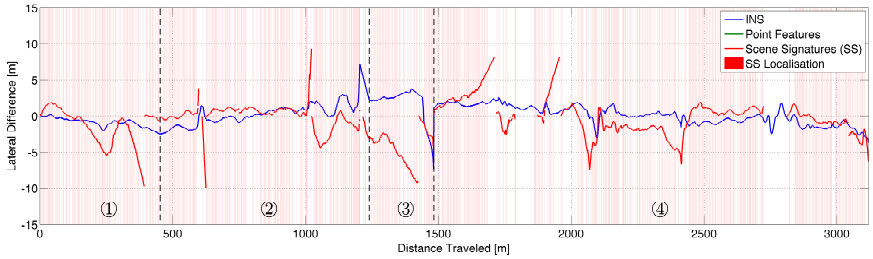
Number of feature matches.

Figure E.3: Localisation results for a sunny run through segments 1-6 (Oxford).

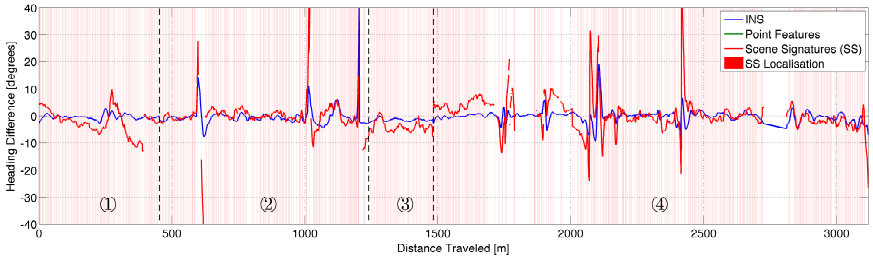
Appendix F

Oxford Localisation Results with Distraction Suppression

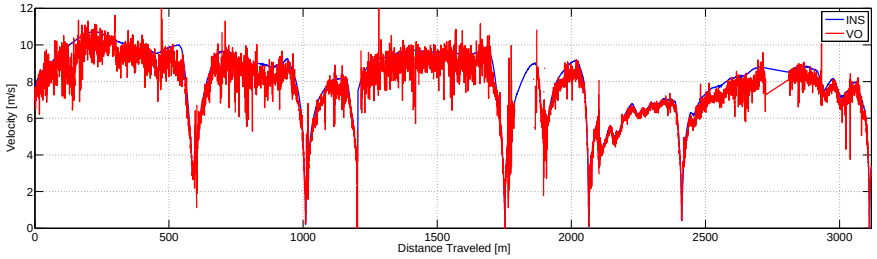
The following figures show the localisation results for Oxford using distraction suppression, which was introduced in Chapter 3.



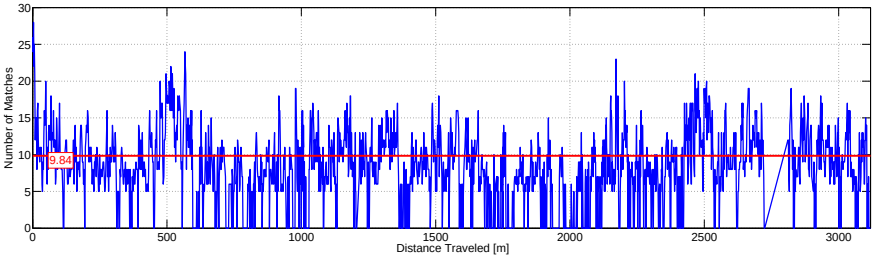
Lateral estimates for a sunny visual memory vs. a dark, night run.



Heading estimates for a sunny visual memory vs. a dark, night run.

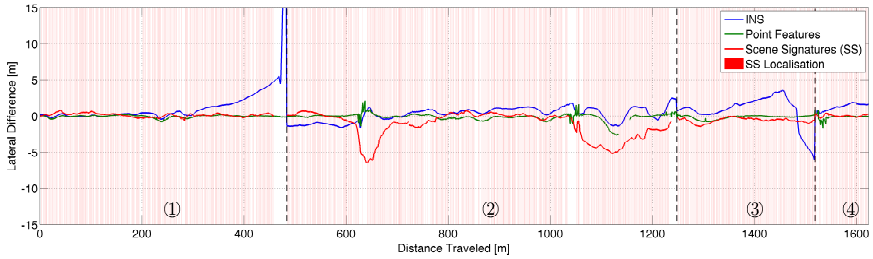


Live VO profile against groundtruth.

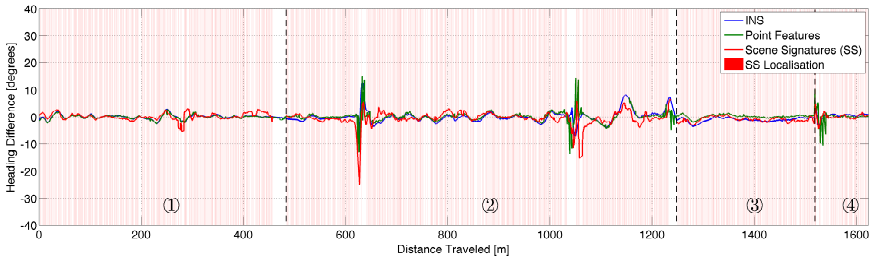


Number of feature matches.

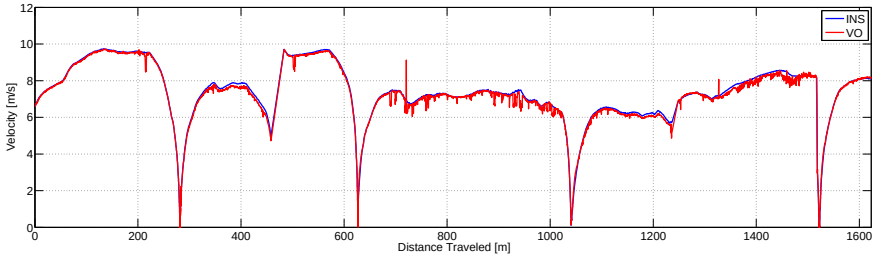
Figure F.1: Localisation results for the night run through segments 1-4 (Oxford).



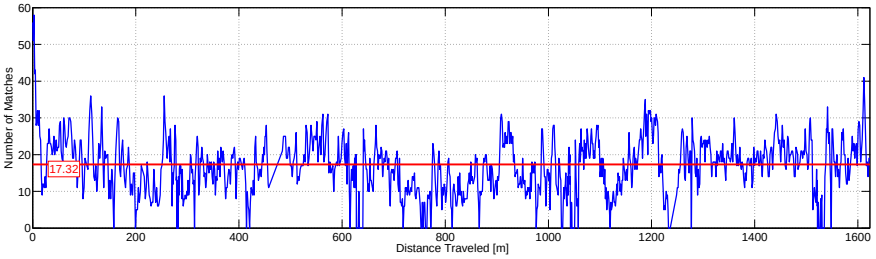
Lateral estimates for a sunny visual memory vs. a sunny run.



Heading estimates for a sunny visual memory vs. a sunny run.

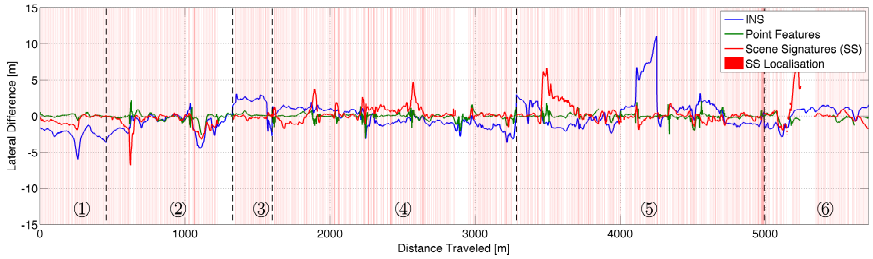


Live VO profile against groundtruth.

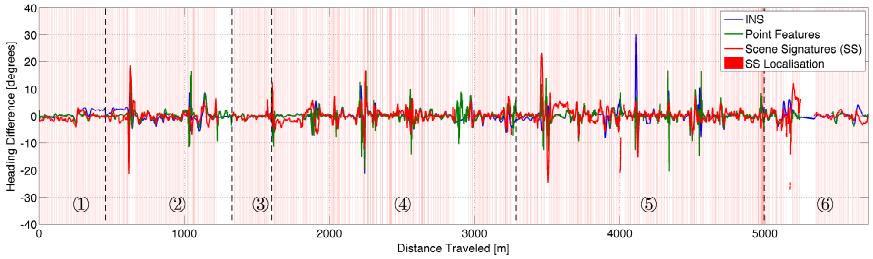


Number of feature matches.

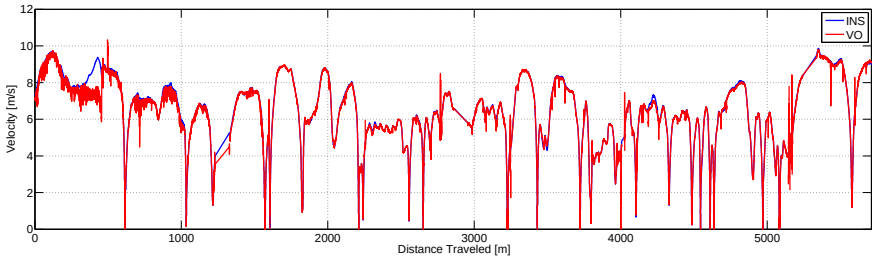
Figure F.2: Localisation results for a sunny run through segments 1-4 (Oxford).



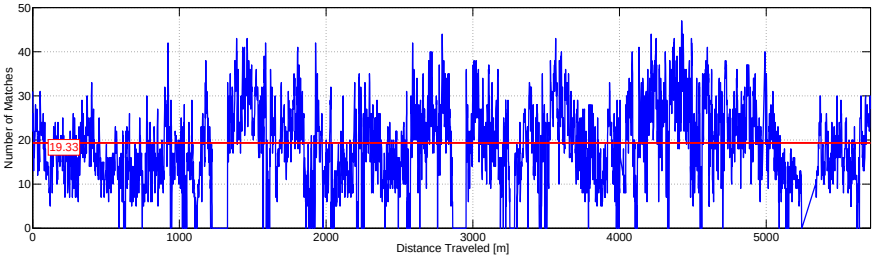
Lateral estimates for a sunny visual memory vs. a sunny run.



Heading estimates for a sunny visual memory vs. a sunny run.



Live VO profile against groundtruth.



Number of feature matches.

Figure F.3: Localisation results for a sunny run through segments 1-6 (Oxford).

Bibliography

Alcantarilla, P., Yebes, J., Almazán, J., and Bergasa, L. (2012). On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments. In *Proceedings of the International Conference on Robotics and Automation*, RiverCenter, Saint Paul, Minnesota, USA.

Anati, R., Scaramuzza, D., Derpanis, K., and Daniilidis, K. (2012). Robot localization using soft object detection. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, Minnesota, USA.

Atanasov, N., Zhu, M., Daniilidis, K., and Pappas, G. J. (2014). Semantic localisation via the matrix permanent. In *Proceedings of Robotics Science and Systems (RSS)*, Berkeley, USA.

Baldwin, I. and Newman, P. (2012). Road vehicle localization with 2d push-broom lidar and 3d priors. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Saint Paul, Minnesota, USA.

Bao, S. Y. and Savarese, S. (2011). Semantic structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2025–2032.

Barfoot, T., Forbes, J. R., and Furgale, P. (2011). Pose estimation using linearized rotations and quaternion algebra. *Acta astronautics*, 68(1-2):101–112.

Barfoot, T. and Furgale, P. (2014). Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Transactions on Robotics*, 30(3):679–693.

Bay, H., Ess, A., Tuytelaars, T., and Gool, L. (2008). Surf: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359.

Bayes, T. (1764). Essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*.

Bosse, M., Newman, P., Leonard, J., and Teller, S. (2004). Simultaneous localization and map building in large-scale cyclic environments using the atlas framework. *The International Journal of Robotics Research*, 23(12):1113–1139.

Bosse, M. and Zlot, R. (2008). Map matching and data association for large-scale two-dimensional laser scan-based slam. *The International Journal of Robotics Research*, 27(6):667–691.

Calonder, M., Lepetit, V., Ozuysal, M., Trzcinski, T., Strecha, C., and Fua, P. (2012). Brief: Computing a local binary descriptor very fast. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1281–1298.

Castle, R. O., Gawley, D. J., Klein, G., and Murray, D. W. (2007). Towards simultaneous recognition, localization and mapping for hand-held and wearable cameras. In *Proceedings of the IEEE International Conference on In Robotics and Automation (ICRA)*.

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

Churchill, W. and Newman, P. (2012). Practice makes perfect? managing and lever-

aging visual experiences for lifelong navigation. In *Proceedings of the International Conference on Robotics and Automation*, Saint Paul, Minnesota, USA.

Corke, P., Paul, R., Churchill, W., and Newman, P. (2013). Dealing with Shadows: Capturing Intrinsic Scene Appearance for Image-based Outdoor Localisation. *IEEE International Conference on Intelligent Robots and Systems*.

Cummins, M. and Newman, P. (2007). Probabilistic appearance based navigation and loop closing. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*.

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.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 886–893, San Diego, California, USA.

Davison, A. and Murray, D. (2002). Simultaneous localization and map-building using active vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7).

Davison, A., Reid, I., Motlon, N., and Stasse, O. (2007). Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6).

Denham, W. and Pines, S. (1966). Sequential estimation when measurement function nonlinearity is comparable to measurement error. *American Institute of Aeronautics and Astronautics*, 4(6):1071–1076.

Dickmanns, E. D. and Zapp, A. (1987). Autonomous high speed road vehicle guidance by computer vision. In *10th Triennial World Congress of the International Federation of Automatic Control*.

Doersch, C., Singh, S., Gupta, A., Sivic, J., and Efros, A. (2012). What makes paris look like paris? *ACM Transactions on Graphics*.

Durrant-Whyte, F. and Bailey, T. (2006). Simultaneous localization and mapping: part i. In *Robotics & Automation Magazine*, volume 13, pages 99–110. IEEE.

Durrant-Whyte, H. F. (1988). Uncertain geometry in robotics. *IEEE Transactions on Robotics and Automation*, 4(1):23–31.

Ebner, M. (2007). *Color Constancy*. Wiley-IS&T Series in Imaging Science and Technology.

Ess, A., Schindler, K., Leibe, B., and van Gool, L. (2010). Object detection and tracking for autonomous navigation in dynamic environments. *International Journal of Robotics Research*, 29.

Euler, L. (1770). Problema algebraicum ob affectiones prorsus singulares memorabile. *Commentatio 407 indicis Enestroemiani, Novi commentarii academiae scientiarum Petropolitanae*.

Finlayson, G., Drew, M., and Lu, C. (2004). Intrinsic images by entropy minimization. *Computer Vision-ECCV 2004*, pages 582–595.

Finlayson, G., Hordley, S., Lu, C., and Drew, M. (2006). On the removal of shadows from images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(1):59–68.

Furgale, P. (2011). *Extensions to the visual odometry pipeline for the exploration of planetary surfaces*. PhD thesis, University of Toronto Institute for Aerospace Studies.

Furgale, P. and Barfoot, T. (2010). Visual teach and repeat for long-range rover autonomy. *Journal of Field Robotics, special issue on "Visual mapping and navigation outdoors"*, 27(5):534–560.

Geiger, A., Roser, M., and Urtasun, R. (2010). Efficient large-scale stereo matching. In *Asian Conference on Computer Vision (ACCV)*, Queenstown, New Zealand.

Gelb, A. (1974). *Applied Optimal Estimation*. MIT Press.

Grewal, M. S. and Andrews, A. P. (2010). Applications of kalman filtering in aerospace 1960 to the present. *IEEE Control Systems Magazine*, pages 69–78.

Guo, R., Dai, Q., and Hoiem, D. (2011). Single-image shadow detection and removal using paired regions. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2033–2040. IEEE.

Haritaoglu, I., Harwood, D., and Davis, L. (2000). W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–830.

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151.

Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.

Hayman, E. and olof Eklundh, J. (2003). Statistical background subtraction for a mobile observer. In *IEEE International Conference on Computer Vision*.

Horbert, E., Rematas, K., and Leibe, B. (2011). Level-set person segmentation and tracking with multi-region appearance models and top-down shape information. In *Proceedings of the International Conference on Computer Vision*.

Huber, P. (1981). *Robust Statistics*. John Wiley and Sons, New York.

Irani, M. and Anandan, P. (1998). A unified approach to moving object detection in 2d and 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ito, K. and Xiong, K. (1999). Gaussian filters for nonlinear filtering problems. *IEEE Trans. on Automatic Control*, 45(5):910–927.

Jazwinski, A. H. (1970). *Stochastic Processes and Filtering Theory*. New York: Academic Press.

Julier, S., Uhlmann, J., and Durrant-Whyte, H. (1995). A new approach for filtering nonlinear systems. In *Proc. of the American Control Conf.*, volume 3, pages 1628–1632, Seattle WA, USA.

Kaess, M., Johannson, H., Roberts, R., Ila, V., Leonard, J., and Dellaert, F. (2012). isam2: Incremental smoothing and mapping using the bayes tree. *Internatioanl Journal of Robotics Research*, 31(2):216–235.

Kaess, M., Ranaganthan, A., and Dellaert, F. (2008). isam: incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378.

Kaestner, R., Engelhard, N., Triebel, R., and Siegwart, R. (2010). A bayesian approach to learning 3d representations of dynamic environments. In *Proceedings of the 12th International Symposium on Experimental Robotics*.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering (Transactions of the American Society of Mechanical Engineers)*, 82.

Kalman, R. E. and Bucy, R. S. (1961). New results in linear filtering and prediction theory. *Journal of Fluids Engineering*, 83(1).

Ko, D. W., Yi, C., and Suh, I. H. (2013). Semantic mapping and navigation: A bayesian approach. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS)*, pages 2630–2636.

Konolige, K., Bowman, J., Chen, J., Mihelich, P., Calonder, M., Lepetit, V., and Fua, P. (2010). View-based maps. *The International Journal of Robotics Research*, 29(8):941–957.

Kwatra, V., Han, M., and Dai, S. (2012). Shadow removal for aerial imagery by information theoretic intrinsic image analysis. In *Computational Photography (ICCP), 2012 IEEE International Conference on*, pages 1–8. IEEE.

Lategahn, H., Beck, J., Kitt, B., and Stiller, C. (2013). How to learn an illumination robust image feature for place recognition. In *IEEE Intelligent Vehicles Symposium*, Gold Coast, Australia.

Lefebvre, T., Bruyninckx, H., and De Schutter, J. (2001). Kalman filters for nonlinear systems: a comparison of performance. *Int'l. Journal of Control*, 77(7):639–653.

Leibe, B., Schindler, K., Cornelis, N., and van Gool, L. (2008). Coupled detection and tracking from static cameras and moving vehicles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1683–1698.

Lenz, P., Ziegler, J., Geiger, A., and Roser, M. (2011). Sparse scene flow segmentation for moving object detection in urban environments. In *IEEE Intelligent Vehicles Symposium*, pages 926–932.

Leonard, J. and Durrant-Whyte, H. (1991). Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation*.

Levenberg, K. (1944). A method for the solution of certain non-linear problems in least squares. *The Quarterly of Applied Mathematics*, 2:164–168.

Levinson, J., Montemerlo, M., and Thrun, S. (2010). Map-based precision vehicle localization in urban environments. In *Proceedings of Robotics Science and Systems*.

Li, Y., Wu, B., and Nevatia, R. (2008). Human detection by searching in 3d space using camera and scene knowledge. In *International Conference on Pattern Recognition*.

Liu, C. (2009). *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, MIT.

Lowe, D. (2004). Distinctive image features from scale-invariant key points. *International Journal of Computer Vision*, 60(2):91–110.

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 *Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation*, Hong Kong, China.

Maddern, W., Stewart, A., and Newman, P. (2014b). Laps-ii: 6-dof day and night

visual localisation with prior 3d structure for autonomous road vehicles. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, Dearborn, MI, USA.

Maddern, W. and Vidas, S. (2012). Towards Robust Night and Day Place Recognition using Visible and Thermal Imaging. *Robotics Science and Systems*.

Mahalanobis, P. (1936). On the generalized distance in statistics. *Proccedings of the National Institute of Science*, 2(1):49–55.

Markov, A. A. (1906). Rasprostranenie zakona bol'shih chisel na velichiny, zavisyaschie drug ot druga. *Izvestiya Fiziko-matematicheskogo obschestva pri Kazanskom universitete, 2-ya seriya, tom*, 15(94):135–156.

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. doi:10.5244/C.16.36.

McManus, C. (2010). The unscented kalman filter for state estimation. Presented at the Simultaneous Localization and Mapping (SLAM) Workshop, 7th Canadian Conference on Computer Vision (CRV).

McManus, C., Churchill, W., Maddern, W., Stewart, A., and Newman, P. (2014a). Shady dealings: Robust, long-term visual localisation using illumination invariance. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China.

McManus, C., Churchill, W., Napier, A., Davis, B., and Newman, P. (2013). Distraction suppression for vision-based pose estimation at city scales. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany.

McManus, C., Furgale, P., Stenning, B., and Barfoot, T. D. (2012). Visual Teach and Repeat Using Appearance-Based Lidar. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*.

McManus, C., Upcroft, B., and Newman, P. (2014b). Scene signatures: Localised and point-less features for localisation. In *Proceedings of Robotics Science and Systems (RSS)*, Berkley, CA, USA.

McManus, C., Upcroft, B., and Newman, P. (2015). Learning place-dependent feature detectors for long-term, outdoor navigation. *In Preperation for Submission to Autonomous Robots, special issue on selected papers from RSS 2014*.

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

Milford, M. and Wyeth, G. (2012). Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Saint Paul, Minnesota, USA.

Moutarlier, P. and Chatila, R. (1989a). An experimental system for incremental environment modeling by an autonomous mobile robot. In *1st International Symposium on Experimental Robotics*, Montreal, Canada.

Moutarlier, P. and Chatila, R. (1989b). Stoachastic multisensory data fusion for mobile robot location and environment modeling. In *5th International Symposium on Robotics Research*, Tokyo, Japan.

Muller, D., Meuter, M., and Park, S.-B. (2008). Motion segmentation using interest points. In *IEEE Intelligent Vehicles Symposium*.

Murray, R. M., Li, Z., and Sastry, S. S. (1994). *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL, USA: CRC.

Namdev, R., Kundu, A., Krishna, K., and Jawahar, C. (2012). Motion segmentation of multiple objects from a freely moving monocular camera. In *Proceedings of the IEEE International Conference on Robotics and Automation*, RiverCenter, Saint Paul, Minnesota, USA.

Napier, A. and Newman, P. (2012). Generation and exploitation of synthetic overhead images for road vehicle localisation. In *Proceedings of the International Conference on Robotics and Automation*, RiverCenter, Saint Paul, Minnesota, USA.

Napier, A., Sibley, G., and Newman, P. (2010). Real-time bounded-error pose estimation for road vehicles using vision. In *Intelligent Transportation Systems Conference (ITSC)*.

Narayanan Sundaram, K. and Brox, T. (2011). Dense point trajectories by gpu-accelerated large displacement optical flow. In *European Conference on Computer Vision*.

Newman, P. (1999). *On the structures and solution of simultaneous localization and mapping problem*. PhD thesis, Australian Centre for Field Robotics, Sidney.

Newton, I. (1968). *The Mathematical paper of Isaac Newton, Vol 11*, chapter De analysi per aequationes infinitas, pages 207–247. Cambridge University Press.

Norgaard, N., Poulsen, N., and Ravn, O. (2000). New developments in state estimation for nonlinear systems. *Automatica*, 36(11):1627–1638.

Piccardi, M. (2004). Background subtraction techniques: a review. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*.

Piniés, P., Paz, L. M., Gálvez-López, D., and Tardós, J. D. (2010). Ci-graph simultaneous localisation and mappin for three-dimensional reconstruction of large and complex environments using a multicamera system. *Journal of Field Robotics*, 27(5):561–586.

Pink, O. and Stiller, C. (2010). Automated map generation form aerial images for precise vehicle localization. In *Intelligent Transportation Systems Conference (ITSC)*.

Pomerleau, D. (1989). Alvinn: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems*.

Potter, J. E. and Stern, R. G. (1963). Statistical filtering of space navigation measurements. In *Proceedings of the AIAA Guidance Control Conference*.

Raphson, J. (1690). *Analysis Aequationum universalis seu ad aequationes algebraicas resolvendas methodus generalis, et expedita, ex nova infinitarum serierum doctrina, deducta ac demonstrata*. London.

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

Ratnasingam, S. and McGinnity, T. (2012). Chromaticity space for illuminant invariant recognition. *IEEE Transactions on Image Processing*, 21.

Ren, Y., Chua, C.-S., and Ho, Y.-K. (2003). Statistical background modeling for non-stationary camera. *Pattern Recognition Letters*, 24.

Renato F. Salas-Moreno, Richard A. Newcombe, H. S. P. H. J. K. and Davison, A. J. (2013). Slam++: Simultaneous localisation and mapping at the level of

object. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Richardson, A. and Olson, E. (2013). Learning convolutional filters for interest point detection. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*.

Rodrigues, O. (1816). *Mémoire sur l'attraction des sphéroi"des*. PhD thesis, Corresp. sur l'École Royal Polytech.

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

Rosten, E., Reitmayr, G., and Drummond, T. (2005). Real-time video annotations for augmented reality. In *Advances in Visual Computing*.

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

Salas, J. and Tomatsi, C. (2011). People detection using color and depth images. *Lecture Notes in Computer Science - Pattern Recognition*, 6718:127–135.

Sheikh, Y., Javed, O., and Kanade, T. (2009). Background subtraction for freely moving cameras. In *IEEE 12th International Conference on Computer Vision*.

Sibley, G. (2007). *Long Range Stereo Data-fusion From Moving Platforms*. PhD thesis, University of Southern California.

Sibley, G., Mei, C., Reid, I., and Newman, P. (2010). Vast-scale outdoor navigation using adaptive relative bundle adjustment. *The Int. Journal of Robotics Research*, 29(8):958–980.

Singh, S., Gupta, A., and Efros, A. A. (2012). Unsupervised discovery of mid-level discriminative patches. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Smith, R., Self, M., and Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, pages 167–193.

Stewart, A. and Newman, P. (2012). Laps - localisation using appearance of prior structure: 6-dof monocular camera localisation using prior pointclouds. In *Proceedings of the International Conference on Robotics and Automation*, Saint Paul, Minnesota, USA.

Stillwell, J. (2008). *Naive Lie Theory*. Springer.

Strasdat, H. (2012). *Local Accuracy and Global Consistency for Efficient Visual SLAM*. PhD thesis, Imperial College London.

Strasdat, H., Montiel, J., and Davison, A. (2010). Real-time monocular slam: Why filter? In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2657–2664, Anchorage, Alaska, United States.

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

Swerling, P. (1958). A proposed stagewise differential correction procedure for satellite tracking and predicition. Technical report, Technical Report P-1292, RAND Corporation.

Taludker, A., Goldberg, S., Matthies, L., and Ansar, A. (2003). Real-time detection of moving objects in a dynamic scene from moving robotic vehicles. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotics and Systems*.

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

Thrun, S., Burgard, W., and Fox, D. (2005a). *Probabilistic Robotics*. The MIT Press.

Thrun, S., Fox, D., Burgard, W., and Dellaer, F. (2001). Robust monte carlo localization for mobile robots. *To appear in Artificial Intelligence*.

Thrun, S., Liu, Y., Koller, D., Ng, A. Y., Ghahramani, Z., and Durrant-Whyte, H. F. (2004). Simultaneous localisation and mapping with sparse extended information filters. *International Journal of Robotics Research*, 23(7).

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., and Mahoney, P. (2005b). Stanley: The robot that won the darpa grand challenge. *Journal of Field Robotics*.

Thrun, S. and Montermerlo, M. (2006). The graphslam algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 25(5–6):403–429.

Torralba, A., Murphy, K., and Freeman, W. (2007). Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):854–869.

Tsugawa, S., Yatabe, T., Hirose, T., and Matsumoto, S. (1979). An automobile with artifical intelligence. In *the International Joint Conference on Artifical Intelligence (IJCAI)*, pages 893–895, Tokyo, Japan.

Upcroft, B., McManus, C., Churchill, W., Maddern, W., and Newman, P. (2014). Lighting invariant urban street classification. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China.

Urmson, C., Anhalt, J., Bagnell, D., Baker, C., Bittner, R., Clark, M., Dolan, J., Duggins, D., Galatali, T., Geyer, C., Gittleman, M., Harbaugh, S., Hebert, M., Howard, T., Kolski, S., Kelly, A., Likhachev, M., McNaugton, M., Miller, N., Peterson, K., Pilnick, B., Rajkumar, R., Rybski, P., Salesky, B., Seo, Y.-W., Singh, S. Snider, J., Stentz, A., Whittaker, W., Wolkowicki, Z., and Ziglar, J. (2008). Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics, Special Issue on the 2007 DARPA Urban Challenge, Part 1*, 25(8):425–466.

Valgren, C. and Lilienthal, A. (2010). Sift, surf and seasons: Appearance-based long-term localisation in outdoor environments. *Robotics and Autonomous Systems*, (58):149–156.

von Hundelshausen, F. and Sukthankar, R. (2012). D-Nets: Beyond Patch-Based Image Descriptors. In *IEEE International Conference on Computer Vision and Pattern Recognition*.

Wang, D., Posner, I., and Newman, P. (2012). What could move? finding cars, pedestrians and bicyclists in 3d laser data. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Saint Paul, Minnesota, USA.

Wedel, A., Brox, T., Vaudrey, T., Rabe, C., and Franke, U. (2011). Stereoscopic scene flow computation for 3d motion understanding. *International Journal of Computer Vision*, 35(1):29–51.

Williams, S. (2001). *Efficient solutions to autonomous mapping and navigation problems*. PhD thesis, University of Sydney, Australian Centre for Field Robotics.

Wren, C., Azarbayejani, A., Darell, T., and Pentland, A. (1997). Pfinder: Real-time tracking of human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785.

Yi, C., Suh, I. H., Lim, G. H., and Choi, B.-U. (2009). Active-semantic localization with a single consumer-grade camera. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 2161–2166.

Yuan, C., Medioni, G., Kang, J., and Cohen, I. (2007). Detecting motion region in the presence of strong parallax from a moving camera by multiview geometric constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhang, Z. (1997). Parameter estimation techniques: A tutorial with application to conice fitting. *Image and Vision Computing*, 15(1):59–76.

Zhu, J., Samuel, K., Masood, S., and Tappen, M. (2010). Learning to recognize shadows in monochromatic natural images. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 223–230. IEEE.