On learning pseudo-sensors to improve egomotion estimation for mobile autonomy

by

Valentin Peretroukhin

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Institute for Aerospace Studies
University of Toronto

Abstract

On learning pseudo-sensors to improve egomotion estimation for mobile autonomy

Valentin Peretroukhin
Doctor of Philosophy
Graduate Department of Institute for Aerospace Studies
University of Toronto
2019

The ability to estimate *egomotion*, that is, to track one's own pose through an unknown environment, is at the heart of safe and reliable mobile autonomy. By inferring pose changes from sequential sensor measurements, egomotion estimation forms the basis of mapping and navigation pipelines, and permits mobile robots to self-localize within environments where external localization sources are intermittent or unavailable. Visual and inertial egomotion estimation, in particular, have become ubiquitous in mobile robotics due to the availability of high-quality, compact, and inexpensive sensors that capture rich representations of the world. To remain computationally tractable, 'classical' visual-inertial pipelines (like visual odometry and visual SLAM) make simplifying assumptions that, while permitting reliable operation in ideal conditions, often lead to systematic error. In this thesis, we present several data-driven learned *pseudo-sensors* that serve to complement conventional pipelines by inferring latent information from the same data stream. Our approach retains much of the benefits of traditional pipelines, while leveraging high-capacity hyper-parametric models to extract complementary information that can be used to improve uncertainty quantification, correct for systematic bias, and improve robustness to difficult-to-model deleterious effects. We validate our pseudo-sensors on several kilometres of sensor data collected in sundry settings such as urban roads, indoor labs, and planetary analogue sites in the Canadian high arctic.

Epigraph

A little learning is a dangerous thing;
drink deep, or taste not the Pierian
spring: there shallow draughts
intoxicate the brain, and drinking
largely sobers us again.

Alexander Pope

The universe is no narrow thing and the order within it is not constrained by any latitude in its conception to repeat what exists in one part in any other part. Even in this world more things exist without our knowledge than with it and the order in creation which you see is that which you have put there, like a string in a maze, so that you shall not lose your way. For existence has its own order and that no man's mind can compass, that mind itself being but a fact among others.

Cormac McCarthy

Elephants don't play chess.

Rodney Brooks

To all those who encouraged (or, at least, *never discouraged*) my intellectual wanderlust.

Acknowledgements

This document would not have been possible without the generous support and guidance of my supervisor[1], the perennial love of my family and friends[2], and the limitless patience of my lab mates[3]. Thank you all.

[1] as well as all of my collaborators and academic mentors
[2] especially the support and encouragement of Elyse
[3] in humouring my insatiable need for debate and banter

Contents

Notation

a	:	Symbols in this font are real scalars.
$\mathbf{a}$	:	Symbols in this font are real column vectors.
$\mathbf{A}$	:	Symbols in this font are real matrices.
$\mathcal{N}(\boldsymbol{\mu}, \mathbf{R})$	:	Normally distributed with mean $\boldsymbol{\mu}$ and covariance $\mathbf{R}$.
$E[\cdot]$	:	The expectation operator.
$\underrightarrow{\boldsymbol{\mathcal{F}}}_a$	:	A reference frame in three dimensions.
$(\cdot)^\wedge$	:	An operator associated with the Lie algebra for rotations and poses. It produces a matrix from a column vector.
$(\cdot)^\vee$	:	The inverse operation of $(\cdot)^\wedge$
$\mathbf{1}$	:	The identity matrix.
$\mathbf{0}$	:	The zero matrix.
$\mathbf{p}_a^{c,b}$	:	A vector from point b to point c (denoted by the superscript) and expressed in $\underrightarrow{\boldsymbol{\mathcal{F}}}_a$ (denoted by the subscript). This vector can be in homogenous coordinates depending on context.
$\mathbf{C}_{a,b}$	:	The 3×3 rotation matrix that transforms vectors from $\underrightarrow{\boldsymbol{\mathcal{F}}}_b$ to $\underrightarrow{\boldsymbol{\mathcal{F}}}_a$: $\mathbf{p}_a^{c,b} = \mathbf{C}_{a,b}\mathbf{p}_b^{c,b}$.
$\mathbf{T}_{a,b}$	:	The 4×4 transformation matrix that transforms homogeneous points from $\underrightarrow{\boldsymbol{\mathcal{F}}}_b$ to $\underrightarrow{\boldsymbol{\mathcal{F}}}_a$: $\mathbf{p}_a^{c,a} = \mathbf{T}_{a,b}\mathbf{p}_b^{c,b}$.

Chapter 1

Introduction

> To be sure, a writer cannot begin with a thesis; he must rather use his writerly sensitivity to intuit what is going on, even if he cannot understand its implications.
>
> GARY MORSON, *How the great truth dawned*

1.1 Autonomy and humanity through the ages

Autonomous systems, in some form, have been imagined and realized for the bulk of recorded history. In ancient Greek mythology, the god Hephaestus, the 'patron of invention and technology' (Mayor, 2019) was said to create talking mechanical handmaidens, while early Hindu and Buddhist texts tell of *yantakara* that lived in Greece and created machines that helped in trade and farming. The secret methods of the *yantakara* (the early 'roboticists') were closely guarded, and mechanical assassins were said to pursue and kill any person who revealed their techniques[1] (Mayor, 2019).

Figure 1.1: A 'robot' rebellion from Karel Capek's 1920 play, *Rossum's Universal Robots*.

[1] Please be careful distributing this thesis.

Since the industrial revolution, the idea of an autonomous machine–one that requires no, or very minimal, human intervention or oversight to operate–has been imagined in different ways. Depending on one's perspective, autonomous machines have perennially promised to either usher in a utopia of freedom, or threatened to bring about an age of job loss and social upheaval that worsens socioeconomic divisions. Much like the Luddites of the 19th century, the social critics of the 21st century have continued the dialectic to understand the social ramifications of modern *yantakara* and their newly-created autonomous hand-maidens.

These controversial origins are embedded even within the modern name of for the academic field, *robotics*. The word *robot* comes from the title of a science fiction play, R.U.R.: Rossum's Universal Robots, written by the Czech playwright Karel Capek in 1920 (see Figure 1.2). In naming the play, the word *robot* was derived from the Slavic term for slave, *rab*, and its Czech derivative for serf labour, *rabota*, while the name *Rossum* was inspired by the Czech word for reason, or intellect. Indeed, the concept of enslaved or embodied intelligence is at the heart of modern definitions of the discipline of robotics (Redfield, 2019). Much of the popular culture surrounding robots (e.g., Shelley's *Frankenstein*, Asimov's *I, Robot*, Kubrick's and Clarke's *2001: A Space Odyssey*) also paints a complicated picture of humanity's relationship with such enslaved machines. In this dissertation, we focus on improving a specific part of a modern *mobile* autonomy pipeline, while minimizing the use of term *robot* to avoid maelstrom of philosophical and ethical problems that it connotes. We hope this work aids the march of technological progress towards a future which finds some Hegelian synthesis of autonomy and humanity—a future in which human-in-the-loop autonomous systems augment and improve the lot of many people while still negotiating and constantly considering the social costs that come with technological innovation.

1.2 Mobile Autonomy and State Estimation

While the looms and railroads of the industrial revolution were spurred by the discovery of steam engines and electricity, modern *mobile* autonomy was largely born out of the technological arms race of the cold war and the constraints and challenges associated with long-distance flight and extraterrestrial travel (see Grewal and Andrews (2010) for a history of one of the seminal algorithms in mobile autonomy, the Kalman filter). Indeed, much of the work on modern perception algorithms has its origins in the automated compilation of cold-war-era reconnaissance imagery and the design of extraterrestrial rovers like the Mars Exploration Rovers, *Spirit* and *Opportunity* (Scaramuzza and Fraundorfer, 2011a). Similarly, much of the planning and control algorithms originate in American and Soviet defence-funded research (Nilsson, 1984; Thrun et al., 2006).

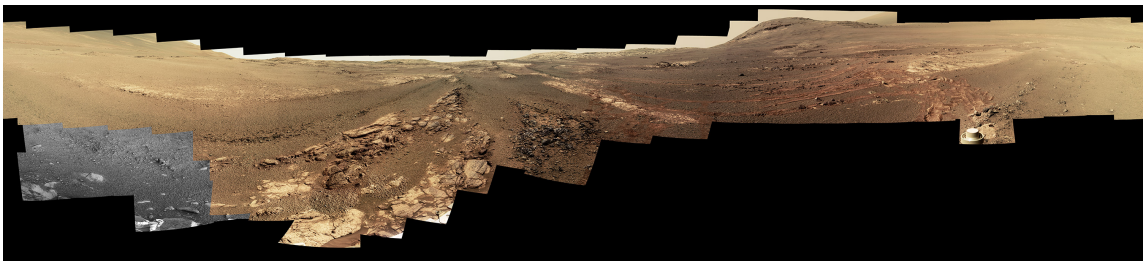
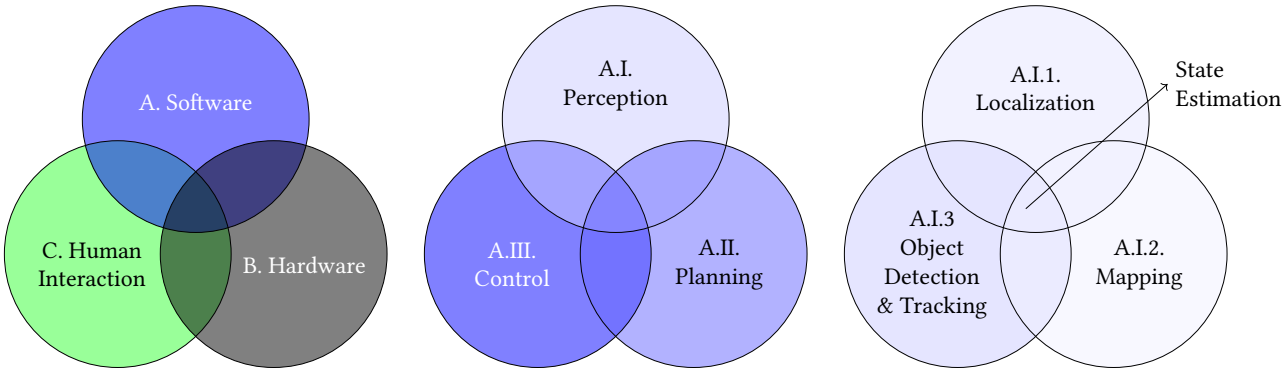


Figure 1.2: The last 360 degree panorama taken by the PanCam apparatus of the Mars Exploration Rover, *Opportunity*, at its final resting place on Mars, the western rim of the Endeavour Crater. Contact with *Opportunity* was lost shortly after this was captured, due to a severe dust storm (credit: NASA/JPL-Caltech/Cornell/ASU).



(a) Research strands in mobile autonomy.

(b) The components of software research.

(c) The components of perception.

Figure 1.3: Venn diagrams of modern mobile autonomy.

Once confined to carefully-controlled factory floors, autonomous mobile platforms have now began to show great promise in improving the safety of human transport, reducing the burden of repetitive, arduous jobs, and more efficiently leveraging limited resources for environmental monitoring. This newly-realized potential can be attributed to several factors: improvements in the cost and efficiency of computing devices (in terms energy efficiency, processing power, and overall size), the availability of relatively cheap, compact, high-quality sensors and rapid prototyping tools, and the development of open-source hardware, software platforms and datasets (e.g., the Robot Operating System, the KITTI Self-Driving Car dataset (Geiger et al., 2013a)).

Despite decades of research, mobile autonomy as a field still has nebulous demarcations between subfields. We have attempted to provide a general overview of the field through a series of Venn diagrams in Figure 1.3. At the highest level, the field can be roughly divided into those researchers who study and develop software, those who study and develop hardware, and those who study and analyze the interaction between autonomous systems (composed of both software and hardware) and humans (Figure 1.3a). There is, of course, a plethora

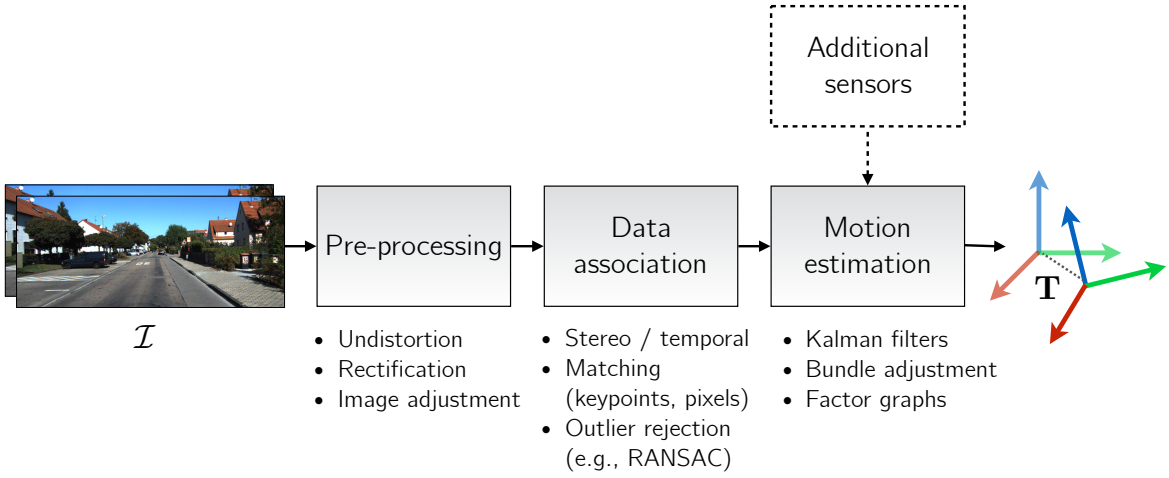


Figure 1.4: A 'classical' visual odometry pipeline consists of several distinct components that have interpretable inputs and outputs.

of overlap between all three of these rough categories. Within the software realm, there has historically been a distinction between those who study algorithms that deal with the perception of the interoceptive and exteroceptive data, those who study how to use that data to plan action, and those who study how to use those plans to control a system to execute that action (Figure 1.3b).

Within perception, which is the focus of this dissertation, there are three general directions of research: localization, mapping, and object detection and tracking. Localization and mapping can refer to self-localization, or egomotion estimation, which deals with the problem of estimating the pose of a moving platform through an unknown world, SLAM, simultaneous localization and mapping, which deals with the former problem and mapping simultaneously or finally, it can refer to localization within a known map given some hitherto unseen observation of that environment. The field of object detection and tracking (whether that be static objects like stop signs, or moving objects like humans, animals or vehicles) uses much of the same underlying mathematics as the former two, but has historically been a separate strand of research. Broadly, the overlap of all three of these pursuits within the field of autonomy and robotics is referred to as *state estimation* (Barfoot, 2017). To provide context for the central concept of this thesis, a *pseudo-sensor*, we will first outline a traditional approach to state estimation.

1.3 The *State* of State Estimation

Central to *classical* state estimation algorithms (which, in this context, refers to the bulk of state estimation research published during what Cadena et al. (2016) call the *classical* and *algorithmic-analysis* ages of SLAM research between 1986 and 2015) is the idea of a pipeline. A pipeline consists several distinguishable blocks that have interpretable inputs and outputs. By carefully processing information contained within sensor data, pipelines facilitate the construction of complex state estimation architectures that can fuse observations from sensors of varied modality to create rich models of the external world and infer the state of a mobile platform within it. This dissertation focuses on egomotion estimation: the problem of accurately and consistently estimating the relative pose of a moving platform. For this task, a variety of different sensors may be useful (e.g., lidar, stereo cameras, or inertial measurement units), and each may allow for various components of a state estimation pipeline. For cameras, egomotion estimation is typically referred to as *visual odometry* or VO for short. In this thesis, we will largely deal with the improvement of a 'classical' VO pipeline– we illustrate its major components in Figure 1.4.

Modern VO pipelines (e.g., Leutenegger et al. (2015); Cvišić and Petrović (2015); Tsotsos et al. (2015a)) have achieved impressive localization accuracy on trajectories spanning several kilometres by carefully extracting and tracking sparse visual features (using *hand-crafted* algorithms) across consecutive images. Simultaneously, significant effort has gone to developing localization pipelines that eschew sparse features in favour of *dense* visual data (Alcantarilla and Woodford, 2016; Forster et al., 2014a,a), typically relying on loss functions that use direct pixel intensities.

In the last five years, a significant part of the visual state estimation literature has also focused on the idea of replacing classical pipelines with parametric modelling through deep convolutional neural networks (CNNs) and data-driven training. Although initially developed for image classification (LeCun et al., 2015), CNN-based measurement models have been applied to numerous problems in geometric state estimation (e.g., homography estimation (DeTone et al., 2016), single image depth reconstruction (Garg et al., 2016), camera relocalization (Kendall and Cipolla, 2016), place recognition (Sünderhauf et al., 2015)). A number of recent CNN-based approaches have also tackled the problem of egomotion estimation, often purporting to obviate the need for classical visual localization pipelines by learning pose changes *end-to-end*, directly from image data (e.g., Melekhov et al. (2017), Handa et al. (2016), Oliveira et al. (2017)).

Despite this surge of excitement, significant debate has emerged within the robotics and computer vision communities regarding the extent to which deep models should replace ex-

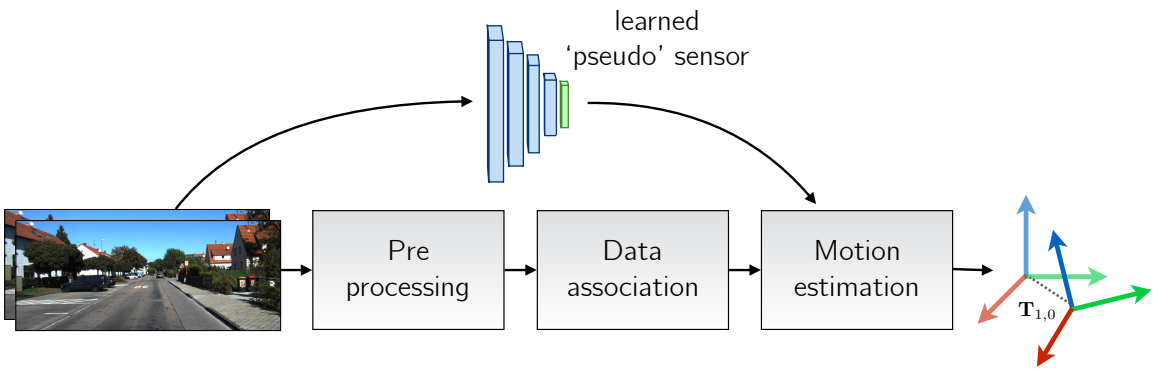


Figure 1.5: A learned *pseudo-sensor* extracts latent information from the same data stream.

isting geometric state estimation algorithms. Owing to their representational power, deep models may move the onerous task of selecting 'good' (i.e., robust to environmental vagaries and sensor motion) visual features from the roboticist to the learned model. By design, deep models also provide a straight-forward formulation for using *dense* data while being flexible in their loss function, and taking full advantage of modern computing architecture to minimize run-time. Despite these potential benefits, current deep regression techniques for state estimation often generalize poorly to new environments, come with few analytical guarantees, and provide only point estimates of latent parameters. Indeed, the most accurate visual egomotion pipelines[2] largely those based on carefully selected sparse features. Furthermore, there is recent empirical evidence (Zhou et al., 2019) that suggests that designing a pipeline with interpretable components (e.g., optical flow, scene segmentation) is crucial to generalization on various visual tasks. We summarize the benefits and detriments to using deep models (as opposed to classical pipelines) in Table 1.1.

1.4 The Learned Pseudo-Sensor

As state estimation enters the robust-perception age (Cadena et al., 2016), algorithms that work in limited contexts will need to be adapted and augmented to ensure they can operate over longer time-periods, and through sundry environments. Towards this end, we introduce the paradigm of the *learned pseudo-sensor*. Learned pseudo-sensors allow one to retain the benefits of classical state estimation pipelines while leveraging the representational power of modern data-drive learning techniques. Instead of completely replacing the classical pipeline, herein we present four ways in which machine learning can be used to train a

[2]Cite KITTI leaderboard

Table 1.1: A comaprison of pipelines and end-to-end deep models for visual egomotion estimation.

	Classical Pipelines	**Deep Models**
Maturity	Decades of literature & domain knowledge	Nascent with few uses in mobile autonomy
Interpretability	Good, each component has interpretable input and output	Poor, often with no interpretable intermediate outputs
Uncertainty	Foundational to *probabilistic robotics*	Few nascent methods (Monte-carlo Dropout (Gal and Ghahramani, 2016), Bootstrap (Osband et al., 2016))
Robustness	Empirically generalizable (Zhou et al., 2019)	Highly dependant on training data
Flexibility	Limited by ingenuity of designer	Limited by training data

hyper-parametric model (a 'pseudo-sensor') that extracts latent information from an existing visual data stream. By fusing the output of these sensors with the output of the pipeline, we can make the final egomotion estimates more accurate and more robust to difficult-to-model effects (Figure 1.5). To accomplish this fusion, we rely on two approaches. The first, which is used in Predictive Robust Estimation (PROBE, and its follow up PROBE-GK, Chapter 4), treats the pseudo-sensor as a heteroscedastic noise model that can be incorporated into a maximum-likelihood loss. The uncertainty quantification provided by this pseudo-sensor is used to re-weight a loss which can then be minimized through traditional non-linear optimization routines during test-time. The second approach (used by Sun-BCNN, DPC-Net, and HydraNet, **??????** respectively) produces geometric quantities (probabilistic estimates of an illumination source, $\mathrm{SE}(3)$ corrections to existing egomotion estimates, and independent probabilistic rotation estimates, respectively), that can be fused with the original pipeline through a factor graph optimization routine.

1.5 Original Contributions

This dissertation consists of several published contributions under the umbrella of a *learned pseudo-sensor* that improves a canonical visual egomotion pipeline. Before detailing each pseudo-sensor, we present some mathematical foundations (Chapter 2) and a common base-

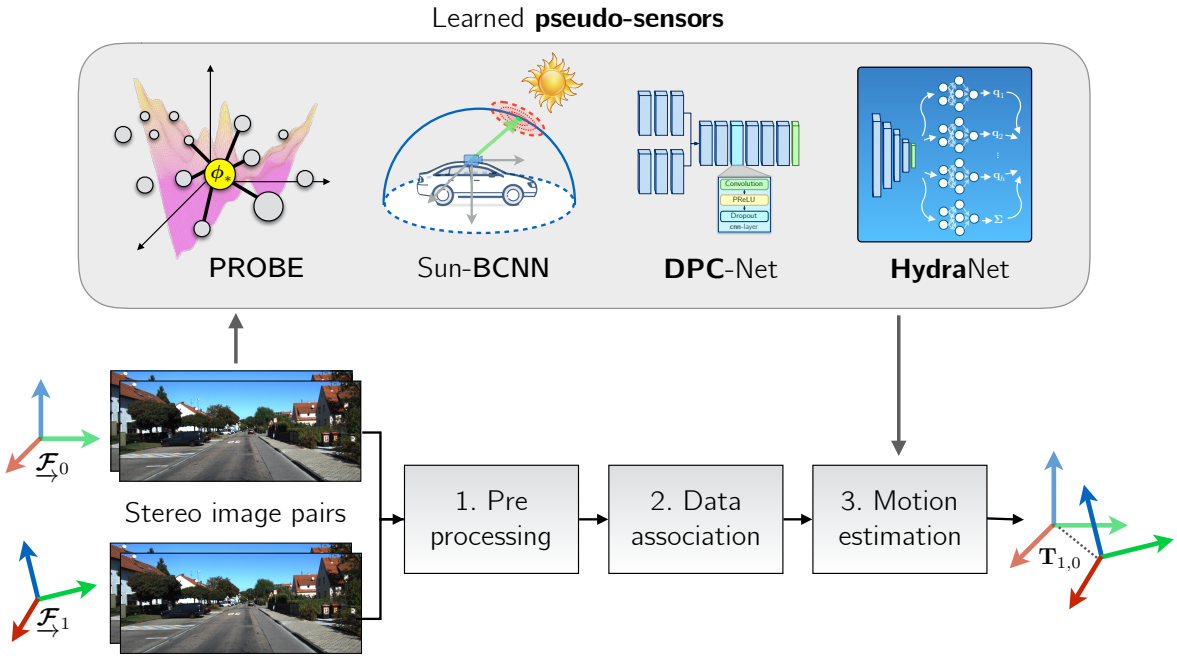


Figure 1.6: This dissertation details four examples of *pseudo-sensors* that improve 'classical' egomotion estimation through data-driven learning.

line for an indirect stereo visual odometry pipeline (Chapter 3) which all four methods build upon. In total, there are two journal papers, and five conference papers associated with our work. Below, we briefly summarize each of the pseudo-sensors and list the publications that are associated with each.

1. **Predictive Robust Estimation (PROBE)**,
   Predictive Robust Estimation (Chapter 4) uses k-NN regression (original PROBE) or Generalized Kernels (Vega-Brown et al., 2014) (PROBE-GK) to train a predictive model for heteroscedastic measurement covariance of stereo reprojection errors to improve the accuracy and consistency of an indirect stereo visual odometry pipeline. It is associated with three publications:

   - Peretroukhin, V., Clement, L., Giamou, M., and Kelly, J. (2015a). PROBE: Predictive robust estimation for visual-inertial navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'15)*, pages 3668–3675, Hamburg, Germany
   - Peretroukhin, V., Clement, L., and Kelly, J. (2015b). Get to the point: Active covariance scaling for feature tracking through motion blur. In *Proceedings of the IEEE International Conference on Robotics and Automation Workshop on Scaling Up*

Active Perception, Seattle, Washington, USA

- Peretroukhin, V., Vega-Brown, W., Roy, N., and Kelly, J. (2016). PROBE-GK: Predictive robust estimation using generalized kernels. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 817–824

2. **Virtual Sun Sensor using a Bayesian Convolutional Neural Network**
Sun-BCNN (**??**) is a technique to infer a probabilistic estimate of the direction of the sun from a single RGB image using a Bayesian Convolutional Neural Networks (BCNN). The method works much like dedicated sun sensors (Lambert et al., 2012), but requires no additional hardware, and can provide mean and covariance estimates that can be readily incorporated into existing visual odometry frameworks. It is associated with three publications listed below. Initial exploratory work was published at ISER 2016, and the BCNN improvement was presented at ICRA 2017. An additional journal paper summarizing the work of the prior two papers, adding data from the Canadian High Arctic and Oxford, and investigating the effect of cloud cover and transfer learning was published in the International Journal of Robotics' Research, Special Issue on Experimental Robotics at the end of 2017.

   - Clement, L., Peretroukhin, V., and Kelly, J. (2017). Improving the accuracy of stereo visual odometry using visual illumination estimation. In Kulic, D., Nakamura, Y., Khatib, O., and Venture, G., editors, *2016 International Symposium on Experimental Robotics*, volume 1 of *Springer Proceedings in Advanced Robotics*, pages 409–419. Springer International Publishing, Berlin Heidelberg. Invited to Journal Special Issue

   - Peretroukhin, V., Clement, L., and Kelly, J. (2017). Reducing drift in visual odometry by inferring sun direction using a bayesian convolutional neural network. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'17)*, pages 2035–2042, Singapore

   - Peretroukhin, V., Clement, L., and Kelly, J. (2018). Inferring sun direction to improve visual odometry: A deep learning approach. *International Journal of Robotics Research*

3. **Deep Pose Corrections (DPC-Net)**
Deep Pose Correction (**??**) is an approach to improving egomotion estimates through pose corrections learned through deep regression. DPC takes as its starting point an efficient, classical localization algorithm that computes high-rate pose estimates. To it,

it adds a Deep Pose Correction Network (DPC-Net) that learns low-rate, 'small' *corrections* from training data that are then fused with the original estimates. DPC-Net does not require any modification to an existing localization pipeline, and can learn to correct multi-faceted errors from estimator bias, sensor mis-calibration or environmental effects. It is associated with a journal publication:

- Peretroukhin, V. and Kelly, J. (2018). DPC-Net: Deep pose correction for visual localization. *IEEE Robotics and Automation Letters*.

4. **Estimating Rotation through Deep Probabilistic Inference with HydraNet**

   Finally, HydraNet (**??**) is a multi-headed network structure that can regress probabilistic estimates of rotation (elements of the matrix Lie group, $\mathrm{SO}(3)$ accounting for both aleatoric and epistemic uncertainty. This uncertainty can then be used to fuse the output of HydraNet with the output of classical egomotion pipelines in a probabilistic factor graph formulation. It is associated with one publication:

   - Peretroukhin, V., Wagstaff, B., and Kelly, J. (2019). Deep probabilistic regression of elements of so(3) using quaternion averaging and uncertainty injection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'19) Workshop on Uncertainty and Robustness in Deep Visual Learning*, pages 83–86, Long Beach, California, USA.

Chapter 2

Mathematical Foundations

> By relieving the brain of all unnecessary work, a good notation sets it free to concentrate on more advanced problems, and, in effect, increases the mental power of the race.
>
> — Alfred North Whitehead

2.1 Coordinate Frames

Before we can present the main contributions of this thesis, it will be useful to first outline the notation and mathematical foundations that underly the work. Throughout this thesis, we largely follow the notation of Barfoot (2017) when dealing with three-dimensional rigid-body kinematics.

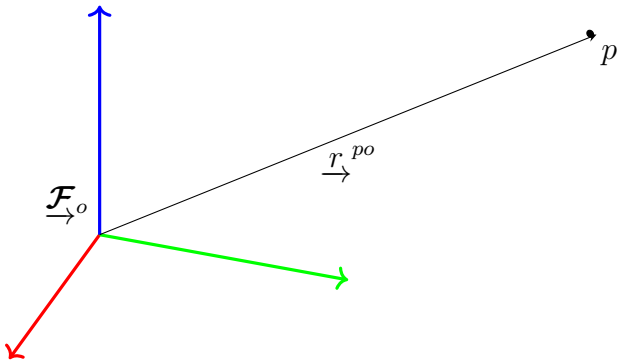


Figure 2.1: A position vector expressed in a coordinate frame.

We refer to a three-dimensional position vector, $\underrightarrow{r}^{po}$, as one that originates at the origin of a coordinate reference frame, $\underrightarrow{\mathcal{F}}_o$, and terminates at the point p. This geometric quantity has

the numerical coordinates $\mathbf{r}_o^{po}$ when expressed in $\underrightarrow{\mathcal{F}}_o$. Often, we will refer to two reference frames such as a world or *inertial* frame, $\underrightarrow{\mathcal{F}}_i$, and a vehicle frame, $\underrightarrow{\mathcal{F}}_v$. Rotation matrices or rigid-body transformations that convert coordinates from $\underrightarrow{\mathcal{F}}_i$ to $\underrightarrow{\mathcal{F}}_v$ will be represented as $\mathbf{T}_{vi}$, and $\mathbf{C}_{vi}$[1], respectively.

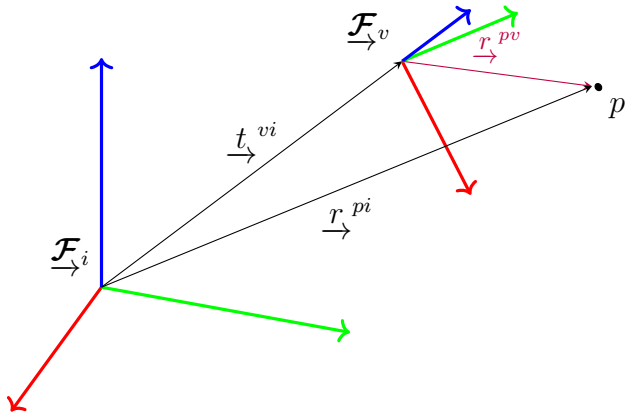


Figure 2.2: Two common references frames used throughout this thesis.

2.2 Rotations

The rotation matrix $\mathbf{C}$ is a member of the matrix Lie group $\mathrm{SO}(3)$ (the Special Orthogonal group) and can be defined as a matrix as follows:

$$\mathrm{SO}(3) = \{\mathbf{C} \in \mathbb{R}^{3\times 3} | \ \mathbf{C}^T\mathbf{C} = \mathbf{1}, \det \mathbf{C} = 1\}. \tag{2.1}$$

Active vs. Passive

An active (or *alibi*) rotation changes the coordinates of a position directly while implicitly assuming that the reference frame is fixed. A passive (or *alias*) rotation rotates the reference frame. Following Barfoot (2017), all rotation matrices in this thesis are passive unless otherwise noted.

Exponential and Logarithmic Maps

Since rotations form a matrix Lie group (we refer the reader to Solà et al. (2018) and Barfoot (2017) for a thorough treatment of Lie groups for state estimation), we can define a surjective

[1]We use $\mathbf{C}$ and not $\mathbf{R}$ for rotation matrices to avoid confusion with common notation for measurement model covariance.

exponential map[2] from three axis-angle parameters, $\boldsymbol{\phi} = \phi\mathbf{a}, \quad \phi \in \mathbb{R}, \mathbf{a} \in S^2$, to a rotation matrix, $\mathbf{C}$:

$$\mathbf{C} = \mathrm{Exp}\left(\boldsymbol{\phi}\right) = \exp\left(\boldsymbol{\phi}^{\wedge}\right) = \sum_{n=0}^{\infty} \frac{1}{n!}(\boldsymbol{\phi}^{\wedge})^n \tag{2.2}$$

$$= \cos\phi\mathbf{1} + (1 - \cos\phi)\mathbf{a}\mathbf{a}^T + \sin\phi\mathbf{a}^{\wedge}, \tag{2.3}$$

where the wedge operator $(\cdot)^{\wedge}$[3] is defined as

$$\mathbf{a}^{\wedge} = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}^{\wedge} = \begin{bmatrix} 0 & -a_2 & a_1 \\ a_2 & 0 & -a_0 \\ -a_1 & a_0 & 0 \end{bmatrix}. \tag{2.4}$$

Equation (2.3) is known as the Euler-Rodriguez formula and it can also be derived geometrically, starting from Euler's theorem that any rotation can be expressed as an axis of rotation and an angle of rotation about that axis. Although the map in Equation (2.2) is surjective, we can define an inverse map if we restrict its domain to $0 \leq \phi < \pi$:

$$\boldsymbol{\phi} = \mathrm{Log}\left(\mathbf{C}\right) = \log\left(\mathbf{C}\right)^{\vee} = \frac{\phi(\mathbf{C} - \mathbf{C}^T)^{\vee}}{2\sin\phi}, \tag{2.5}$$

where $\phi = \arccos\frac{\mathrm{tr}(\mathbf{C})-1}{2}$ and the *vee* operator, $(\cdot)^{\vee} : \mathbb{R}^{3\times3} \to \mathbb{R}^3$, is defined as the unique inverse of the wedge operator $(\cdot)^{\wedge}$. Note Equation (2.5) is undefined at both $\phi = 0$ and at $\phi = \pi$. In the former case, we can use a small-angle approximation and define

$$\mathrm{Log}\left(\mathbf{C}\right) \approx (\mathbf{C} - \mathbf{1})^{\vee} \text{ when } \phi \approx 0. \tag{2.6}$$

The latter case, (when $\phi = \pi$), defines the *cut locus* of the space where $\mathrm{Exp}\left(\cdot\right)$ is not a covering map and both $+\boldsymbol{\phi}$ and $-\boldsymbol{\phi}$ map to the same $\mathbf{C}$. This *cut locus* is related to the idea that any three parameterization of $\mathrm{SO}(3)$ will have singularities associated with it.

2.2.1 Unit Quaternions

Another way (and historically, the original way) to represent a general rotation is to use a unit quaternion, $\mathbf{q}$. A unit quaternion has four parameters, a scalar q_ω and a three-dimensional vector component, $\mathbf{q}_v$:

[2] We follow Solà et al. (2018) and also define *capitalized* map for notational clarity.

[3] This operator is often expressed as $(\cdot)^{\times}$ and is known as the skew-symmetric operator.

$$\mathbf{q} = \begin{bmatrix} q_\omega \\ \mathbf{q}_v \end{bmatrix} \in S^3, \quad (\|\mathbf{q}\| = 1). \tag{2.7}$$

Unit quaternions also form a Lie group (Solà et al., 2018) and and lie on a three-dimensional unit sphere within $\mathbb{R}^4$. This manifold represents a double cover of $\mathrm{SO}(3)$ (since both $\mathbf{q}$ and $-\mathbf{q}$ represent the same rotation). As with rotation matrices, we can define a surjective map from three parameters to the group itself,

$$\mathbf{q} = \mathrm{Exp}\left(\boldsymbol{\phi}\right) = \begin{bmatrix} \cos\phi/2 \\ \mathbf{a}\sin\phi/2 \end{bmatrix}. \tag{2.8}$$

Similarly, we can also define a logarithmic map,

$$\boldsymbol{\phi} = \mathrm{Log}\left(\mathbf{q}\right) = 2\mathbf{q}_v \frac{\arctan\left(\|\mathbf{q}_v\|, q_\omega\right)}{\|\mathbf{q}_v\|}. \tag{2.9}$$

To avoid issues with the double cover, we replace $\mathbf{q}$ with $-\mathbf{q}$ if q_ω is negative before evaluating Equation (2.9). Also note again that Equation (2.9) is undefined when $\phi = 0$, but, importantly, we do not face any issues when $\phi = \pi$ due to the half angle. As with rotation matrices, we can use small angle approximations to define:

$$\mathrm{Log}\left(\mathbf{q}\right) \approx \frac{\mathbf{q}_v}{q_\omega}\left(1 - \frac{\|\mathbf{q}_v\|^2}{3q_\omega^2}\right) \quad \text{when} \quad \phi \approx 0. \tag{2.10}$$

A fantastic summary of the history of rotation parameterizations, unit quaternions and the story of Hamilton and Rodriguez can be found in Altmann (1989).

2.3 Spatial Transforms

The rigid body transform $\mathbf{T}$ is a also a member of the matrix Lie group, the Special Euclidian group $\mathrm{SE}(3)$ and can be defined as a 4×4 matrix as follows:

$$\mathrm{SE}(3) = \{\mathbf{T} = \begin{bmatrix} \mathbf{C} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4\times 4} | \ \mathbf{C} \in \mathrm{SO}(3), \mathbf{t} \in \mathbb{R}^3\}. \tag{2.11}$$

As a member of a matrix Lie group, it also admits a surjective exponential map,

$$\mathbf{T} = \mathrm{Exp}\left(\boldsymbol{\xi}\right) = \exp\left(\boldsymbol{\xi}^\wedge\right) = \sum_{n=0}^{\infty} \frac{1}{n!}(\boldsymbol{\xi}^\wedge)^n \tag{2.12}$$

where $\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix} \in \mathbb{R}^6$ and the wedge operator is overloaded (following Barfoot (2017)) as follows:

$$\boldsymbol{\xi}^\wedge \triangleq \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix}^\wedge = \begin{bmatrix} \boldsymbol{\phi}^\wedge & \boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{bmatrix}. \tag{2.13}$$

In practice, we can evaluate the exponential map through the Euler-Rodriguez formula (Equation (2.3)) and by computing the left-Jacobian of $\mathrm{SO}(3)$, $\mathbf{J}$,

$$\mathbf{T} = \mathrm{Exp}\left(\begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\phi} \end{bmatrix}\right) = \begin{bmatrix} \mathbf{C}(\boldsymbol{\phi}) & \mathbf{J}(\boldsymbol{\phi})\boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{2.14}$$

where

$$\mathbf{J}(\boldsymbol{\phi}) = \frac{\sin\phi}{\phi}\mathbf{1} + (1 - \frac{\sin\phi}{\phi})\mathbf{a}\mathbf{a}^T + \frac{1-\cos\phi}{\phi}\mathbf{a}^\wedge. \tag{2.15}$$

2.3.1 Applying Transforms

Applying our notation for coordinate frames (and referring back to Section 2.1), a transform, $\mathbf{T}_{vi}$ can be expressed as

$$\mathbf{T}_{vi} = \begin{bmatrix} \mathbf{C}_{vi} & \mathbf{t}_v^{iv} \\ \mathbf{0}^T & 1 \end{bmatrix}. \tag{2.16}$$

This allows us to use the homogenous point representation for $\mathbf{r}_i^{pi}$ and express the following relation:

$$\begin{bmatrix} \mathbf{r}_v^{pi} \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{C}_{vi} & \mathbf{t}_v^{iv} \\ \mathbf{0}^T & 1 \end{bmatrix}}_{\mathbf{T}_{vi}} \begin{bmatrix} \mathbf{r}_i^{pi} \\ 1 \end{bmatrix} \tag{2.17}$$

which is numerically equivalent to

$$\mathbf{r}_v^{pi} = \mathbf{C}_{vi}\mathbf{r}_i^{pi} + \mathbf{t}_v^{iv} \tag{2.18}$$

2.4 Perturbations

It is often useful to consider a small *perturbation* about an operating point (whether that be a rotation or rigid-body transform). By leveraging a core property of Lie groups (that they are locally 'Euclidian'), we can convert difficult non-linear problems into ones that have local linear approximations.

Using rotations as an example, we can perturb an operating point, $\overline{\mathbf{C}} \triangleq \operatorname{Exp}\left(\overline{\boldsymbol{\phi}}\right)$, in three different ways:

$$\mathbf{C} = \begin{cases} \operatorname{Exp}\left(\delta\boldsymbol{\phi}^{\ell}\right)\overline{\mathbf{C}} & \text{left perturbation,} \\ \operatorname{Exp}\left(\overline{\boldsymbol{\phi}} + \delta\boldsymbol{\phi}^{m}\right) & \text{middle perturbation,} \\ \overline{\mathbf{C}}\operatorname{Exp}\left(\delta\boldsymbol{\phi}^{r}\right) & \text{right perturbation.} \end{cases} \tag{2.19}$$

We can relate all the left and middle perturbations through the left Jacobian of $\mathrm{SO}(3)$ with the following useful identity,

$$\operatorname{Exp}\left(\left(\boldsymbol{\phi} + \delta\boldsymbol{\phi}^{m}\right)\right) \approx \operatorname{Exp}\left(\mathbf{J}(\boldsymbol{\phi})\delta\boldsymbol{\phi}^{m}\right)\operatorname{Exp}\left(\boldsymbol{\phi}\right). \tag{2.20}$$

This allows us to write $\delta\boldsymbol{\phi}^{\ell} \approx \mathbf{J}(\boldsymbol{\phi})\delta\boldsymbol{\phi}^{m}$ and elucidates why $\mathbf{J}$ is called the *left* Jacobian.

In this thesis, we will use the left and middle perturbations when appropriate. Using small angle approximations, the Euler-Rodriguez formula (Equation (2.3)) yields $\operatorname{Exp}\left(\delta\boldsymbol{\phi}\right) \approx \mathbf{1} + \delta\boldsymbol{\phi}^{\wedge}$, which allows us to write the useful formula for the left perturbation:

$$\mathbf{C} = (\mathbf{1} + (\delta\boldsymbol{\phi}^{\ell})^{\wedge})\overline{\mathbf{C}}. \tag{2.21}$$

Similarly, we can write analogous expressions for a rigid body transform, $\mathbf{T} \in \mathrm{SE}(3)$, as composed of an operating point $\overline{\mathbf{T}} \triangleq \operatorname{Exp}\left(\overline{\boldsymbol{\xi}}\right)$, and a small perturbation about that operating point:

$$\mathbf{T} = \begin{cases} \operatorname{Exp}\left(\delta\boldsymbol{\xi}^{\ell}\right)\overline{\mathbf{T}} & \text{left perturbation,} \\ \operatorname{Exp}\left(\overline{\boldsymbol{\xi}} + \delta\boldsymbol{\xi}^{m}\right) & \text{middle perturbation,} \\ \overline{\mathbf{T}}\operatorname{Exp}\left(\delta\boldsymbol{\xi}^{r}\right) & \text{right perturbation.} \end{cases} \tag{2.22}$$

Now, we can also note a similar identity for $\mathrm{SE}(3)$,

$$\operatorname{Exp}\left(\left(\boldsymbol{\xi} + \delta\boldsymbol{\xi}^{m}\right)\right) \approx \operatorname{Exp}\left(\left(\boldsymbol{\mathcal{J}}(\boldsymbol{\xi})\delta\boldsymbol{\xi}^{m}\right)\right)\operatorname{Exp}\left(\boldsymbol{\xi}\right), \tag{2.23}$$

where $\boldsymbol{\mathcal{J}}$ is the left Jacobian of $\mathrm{SE}(3)$ and defined as

$$\boldsymbol{\mathcal{J}}(\boldsymbol{\xi}) \triangleq \begin{bmatrix} \mathbf{J}(\boldsymbol{\phi}) & \mathbf{Q}(\boldsymbol{\xi}) \\ \mathbf{0} & \mathbf{J}(\boldsymbol{\phi}) \end{bmatrix}, \tag{2.24}$$

where $\mathbf{Q}(\boldsymbol{\xi})$ can be evaluated analytically (see Barfoot (2017)). This again allows us to write $\delta\boldsymbol{\xi}^{\ell} \approx \boldsymbol{\mathcal{J}}(\boldsymbol{\xi})\delta\boldsymbol{\xi}^{m}$ and form a similar expression,

$$\mathbf{T} = (\mathbf{1} + (\delta\boldsymbol{\xi}^{\ell})^{\wedge})\overline{\mathbf{T}}. \tag{2.25}$$

To derive locally linear systems from sets of rigid-body transforms, or 'poses', we can apply Equation (2.25). To update an operating point, we solve for $\delta\boldsymbol{\xi}^{\ell}$ and then use the constraint-sensitive update $\mathbf{T} \leftarrow \mathrm{Exp}\left(\delta\boldsymbol{\xi}^{\ell}\right)\overline{\mathbf{T}}$.

2.5 Uncertainty

We can also use perturbation theory to implicitly define uncertainty on constrained manifolds (see Barfoot and Furgale (2014a) for a thorough discussion).

Given a concentrated normal density, $\delta\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{6\times 6})$, we can *inject* this unconstrained density onto the Lie group through left perturbations about some mean:

$$\mathbf{T} = \mathrm{Exp}\left(\delta\boldsymbol{\xi}\right)\overline{\mathbf{T}} \tag{2.26}$$

This allows us to keep track of a random variable, $\mathbf{T}$, by keeping its mean in group form, $\overline{\mathbf{T}}$, while its second statistical moment is stored as a standard 6×6 covariance matrix, $\boldsymbol{\Sigma}$.

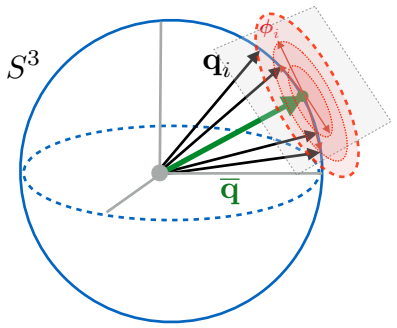


Figure 2.3: We can define uncertainty in the left tangent space of a mean element of a Lie group (here illustrated for unit quaternions).

We can define an analogous density for rotation matrices given normal densities over rotation perturbations $\delta\boldsymbol{\phi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{3\times 3})$,

$$\mathbf{C} = \mathrm{Exp}\left(\delta\boldsymbol{\phi}\right)\overline{\mathbf{C}}, \tag{2.27}$$

and also, for unit quaternions,

$$\mathbf{q} = \mathrm{Exp}\left(\delta\boldsymbol{\phi}\right) \otimes \overline{\mathbf{q}} \tag{2.28}$$

where $\otimes$ refers to the standard quaternion product operator Sola (2017).

Chapter 3

Classical Visual Odometry

> Eventually, my eyes were opened, and I really understood nature.
>
> — Claude Monet

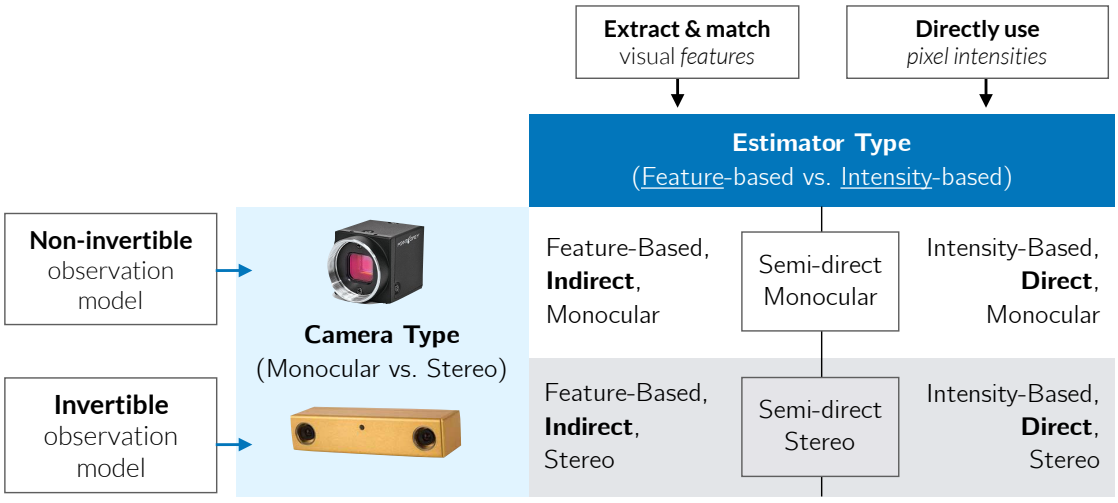


Figure 3.1: A taxonomy of different types of visual odometry.

Visual odometry (VO) has a rich history in mobile robotics and computer vision. As this dissertation largely deals with the improvement of a baseline visual odometry pipeline, we first outline the components of what we have chosen to be a canonical VO system. For a seminal tutorial on visual odometry and its more general cousin, visual SLAM, we refer the reader to two seminal papers: Scaramuzza and Fraundorfer (2011a) and Cadena et al. (2016).

3.1 A taxonomy of VO

VO can be largely divided along two dimensions (c.f. Figure 3.1): the type of camera (monocular vs. stereo) and the type of data association (indirect, or feature-based vs. direct, or pixel intensity-based).

Monocular vs. Stereo: The first distinction is based on the type of camera used by the VO pipeline. Monocular VO methods use a single camera to infer motion and can use a single compact, low-power vision sensor. They do not require any extrinsic calibration but must rely on known visual cues or external information (e.g., wheel odometry, inertial measurements) to provide metric egomotion estimates. Conversely, stereo VO methods use a stereo camera to triangulate objects with metric scale. This allows stereo VO to provide metricaly-acurate egomotion estimates. However, stereo methods rely on accurate extrinsic calibration, and their ability to resolve depth is limited by the baseline distance between the stereo pair.

Direct vs. Indirect: The second distinction is based on the type of data association used to match sequential images. Direct methods make the assumption of brightness constancy, and attempt to *directly* maximize the similarity of pixel intensities. Indirect methods, however, rely on image features detectors to extract a set of salient landmarks, and then match these landmarks across images (typically through some sort of invariant descriptor).

3.2 A classical VO pipeline

In this thesis, we apply our learned pseudo-sensors to a baseline stereo, indirect visual odometry pipeline largely based on the work of Furgale (2011). We choose this baseline system for its computational efficiency and robustness. We briefly summarize the main components of the dpipeline here.

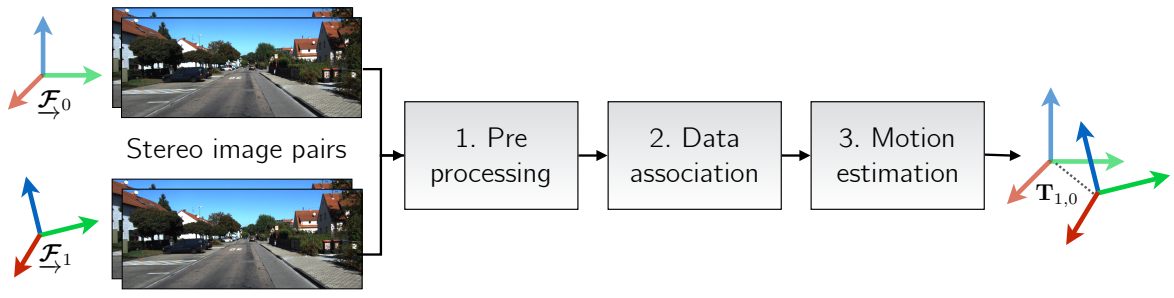
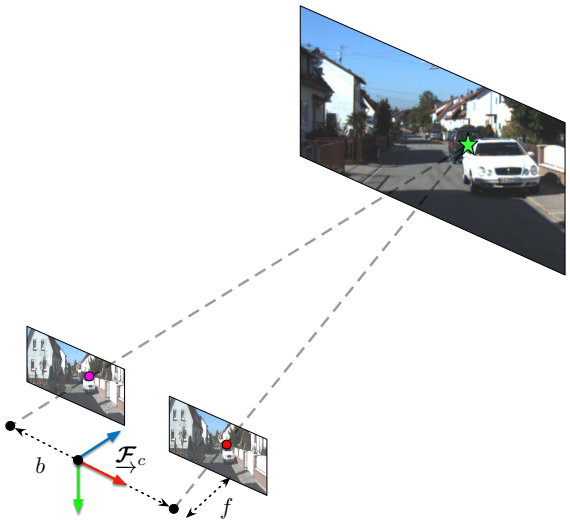
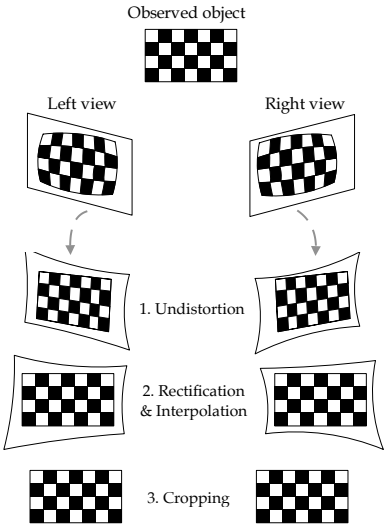


Figure 3.2: A 'classical' stereo visual odometry pipeline consists of several distinct components that have interpretable inputs and outputs.

3.2.1 Preprocessing



(a) Ideal stereo camera.



(b) Rectification and undistortion process. Figure adapted from Florez (2010).

Figure 3.3: Preprocessing components.

During preprocessing, we use a lens model (assumed to be known apriori) to undistort each stereo image. Further, using the camera extrinsic parameters (also assumed to be known), we *rectify* the stereo pair such that the images can be assumed to come from two cameras whose principal axes are parallel (Figure 3.3). Finally, we also assume that the stereo camera intrinsics are known a priori or compute them through a calibration process (Furgale et al., 2013).

3.2.2 Data Association

Feature Extraction and Matching

In this thesis, we focus on indirect stereo visual odometry for its computational efficiency. Although a number of different types of indirect feature extraction and matching methods can be used towards this end, we choose to use the `viso2` (Geiger et al., 2011b) image feature extraction and matching algorithm. In `viso2`, features are extracted using blob and corner masks with non-minimum and non-maximum suppression. Unlike other features detectors that do not assume a particular camera motion, `viso2` assumes a smooth camera trajectory that permits fast matching through a simple sum-of-absolute-difference error metric of 11×11 windows of Sobel filter responses. Finally, features are matched across a stereo-pair and

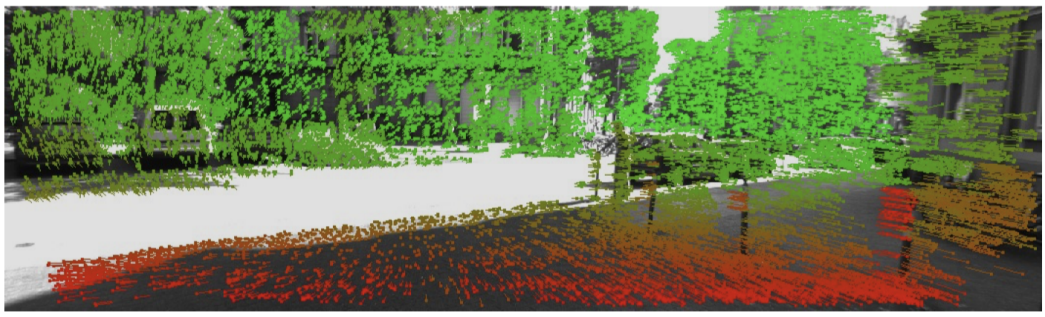


Figure 3.4: Feature tracking using `libviso2`, taken from Geiger et al. (2011a). Colours correspond to depth.

forward in time, to ensure that a single feature exists across two consecutive stereo camera poses.

Each extract feature corresponds to a point in space, expressed in homogeneous coordinates in the camera frame as $\boldsymbol{p}_{i,t} := \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \end{bmatrix}^T \in \mathbb{P}^3$. Given our intrinsics and extrinsic calibration parameters, our idealized stereo-camera model, $\mathbf{f}$, projects a landmark expressed in homogeneous coordinates into image space, so that $\mathbf{y}_{i,t}$, the stereo pixel coordinates of landmark i in the first camera pose at time t, is given by

$$\mathbf{y}_{i,t} = \begin{bmatrix} u_l \\ v_l \\ u_r \\ v_r \end{bmatrix} = \mathbf{f}(\boldsymbol{p}_{i,t}) = \mathbf{M}\frac{1}{p_3}\boldsymbol{p}_{i,t}, \tag{3.1}$$

where

$$\mathbf{M} = \begin{bmatrix} f & 0 & c_u & f\frac{b}{2} \\ 0 & f & c_v & 0 \\ f & 0 & c_u & -f\frac{b}{2} \\ 0 & f & c_v & 0 \end{bmatrix}. \tag{3.2}$$

Here, $\{c_u, c_v\}$, $\{f_u, f_v\}$, and b are the principal points, focal lengths and baseline of the stereo camera respectively. Note that in this formulation, the stereo camera frame is centered between the two individual lenses.

Outlier Rejection

To filter out any residual outlier matches, we use a three-point random sample consensus algorithm (RANSAC, Fischler and Bolles (1981)) based on an analytic solution to the six degree-

of-freedom motion (Umeyama, 1991).

3.2.3 Maximum Likelihood Motion Solution

We will define $\mathbf{T}_t \in \mathrm{SE}(3)$, as the rigid transform between two subsequent stereo camera poses, $\underrightarrow{\mathcal{F}}_{c_0}$ and $\underrightarrow{\mathcal{F}}_{c_1}$

$$\mathbf{T}_t = \mathbf{T}_{c_1 w} \mathbf{T}_{c_0 w}^{-1}, \tag{3.3}$$

where $\underrightarrow{\mathcal{F}}_{w}$ is a privileged world frame. After data association, we assume we have a set of N_t matches, $\{\mathbf{y}_{i,c_0}, \mathbf{y}_{i,c_1}\}_{i=1}^{N_t}$, between visual landmarks in the subsequent camera frames. For each match, we define an error function, $\mathbf{e}_i(\mathbf{T}_t)$, that relates the rigid transform to these stereo feature matches. Throughout this dissertation, we assume that these errors are corrupted by zero-mean independent Gaussian noise with the (potentially heteroscedastic) covariance, $\mathbf{\Sigma}_{i,t}$;

$$\mathbf{e}_i(\mathbf{T}_t) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Sigma}_{i,t}\right). \tag{3.4}$$

Under this noise model, the maximum likelihood transform, $\mathbf{T}_t^*$, is given by

$$\mathbf{T}_t^* = \underset{\mathbf{T}\in\mathrm{SE}(3)}{\operatorname{argmax}} \prod_{i=1}^{N_t} p(\mathbf{e}_i(\mathbf{T}_t)) = \underset{\mathbf{T}\in\mathrm{SE}(3)}{\operatorname{argmin}} \sum_{i=1}^{N_t} \mathbf{e}_i(\mathbf{T}_t)^T \mathbf{\Sigma}_{i,t}^{-1} \mathbf{e}_i(\mathbf{T}_t). \tag{3.5}$$

We will define the error function in two different ways.

Point Cloud Error

First, we can follow classical approach (Maimone et al., 2007) and define $\mathbf{e}_i(\mathbf{T}_t)$ based on a three-dimensional point cloud error. To do this, we invert our stereo camera model to triangulate pairs of points in each frame, $\boldsymbol{p}_{i,c_0} = \mathbf{f}^{-1}(\mathbf{y}_{i,c_0})$ and $\boldsymbol{p}_{i,c_1} = \mathbf{f}^{-1}(\mathbf{y}_{i,c_1})$,

$$\mathbf{e}_i(\mathbf{T}_t) = \mathbf{D}(\boldsymbol{p}_{i,c_1} - \mathbf{T}_t \boldsymbol{p}_{i,c_0}), \tag{3.6}$$

where $\mathbf{D} = \begin{bmatrix} \mathbf{1}_{3\times 3} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{3\times 4}$ converts homogenous coordinates into Euclidian coordinates.

We follow Maimone et al. (2007) and assume each stereo projection is corrupted by additive Gaussian noise,

$$\mathbf{y}_{i,c} \sim \mathcal{N}\left(\bar{\mathbf{y}}_{i,c}, \mathbf{R}_{i,c}\right), \tag{3.7}$$

then we can compute a density on the error function itself through first order noise prop-

agation as

$$\mathbf{e}_i(\mathbf{T}_t) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Sigma}_{i,t}\right), \tag{3.8}$$

where

$$\mathbf{\Sigma}_{i,t} = \mathbf{D}\mathbf{G}_{i,c_1}\mathbf{R}_{i,c_1}\mathbf{G}_{i,c_1}^T\mathbf{D}^T + \mathbf{D}\mathbf{T}_t\mathbf{G}_{i,c_0}\mathbf{R}_{i,c_0}\mathbf{G}_{i,c_0}^T\mathbf{T}_t^T\mathbf{D}^T \tag{3.9}$$

with $\mathbf{G}_{i,c} = \left.\dfrac{\partial \mathbf{f}^{-1}}{\partial \mathbf{y}}\right|_{\mathbf{y}_{i,c}}$.

Reprojection Error

Alternatively, we can represent reprojection errors in the second frame directly as

$$\mathbf{e}_i(\mathbf{T}_t) = \mathbf{y}_{i,c_1} - \mathbf{f}(\mathbf{T}_t\mathbf{f}^{-1}(\mathbf{y}_{i,c_0})), \tag{3.10}$$

and model reprojection errors directly as

$$\mathbf{e}_i(\mathbf{T}_t) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Sigma}_{i,t}\right) = \mathcal{N}\left(\mathbf{0}, \mathbf{R}_{i,t}\right), \tag{3.11}$$

where we abuse notation (slightly) and replace the index for the camera frames c_0 or c_1 with t to indicate that this covariance refers to the reprojection error that involves both sets of features.

Solution via Gauss-Newton Optimization

In either case, we have now defined a weighted nonlinear least squares problem which can be solved iteratively using standard techniques. For our purposes, we opt to use Gauss-Newton optimization and follow Barfoot (2017) to optimize constrained poses.

Namely, at a given iteration n, we linearize the error function $\mathbf{e}_i(\mathbf{T}_t)$, about an operating point $\mathbf{T}_t^{(n)} \in \mathrm{SE}(3)$, which results in a quadratic approximation to Equation (A.3). To linearize, we consider the left perturbations $\delta\boldsymbol{\xi} \in \mathbb{R}^6$ represented in exponential coordinates:

$$\mathbf{T}_t = \mathrm{Exp}\left(\delta\boldsymbol{\xi}\right)\mathbf{T}_t^{(n)} \approx (\mathbf{1} + \delta\boldsymbol{\xi}^{\wedge})\mathbf{T}_t^{(n)}. \tag{3.12}$$

This allows us to transform Equation (A.3) into a linear least squares objective in $\delta\boldsymbol{\xi}$:

$$\mathcal{L}(\delta\boldsymbol{\xi}) = \frac{1}{2}\sum_{i=1}^{N_t}\left(\mathbf{e}_i - \mathbf{J}_i\delta\boldsymbol{\xi}\right)^T\mathbf{\Sigma}_i^{-1}\left(\mathbf{e}_i - \mathbf{J}_i\delta\boldsymbol{\xi}\right) \tag{3.13}$$

where $\mathbf{J}_i = \left.\frac{\partial \mathbf{e}_i}{\partial \delta \boldsymbol{\xi}}\right|_{\mathbf{T}_t^{(n)}}$, $\mathbf{e}_i = \mathbf{e}_i(\mathbf{T}_t^{(n)})$, and $\boldsymbol{\Sigma}_i = \boldsymbol{\Sigma}_{i,t}(\mathbf{T}_t^{(n)})$. The minimum to this objective can be solved for analytically by solving the normal equations. This results in the optimal parameters,

$$\delta \boldsymbol{\xi}^\star = \left(\sum_{i=1}^{N_t} \mathbf{J}_i^T \, \boldsymbol{\Sigma}_i^{-1} \mathbf{J}_i \right)^{-1} \sum_{i=1}^{N_t} \mathbf{J}_i^T \, \boldsymbol{\Sigma}_i^{-1} \mathbf{e}_i. \tag{3.14}$$

We then update the operating point and proceed to the next iteration,

$$\mathbf{T}^{(n+1)} = \operatorname{Exp}\left(\delta \boldsymbol{\xi}^\star\right) \mathbf{T}^{(n)}. \tag{3.15}$$

There are many reasonable choices for both the initial transform $\mathbf{T}^{(0)}$ and for the conditions under which we terminate iteration. We initialize the estimated transform to identity, and iteratively perform the update given by eq. (3.15) until we see a relative change in the squared error of less than one percent after an update.

3.3 Robust Estimation

Since eq. (3.13) assigns cost values that grow quadratically with measurement error, it is very sensitive to outlier measurements. A common solution to this problem is to replace the L_2 cost function with one that is less sensitive to large measurement errors (MacTavish and Barfoot, 2015). These robust cost functions are collectively known as M-estimators, and many variants exist. Each uses a re-weighting function, $\rho(\cdot)$,

$$\mathbf{T}^* = \underset{\mathbf{T} \in \mathrm{SE}(3)}{\operatorname{argmin}} \sum_{i=1}^{N_t} \rho\left({\mathbf{e}_i}^T \, \boldsymbol{\Sigma}_{i,t}^{-1} \mathbf{e}_i \right) = \underset{\mathbf{T} \in \mathrm{SE}(3)}{\operatorname{argmin}} \sum_{i=1}^{N_t} \rho(\epsilon_i), \tag{3.16}$$

where. given a parameter c, some common examples include:

$$\rho(\epsilon) = \begin{cases} \frac{c^2}{2} \log\left(1 + \frac{\epsilon^2}{c^2}\right) & \text{Cauchy,} \\ \frac{1}{2} \frac{\epsilon^2}{c^2 + \epsilon^2} & \text{Geman-McClure (Geman et al., 1992),} \\ \begin{cases} \frac{\epsilon^2}{2} & \text{if } \|\epsilon\| < c \\ c \|\epsilon\| - \frac{c^2}{2} & \text{if } \|\epsilon\| \geq c \end{cases} & \text{Huber (Huber, 1964).} \end{cases} \tag{3.17}$$

3.4 Outstanding Issues

There are several outstanding limitations of classical visual odometry pipelines that we can address with learned pseudo-sensors.

Table 3.1: **Data efficiency vs. computational efficiency**

Synopsis	**Addressed by**
Classical VO pipelines face a difficult-to-optimize trade-off between using all of the information contained within image and while still remaining computationally tractable.	PROBE, DPC-Net, Sun-BCNN, HydraNet

Table 3.2: **Systematic bias**

Synopsis	**Addressed by**
Stereo visual odometry can incur systematic bias through poor extrinsic or intrinsic calibration, stereo triangulation errors, poor feature *spread* (i.e., concentration of features on one side of an image), and poor data association due self-similar textures.	DPC-Net

Table 3.3: **Homoscedastic uncertainty**

Synopsis	**Addressed by**
Stationary, homoscedastic noise in observation models can often reduce the consistency and accuracy of state estimates. This is especially true for complex, inferred measurement models. In visual data, inferred visual observations can be degraded not only due to sensor imperfections (e.g. poor intrinsic calibration, digitization effects, motion blur), but also as a result of the observed environment (e.g. self-similar scenes, specular surfaces, textureless environments).	PROBE, Sun-BCNN, HydraNet

Chapter 4

Predictive Robust Estimation

> Information is the resolution of uncertainty.
>
> Claude Shannon

4.1 Introduction

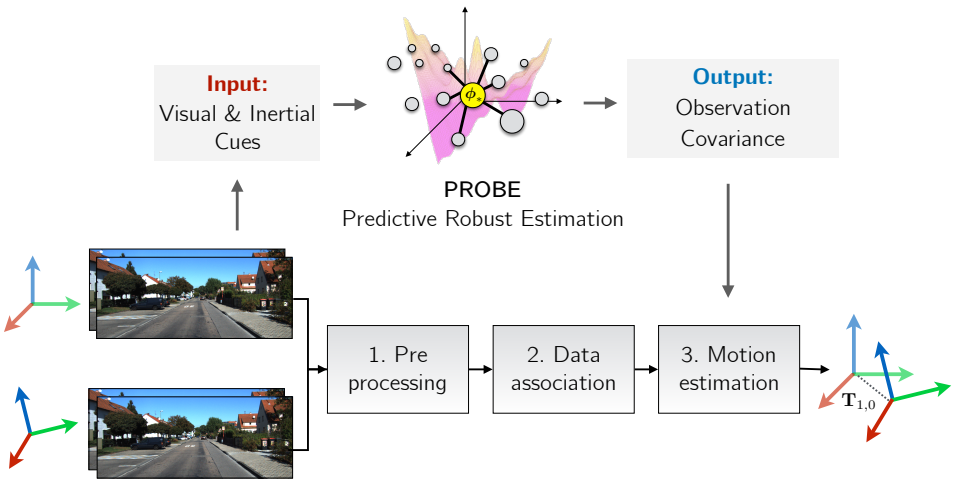


Figure 4.1: PROBE builds a predictive noise model for stereo visual odometry.

The first pseudo-sensor we present is a technique we call PRedictive ROBust Estimation, or PROBE. This approach uses non-parametric learning to build a model for anisotropic observation covariances for a stereo visual odometry pipeline. Namely, we apply the method of Generalized Kernels to a Bayesian treatment of covariance estimation. We show that by assuming a particular covariance prior over re-projection errors, we can then naturally derive

a robust least squares objective that resembles the widely-used Cauchy loss. The parameters of this robust loss are predicted (hence *predictive* robust estimation) for each error term as a function of a prediction space that we define.

PROBE was initially published as a simpler non-Bayesian technique that learned isotropic covariances through a k-nearest-neighbours approach (see Appendix A for more details). The following two publications summarize this initial technique:

1. Peretroukhin, V., Clement, L., Giamou, M., and Kelly, J. (2015a). PROBE: Predictive robust estimation for visual-inertial navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'15)*, pages 3668–3675, Hamburg, Germany

2. Peretroukhin, V., Clement, L., and Kelly, J. (2015b). Get to the point: Active covariance scaling for feature tracking through motion blur. In *Proceedings of the IEEE International Conference on Robotics and Automation Workshop on Scaling Up Active Perception*, Seattle, Washington, USA

We significantly extended this technique to full anisotropic covariances and generalized kernels in the following publication:

1. Peretroukhin, V., Vega-Brown, W., Roy, N., and Kelly, J. (2016). PROBE-GK: Predictive robust estimation using generalized kernels. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 817–824 .

We will present this latter technique in this chapter.

4.2 Motivation

Robot navigation relies on an accurate quantification of sensor noise or uncertainty in order to produce reliable state estimates. In practice, this uncertainty is often fixed for a given sensor and experiment, whether by automatic calibration or by manual tuning. Although a fixed measure of uncertainty may be reasonable in certain static environments, dynamic scenes frequently exhibit many effects that corrupt a portion of the available observations. For visual sensors, these effects include, for example, self-similar textures, variations in lighting, moving objects, and motion blur. Further, there may be useful information available in these observations that would normally be rejected by a fixed-threshold outlier rejection scheme. Ideally, we would like to retain some of these observations in our estimator, while still placing more trust in observations that do not suffer from such effects.

4.3 Related Work

There is a large and growing body of work on the problem of deriving accurate, consistent state estimates from visual data. Although our approach to noise modelling is applicable in other domains, for simplicity we focus our attention on the problem of inferring egomotion from features extracted from sequential pairs of stereo images; see Sünderhauf and Protzel (2007) for a survey of techniques. The spectrum of alternative approaches to visual state estimation include monocular techniques, which may be feature-based (Scaramuzza and Fraundorfer, 2011b), direct (Irani and Anandan, 2000), or semi-direct (Forster et al., 2014b).

Apart from simply rejecting outliers, a number of recent approaches attempt to select the optimal set of features to produce an accurate localization estimate from tracked visual features. For example, Tsotsos et al. (2015b) amend Random Sample Consensus (RANSAC) with statistical hypothesis testing to ensure that tracked visual features have normally distributed residuals before including them in the estimator. Unlike our predictive approach, their technique relies on the availability of feature tracks, and requires scene overlap to work continuously. In a different approach, Zhang and Vela (2015) choose an optimally observable feature subset for a monocular SLAM pipeline by selecting features with the highest *informativeness* - a measure calculated based on the observability of the SLAM subsystem. Observability, however, is governed by the 3D location of the features, and therefore cannot predict systematic feature degradation due to environmental or sensor-based effects.

4.4 Predictive Robust Estimation for VO

We present a principled, data-driven way to build a noise model for visual odometry. We combine our previous work on predictive robust estimation with isotropic covariances (Appendix A) with work on covariance estimation (Vega-Brown and Roy, 2013) to formulate a predictive robust estimator for a stereo visual odometry pipeline. We frame the traditional non-linear least squares optimization problem as a problem of maximum likelihood estimation with a Gaussian noise model, and infer a distribution over the covariance matrix of the Gaussian noise from a predictive model learned from training data. This results in a Student's t distribution over the noise, and naturally yields a robust nonlinear least-squares optimization problem. In this way, we can predict, in a principled manner, how informative each visual feature is with respect to the final state estimate, which allows our approach to intelligently weight observations to produce more accurate odometry estimates.

4.4.1 Bayesian Noise Model for Visual Odometry

We adopt the motion solution for visual odometry based on reprojection errors presented in Section 3.2.3. In brief, this technique assumes independent Gaussian errors on stereo reprojections of a landmark from one frame, $\underrightarrow{\mathcal{F}}_{c_0}$ into a subsequent frame, $\underrightarrow{\mathcal{F}}_{c_1}$:

$$\mathbf{e}_i(\mathbf{T}_t) = \mathbf{e}_{i,t} = \mathbf{y}_{i,c_1} - \mathbf{f}(\mathbf{T}_t \mathbf{f}^{-1}(\mathbf{y}_{i,c_0})) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{i,t}). \tag{4.1}$$

Maximizing the likelihood of these errors is then equivalent to solving the following weighted non-linear least squares objective for $\mathbf{T}_t \in \mathrm{SE}(3)$ the rigid-body transform that transforms points in $\underrightarrow{\mathcal{F}}_{c_0}$ to those in $\underrightarrow{\mathcal{F}}_{c_1}$:

$$\mathbf{T}_t^* = \underset{\mathbf{T}\in\mathrm{SE}(3)}{\operatorname{argmax}} \prod_{i=1}^{N_t} p(\mathbf{e}_{i,t}; \mathbf{T}_t, \mathbf{R}_{i,t}) \tag{4.2}$$

$$= \underset{\mathbf{T}\in\mathrm{SE}(3)}{\operatorname{argmin}} \sum_{i=1}^{N_t} {\mathbf{e}_{i,t}}^T \mathbf{R}_{i,t}^{-1} \mathbf{e}_{i,t}. \tag{4.3}$$

With PROBE, instead of treating $\mathbf{R}_{i,t}$ as fixed, we build a model for it as a function of some useful *predictor*, $\boldsymbol{\phi}_{i,t}$.

$$\mathbf{R}_{i,t} = \mathbf{R}(\boldsymbol{\phi}_{i,t}). \tag{4.4}$$

Each predictor can be computed based on the stereo track ($\{\mathbf{y}_{i,c_0}, \mathbf{y}_{i,c_1}\}$) and additional visual[1] and inertial cues, allowing us to model effects like motion blur and self-similar textures. Further, instead of treating the covariance as a point function $\mathbf{R}(\boldsymbol{\phi}_{i,t})$, we instead build a non-parametric model of covariance *density* based on a training dataset, $\mathcal{D}$,

$$p(\mathbf{R}_{i,t}) = p(\mathbf{R}|\mathcal{D}, \boldsymbol{\phi}_{i,t}). \tag{4.5}$$

We will seek the transform that then maximizes the posterior predictive distribution of the errors, given this posterior:

$$\mathbf{T}_t^* = \underset{\mathbf{T}\in\mathrm{SE}(3)}{\operatorname{argmax}} \prod_{i=1}^{N_t} \int p(\mathbf{e}_{i,t}; \mathbf{T}_t|\mathbf{R}_{i,t}) p(\mathbf{R}|\mathcal{D}, \boldsymbol{\phi}_{i,t}) d\mathbf{R} \tag{4.6}$$

Although at first this may seem unwieldy, we present an efficient method for computing the posterior and show that a particular formulation allows it to be marginalized out analytically to arrive at a simple posterior predictive distribution with a straight-forward objective for

[1] Including potentially data from all four images in the pair of stereo images.

achieving a maximum likelihood egomotion transform.

4.4.2 Generalized Kernels

The technique of generalized kernels (Vega-Brown et al., 2014) combines the benefits of kernel density estimation with Bayesian inference. The basic idea is as follows. Consider a dataset of inputs, $\mathbf{x}$, and outputs, $\mathbf{y}$, and a dataset of independent observations $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_N, \mathbf{y}_N)\}$. We are given a new 'test' input $\mathbf{x}^*$, and are asked to infer the likelihood of a observing a given output at this input:

$$p(\mathbf{y}|\mathbf{x}^*, \mathcal{D}). \tag{4.7}$$

If we associate a set of latent parameters, $\boldsymbol{\pi}$, with each input $\mathbf{x}$, and assume a known likelihood function $p(\mathbf{y}|\boldsymbol{\pi})$, we can infer a distribution over $\boldsymbol{\pi}$ and then marginalize it out to arrive at the desired likelihood

$$p(\mathbf{y}|\mathbf{x}^*, \mathcal{D}) = \int_{\boldsymbol{\pi}} \underbrace{p(\mathbf{y}|\boldsymbol{\pi}^*)}_{\text{Known likelihood function}} \underbrace{p(\boldsymbol{\pi}^*|\mathbf{x}^*, \mathcal{D})}_{\text{Parameter posterior}} d\boldsymbol{\pi}^*. \tag{4.8}$$

This is called the posterior predictive distribution. Using Bayes rule, we can write

$$p(\boldsymbol{\pi}^*|\mathbf{x}^*, \mathcal{D}) \propto \int \left(\prod_{i=1}^{N} p(\mathbf{y}_i|\boldsymbol{\pi}_i) d\boldsymbol{\pi}_i \right) p(\boldsymbol{\pi}_{1:N}, \boldsymbol{\pi}^*|\mathbf{x}^*, \mathbf{x}_{1:N}) \tag{4.9}$$

The technique of generalized kernels makes the assumption that the parameters $\boldsymbol{\pi}_{1:N}$ are conditionally independent given the target parameters, $\boldsymbol{\pi}^*$. This gives the distribution:

$$p(\boldsymbol{\pi}_{1:N}, \boldsymbol{\pi}^*|\mathbf{x}^*, \mathbf{x}_{1:N}) = \left(\prod_{i=1}^{N} p(\boldsymbol{\pi}_i|\boldsymbol{\pi}^*\mathbf{x}^*, \mathbf{x}_i) \right) p(\boldsymbol{\pi}^*|\mathbf{x}^*) \tag{4.10}$$

which combined with Equation (4.9) results in

$$p(\boldsymbol{\pi}^*|\mathbf{x}^*, \mathcal{D}) \propto \prod_{i=1}^{N} \underbrace{p(\mathbf{y}_i|\boldsymbol{\pi}^*, \mathbf{x}_i, \mathbf{x}^*)}_{\text{Extended likelihood}} \underbrace{p(\boldsymbol{\pi}^*|\mathbf{x}^*)}_{\text{Prior}}. \tag{4.11}$$

Now, the piece-de-resistance of generalized kernels is that the *extended* likelihood can be written as function of the known likelihood $p(\mathbf{y}_i|\boldsymbol{\pi}_i)$ if we assume it is the maximum entropy distribution whose information divergence from the likelihood is bounded by the metric $\rho(\mathbf{x}^*, \mathbf{x}_i)$. Specifically, in Vega-Brown et al. (2014), it is shown that in this case, the extended

likelihood must have the form:

$$p(\mathbf{y}|\boldsymbol{\pi}^*, \mathbf{x}, \mathbf{x}^*) \propto p(\mathbf{y}|\boldsymbol{\pi})^{k(\mathbf{x}^*, \mathbf{x})}, \tag{4.12}$$

where $k(\cdot, \cdot)$ is a kernel function[2] that is uniquely defined by ρ. The intuition behind this is that we expect the extended likelihood to equal the known likelihood if $\mathbf{x}^* = \mathbf{x}_i$ (and therefore $\boldsymbol{\pi}^* = \boldsymbol{\pi}_i$, resulting in $p(\mathbf{y}_i|\boldsymbol{\pi}^*, \mathbf{x}_i, \mathbf{x}^*) = p(\mathbf{y}_i|\boldsymbol{\pi}_i)$) and diverge in some smooth way when $\mathbf{x}^* \neq \mathbf{x}_i$. Combining Equation (4.11) with Equation (4.12), we arrive at an expression for the posterior over parameters as

$$p(\boldsymbol{\pi}|\mathbf{x}, \mathcal{D}) \propto \prod_{i=1}^{N} p(\mathbf{y}|\boldsymbol{\pi})^{k(\mathbf{x}, \mathbf{x}_i)} p(\boldsymbol{\pi}|\mathbf{x}), \tag{4.13}$$

which can be evaluated in closed form for appropriate an appropriate likelihood and prior. Namely, for PROBE, we will assume Gaussian likelihoods for the reprojection errors (and therefore the observations $\mathbf{y}_{i,c_1}$), and inverse Wishart priors for covariance matrices (this will result in inverse Wishart posteriors due to conjugacy). The input, $\mathbf{x}$, will be the vector of predictors $\boldsymbol{\phi}$.

4.4.3 Generalized Kernels for Visual Odometry

In order to exploit conjugacy to a Gaussian noise model, we formulate our prior knowledge about this function using an inverse Wishart (IW) distribution over positive definite $d \times d$ matrices (the IW distribution has been used as a prior on covariance matrices in other robotics and computer vision contexts, see for example, (Fitzgibbon et al., 2007)). This distribution is defined by a scale matrix $\boldsymbol{\Psi} \in \mathbb{R}^{d \times d} \succ 0$ and a scalar quantity called the degrees of freedom $\nu \in \mathbb{R} > d - 1$:

$$\begin{aligned} p(\mathbf{R}) &= \text{IW}(\mathbf{R}; \boldsymbol{\Psi}, \nu) \\ &= \frac{|\boldsymbol{\Psi}|^{\nu/2}}{2^{\frac{\nu d}{2}} \Gamma_d(\frac{\nu}{2})} |\mathbf{R}|^{-\frac{\nu+d+1}{2}} \exp\left(-\frac{1}{2}\text{tr}\left(\boldsymbol{\Psi}\mathbf{R}^{-1}\right)\right). \end{aligned} \tag{4.14}$$

We use the scale matrix to encode our prior estimate of the covariance, and the degrees of freedom to encode our confidence in that estimate. Specifically, if we estimate the covariance $\mathbf{R}$ associated with predictor $\boldsymbol{\phi}$ to be $\hat{\mathbf{R}}$ with a confidence equivalent to seeing n independent samples of the error from $\mathcal{N}(\mathbf{0}, \hat{\mathbf{R}})$, we would choose $\nu(\boldsymbol{\phi}) = n$ and $\boldsymbol{\Psi}(\boldsymbol{\phi}) = n\hat{\mathbf{R}}$. Given a

[2] i.e., $k(\mathbf{x}, \mathbf{x}) = 1 \forall \mathbf{x}$ and $k(\mathbf{x}, \mathbf{x}') \in [0, 1] \forall \mathbf{x}, \mathbf{x}'$.

sequence of observations and ground truth transformations,

$$\mathcal{D} = \{\mathcal{I}_t, \mathbf{T}_t\}, \quad t \in [1, N] \tag{4.15}$$

where

$$\mathcal{I}_t = \{\mathbf{y}_{i,c_0}, \mathbf{y}_{i,c_1}, \boldsymbol{\phi}_{i,t}\} \quad i \in [1, N_t], \tag{4.16}$$

we can use the procedure of generalized kernel estimation as described above to infer a posterior distribution over the covariance matrix $\mathbf{R}_*$ associated with some query predictor vector $\boldsymbol{\phi}_*$:

$$\begin{aligned} p(\mathbf{R}_*|\mathcal{D}, \boldsymbol{\phi}_*) \propto \prod_{i,t} \mathcal{N}(\mathbf{e}_{i,t}|\mathbf{0}, \mathbf{R}_*)^{k(\boldsymbol{\phi}_*, \boldsymbol{\phi}_{i,t})} \\ \times \mathrm{IW}(\mathbf{R}_*; \boldsymbol{\Psi}(\boldsymbol{\phi}_*), \nu(\boldsymbol{\phi}_*)) \qquad (4.17) \\ = \mathrm{IW}(\mathbf{R}_*; \boldsymbol{\Psi}_*, \nu_*). \qquad (4.18) \end{aligned}$$

Here, $\mathbf{e}_{i,t} = \mathbf{y}_{i,c_1} - \mathbf{f}(\mathbf{T}_t \mathbf{f}^{-1}(\mathbf{y}_{i,c_0}))$ as before. The function $k : \mathbb{R}^M \times \mathbb{R}^M \to [0, 1]$ is a kernel function which measures the similarity of two points in predictor space. Note also that the posterior parameters $\boldsymbol{\Psi}_*$ and ν_* can be computed in closed form (see Vega-Brown et al. (2014)) as

$$\boldsymbol{\Psi}_* = \boldsymbol{\Psi}(\boldsymbol{\phi}_*) + \sum_{i,t} k(\boldsymbol{\phi}_*, \boldsymbol{\phi}_{i,t}) \mathbf{e}_{i,t}\ {\mathbf{e}_{i,t}}^T, \tag{4.19}$$

$$\nu_* = \nu(\boldsymbol{\phi}_*) + \sum_{i,t} k(\boldsymbol{\phi}_*, \boldsymbol{\phi}_{i,t}). \tag{4.20}$$

If we marginalize over the covariance matrix, we find that the posterior predictive distribution is a multivariate Student's t distribution:

$$\begin{aligned} p(\mathbf{y}_{i,c_1}|\mathbf{T}_t, \mathbf{y}_{i,c_0}, \mathcal{D}, \boldsymbol{\phi}_{i,t}) \qquad (4.21) \\ = \int \mathrm{d}\mathbf{R}_{i,t} \mathcal{N}\left(\mathbf{e}_{i,t}; \mathbf{0}, \mathbf{R}_{i,t}\right) \mathrm{IW}(\mathbf{R}_{i,t}; \boldsymbol{\Psi}_*, \nu_*) \qquad (4.22) \\ = \mathrm{t}_{\nu_*-d+1}\left(\mathbf{e}_{i,t}; \mathbf{0}, \frac{1}{\nu_* - d + 1}\boldsymbol{\Psi}_*\right) \qquad (4.23) \\ = \frac{\Gamma(\frac{\nu_*+1}{2})}{\Gamma(\frac{\nu_*-d+1}{2})} |\boldsymbol{\Psi}_*|^{-\frac{1}{2}} \pi^{-\frac{d}{2}} \left(1 + {\mathbf{e}_{i,t}}^T {\boldsymbol{\Psi}_*}^{-1} \mathbf{e}_{i,t}\right)^{-\frac{\nu_*+1}{2}}. \qquad (4.24) \end{aligned}$$

Given a new landmark and predictor vector, we can infer a noise model by evaluating eqs. (4.19) and (4.20). In order to accelerate this computation, it is helpful to choose a kernel function

with finite support: that is, $k(\boldsymbol{\phi}, \boldsymbol{\phi}') = 0$ if $\|\boldsymbol{\phi} - \boldsymbol{\phi}'\|_2 > \rho$. Then, by indexing our training data in a spatial index such as a k-d tree, we can identify the subset of samples relevant to evaluating the sums in eqs. (4.19) and (4.20) in $\mathcal{O}(\log N + \log N_t)$ time. Algorithm 1 describes the procedure for building this model.

Algorithm 1 Build the covariance model given a sequence of observations, $\mathcal{D}$.

```
function BuildCovarianceModel(D)
    Initialize an empty spatial index M
    for all I_t, T_t in D do
        for all {y_{i,c0}, y_{i,c1}, φ_{i,c0}} in I_t do
            e_{i,t} = y_{i,c1} − f(T_t f^{-1}(y_{i,c0}))
            Insert φ_{i,t} into M and store e_{i,t} at its location
        end for
    end for
    return M
end function
```

Once we have inferred a noise model for each landmark in a new image pair, the maximum likelihood optimization problem is given by

$$\mathbf{T}_t^* = \underset{\mathbf{T}_t \in \mathrm{SE}(3)}{\operatorname{argmin}} \sum_{i=1}^{N_t} (\nu_{i,t} + 1) \log\left(1 + {\mathbf{e}_{i,t}}^T \boldsymbol{\Psi}_{i,t}^{-1} \mathbf{e}_{i,t}\right). \tag{4.25}$$

The final optimization problem thus emerges as a nonlinear least squares problem with a rescaled Cauchy-like loss function, with error term ${\mathbf{e}_{i,t}}^T (\frac{1}{\nu_{i,t}+1} \boldsymbol{\Psi}_{i,t})^{-1} \mathbf{e}_{i,t}$ and outlier scale $\nu_{i,t} + 1$. This is a common robust loss function which is approximately quadratic in the reprojection error for ${\mathbf{e}_{i,t}}^T \boldsymbol{\Psi}_{i,t}^{-1} \mathbf{e}_{i,t} \ll \nu_{i,t} + 1$, but grows only logarithmically for ${\mathbf{e}_{i,t}}^T \boldsymbol{\Psi}_{i,t}^{-1} \mathbf{e}_{i,t} \gg \nu_{i,t} + 1$. It follows that in the limit of large $\nu_{i,t}$—in regions of predictor space where there are many relevant samples—our optimization problem becomes the original least-squares optimization problem.

Solving nonlinear optimization problems with the form of Equation (4.25) is a well-studied and well-understood task, and software packages to perform this computation are readily available. Algorithm 2 describes the procedure for computing the transform between a new image pair, treating the optimization of Equation (4.25) as a subroutine.

We observe that Algorithm 2 is predictively robust, in the sense that it uses past experiences not just to predict the reliability of a given image landmark, but also to introspect and estimate its own knowledge of that reliability. Landmarks which are not known to be reliable are trusted less than landmarks which look like those which have been observed previously, where "looks like" is defined by our prediction space and choice of kernel.

Algorithm 2 Compute the transform between two images, given a set, $\mathcal{I}_t$, of landmarks and predictors extracted from an image pair and a covariance model $\mathcal{M}$.

function COMPUTETRANSFORM($\mathcal{I}_t$, $\mathcal{M}$)
  for all $\{\mathbf{y}_{i,c_0}, \mathbf{y}_{i,c_1}, \boldsymbol{\phi}_{i,c_0}\}$ in $\mathcal{I}_t$ **do**
    $\boldsymbol{\Psi}, \nu \leftarrow$ INFERNOISEMODEL($\mathcal{M}$, $\boldsymbol{\phi}_{i,t}$)
    $g(\mathbf{T}) = \mathbf{y}_{i,c_1} - \mathbf{f}(\mathbf{T}\mathbf{f}^{-1}(\mathbf{y}_{i,c_0}))$
    $\mathcal{L} \leftarrow \mathcal{L} + (\nu + 1) \log \left(1 + g(\mathbf{T})^T \boldsymbol{\Psi}^{-1} g(\mathbf{T})\right)$
  end for
  return $\operatorname{argmin}_{\mathbf{T} \in \mathrm{SE}(3)} \mathcal{L}(\mathbf{T})$
end function
function INFERNOISEMODEL($\mathcal{M}$, $\boldsymbol{\phi}_*$)
  NEIGHBORS $\leftarrow$ GETNEIGHBORS($\mathcal{M}$, $\boldsymbol{\phi}_*$, ρ) ▷ ρ is the radius of support of kernel k
  $\boldsymbol{\Psi}_* \leftarrow \boldsymbol{\Psi}(\boldsymbol{\phi}_*)$
  $\nu_* \leftarrow \nu(\boldsymbol{\phi}_*)$
  for $(\boldsymbol{\phi}_{i,t}, \mathbf{e}_{i,t})$ in NEIGHBORS **do**
    $\boldsymbol{\Psi}_* \leftarrow \boldsymbol{\Psi}_* + k(\boldsymbol{\phi}_*, \boldsymbol{\phi}_{i,t}) \mathbf{e}_{i,t}\, \mathbf{e}_{i,t}^T$
    $\nu_* \leftarrow \nu_* + k(\boldsymbol{\phi}_*, \boldsymbol{\phi}_{i,t})$
  end for
  return $\boldsymbol{\Psi}_*, \nu_*$
end function

4.4.4 Inference without ground truth

Algorithm 1 requires access to the true transform between training image pairs. In practice, such ground truth data may be difficult to obtain. In these cases, we can instead formulate a likelihood model $p(\mathcal{D}'|\mathbf{T}_1, \ldots, \mathbf{T}_t)$, where $\mathcal{D}' = \{\mathcal{I}_t\}$ is a dataset consisting only of landmarks and predictors for each training image pair. We can construct a model for future queries by inferring the most likely sequence of transforms for our training images. The likelihood has the following factorized form:

$$p(\mathcal{D}'|\mathbf{T}_{1:T}) \propto \int \prod_{i,t} \mathrm{d}\mathbf{R}_{i,t}\, p(\mathbf{y}_{i,c_1}|\mathbf{y}_{i,c_0}, \mathbf{T}_t, \mathbf{R}_{i,t}) p(\mathbf{R}_{i,t}|\boldsymbol{\phi}_{i,t}, \mathcal{D}, \mathbf{T}_{1:T}). \tag{4.26}$$

We cannot easily maximize this likelihood, since marginalizing over the noise covariances removes the independence of the transforms between each image pair. To render the optimization tractable, we follow previous work (Vega-Brown and Roy, 2013) and formulate an iterative expectation-maximization (EM) procedure. Given an estimate $\mathbf{T}_t^{(n)}$ of the transforms, we can compute the expected log-likelihood conditioned on our current estimate:

$$Q(\mathbf{T}_{1:T}|\mathbf{T}_{1:T}^{(n)}) = \int \left(\prod_{i,t} \mathrm{d}\mathbf{R}_{i,t}\, p(\mathbf{R}_{i,t}|\mathcal{D}_{\setminus i,t}, \mathbf{T}_{1:T}^{(n)}) \right) \log \prod_{i,t} p(\mathbf{y}_{i,c_1}|\mathbf{y}_{i,c_0}, \mathbf{T}_t, \mathbf{R}_{i,t}). \tag{4.27}$$

This has the effect of rendering the likelihood of each transform to be estimated independently. Moreover, the expected log-likelihood can be evaluated in closed form:

$$Q(\mathbf{T}_{1:T}|\mathbf{T}_{1:T}^{(n)}) \cong -\frac{1}{2}\sum_{t=1}^{T}\sum_{i=1}^{N_t} \mathbf{e}_{i,t}{}^T \left(\frac{1}{\nu_{i,t}^{(n)}}\mathbf{\Psi}_{i,t}^{(n)}\right)^{-1} \mathbf{e}_{i,t}. \tag{4.28}$$

The symbol $\cong$ is used to indicate equality up to an additive constant. We can iteratively refine our estimate by maximizing the expected log-likelihood

$$\mathbf{T}_{1:T}^{(n+1)} = \underset{\mathbf{T}_{1:T}\in \mathrm{SE}(3)^T}{\operatorname{argmax}} Q(\mathbf{T}_{1:T}|\mathbf{T}_{1:T}^{(n)}). \tag{4.29}$$

Due to the additive structure of $Q(\mathbf{T}_{1:T}|\mathbf{T}_{1:T}^{(n)})$, this takes the form of T separate nonlinear least-squares optimizations:

$$\mathbf{T}_{t}^{(n+1)} = \underset{\mathbf{T}_t\in \mathrm{SE}(3)}{\operatorname{argmin}} \sum_{i=1}^{N_t} \mathbf{e}_{i,t}{}^T \left(\frac{1}{\nu_{i,t}^{(n)}}\mathbf{\Psi}_{i,t}^{(n)}\right)^{-1} \mathbf{e}_{i,t}. \tag{4.30}$$

Algorithm 3 describes the process of training a model without ground truth. We refer to this process as PROBE-GK-EM, and distinguish it from PROBE-GK-GT (Ground Truth). We note that the sequence of estimated transforms, $\mathbf{T}_{1:T}^{(n)}$, is guaranteed to converge to a local maxima of the likelihood function (Dempster et al., 1977). It is also possible to use a robust loss function (Equation (4.25)) in place of Equation (4.30) during EM training. Although not formally motivated by the derivation above, this approach often leads to lower test errors in practice. Characterizing when and why this robust learning process outperforms its non-robust alternative is outside the scope of this dissertation.

4.5 Prediction Space

A crucial component of our technique is the choice of the vector of predictors $\boldsymbol{\phi}$. In practice, feature tracking quality is often degraded by a variety of effects such as motion blur, moving objects, and textureless or self-similar image regions. The challenge is in determining predictors that account for such effects without requiring excessive computation. In our implementation, we use the following predictors, but stress that the choice of predictors can be tailored to suit particular applications and environments:

- Angular velocity and linear acceleration magnitudes
- Local image entropy

Algorithm 3 Build the covariance model without ground truth given a sequence of observations, $\mathcal{D}'$, and an initial odometry estimate $\mathbf{T}_{1:T}^{(0)}$.

function BUILDCOVARIANCEMODEL($\mathcal{D}'$, $\mathbf{T}_{1:T}^{(0)}$)
  Initialize an empty spatial index $\mathcal{M}$
  for all $\mathcal{I}_t$ in $\mathcal{D}'$ **do**
    for all $\{\mathbf{y}_{i,c_0}, \mathbf{y}_{i,c_1}, \boldsymbol{\phi}_{i,t}\}$ in $\mathcal{I}_t$ **do**
      $\mathbf{e}_{i,t} = \mathbf{y}_{i,c_1} - \mathbf{f}(\mathbf{T}_t^{(0)}\mathbf{f}^{-1}(\mathbf{y}_{i,c_0}))$
      Insert $\boldsymbol{\phi}_{i,t}$ into $\mathcal{M}$ and store $\mathbf{e}_{i,t}$ at its location
    end for
  end for
  repeat
    for all $\mathcal{I}_t$ in $\mathcal{D}'$ **do**
      for all $\{\mathbf{y}_{i,c_0}, \mathbf{y}_{i,c_1}, \boldsymbol{\phi}_{i,t}\}$ in $\mathcal{I}_t$ **do**
        $\boldsymbol{\Psi}, \nu \leftarrow$ INFERNOISEMODEL($\mathcal{M}, \boldsymbol{\phi}_{i,t}$)
        $g(\mathbf{T}) = \mathbf{y}_{i,c_1} - \mathbf{f}(\mathbf{T}\mathbf{f}^{-1}(\mathbf{y}_{i,c_0}))$
        $\mathcal{L} \leftarrow \mathcal{L} + g(\mathbf{T})^T \left(\frac{1}{\nu}\boldsymbol{\Psi}\right)^{-1} g(\mathbf{T})$
      end for
      $\mathbf{T}_t \leftarrow \operatorname{argmin}_{\mathbf{T}\in\mathrm{SE}(3)} \mathcal{L}(\mathbf{T})$
      $\mathbf{e}_{i,t} = \mathbf{y}_{i,c_1} - \mathbf{f}(\mathbf{T}_t^{(0)}\mathbf{f}^{-1}(\mathbf{y}_{i,c_0}))$
      Update the error stored at $\boldsymbol{\phi}_{i,t}$ in $\mathcal{M}$ to $\mathbf{e}_{i,t}$
    end for
  until converged
  return $\mathcal{M}$
end function

- Blur (quantified by the blur metric of Crete et al. (2007))
- Optical flow variance score
- Image frequency composition

We discuss each of these predictors in turn.

4.5.1 Angular velocity and linear acceleration

While most of the predictors in our system are computed directly from image data, the magnitudes of the angular velocities and linear accelerations reported by an IMU (if available) are in themselves good predictors of image degradation (e.g., image blur) and hence poor feature tracking. We do not explicitly correct for bias in linear accelerations because we expect real motion-induced acceleration to trump bias at the timescales of our test trials. As a result, there is virtually no computational cost involved in incorporating these quantities as predictors.

4.5.2 Local image entropy

Entropy is a statistical measure of randomness that can be used to characterize the texture in an image or patch. Since the quality of feature detection is strongly influenced by the strength of the texture in the vicinity of the feature point, we expect the entropy of a patch centered on the feature to be a good predictor of its quality. We evaluate the entropy S in an image patch by sorting pixel intensities into N bins and computing

$$S = -\sum_{i=1}^{N} c_i \log_2(c_i), \tag{4.31}$$

where c_i is the number of pixels counted in the i^{th} bin.

4.5.3 Blur

Blur can arise from a number of sources including motion, dirty lenses, and sensor defects. All of these have deleterious effects on feature tracking quality. To assess the effect of blur in detail, we performed a separate experiment. We recorded images of 32 interior corners of a standard checkerboard calibration target using a low frame-rate (20 FPS) Skybotix VI-Sensor stereo camera and a high frame-rate (125 FPS) Point Grey Flea3 monocular camera rigidly connected by a bar (Figure 4.2). Prior to the experiment, we determined the intrinsic and extrinsic calibration parameters of our rig using the KALIBR[3] package Furgale et al. (2013). The

[3] https://github.com/ethz-asl/kalibr

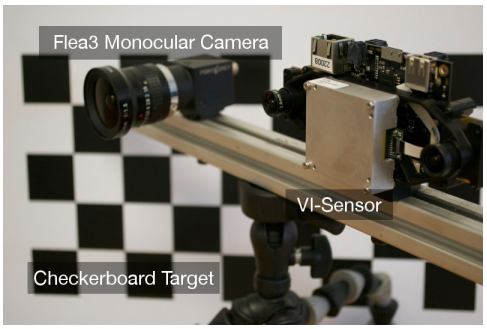


Figure 4.2: The Skybotix VI-Sensor, Point Grey Flea3, and checkerboard target used in our motion blur experiments.

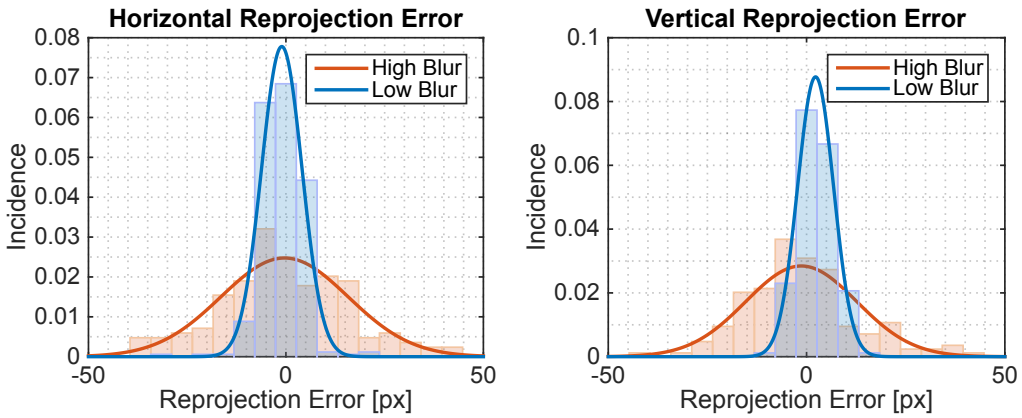


Figure 4.3: Reprojection error of checkerboard corners triangulated from the VI-Sensor and reprojected into the Flea3.We distinguish between high and low blur by thresholding the blur metric Crete et al. (2007).

apparatus underwent both slow and fast translational and rotational motion, which induced different levels of motion blur as quantified by the blur metric proposed by Crete et al. (2007).

We detected checkerboard corners in each camera at synchronized time steps, computed their 3D coordinates in the VI-Sensor frame, then reprojected these 3D coordinates into the Flea3 frame. We then computed the reprojection error as the distance between the reprojected image coordinates and the true image coordinates in the Flea3 frame. Since the Flea3 operated at a much higher frame rate than the VI-Sensor, it was less susceptible to motion blur and so we treated its observations as ground truth. We also computed a tracking error by comparing the image coordinates of checkerboard corners in the left camera of the VI-Sensor computed from both KLT tracking Lucas and Kanade (1981) and re-detection.

Figure 4.4 shows histograms and fitted normal distributions for both reprojection error and tracking error. From these distributions we can see that the errors remain approximately zero-mean, but that their variance increases with blur. This result is compelling evidence that the effect of blur on feature tracking quality can be accounted for by scaling the feature

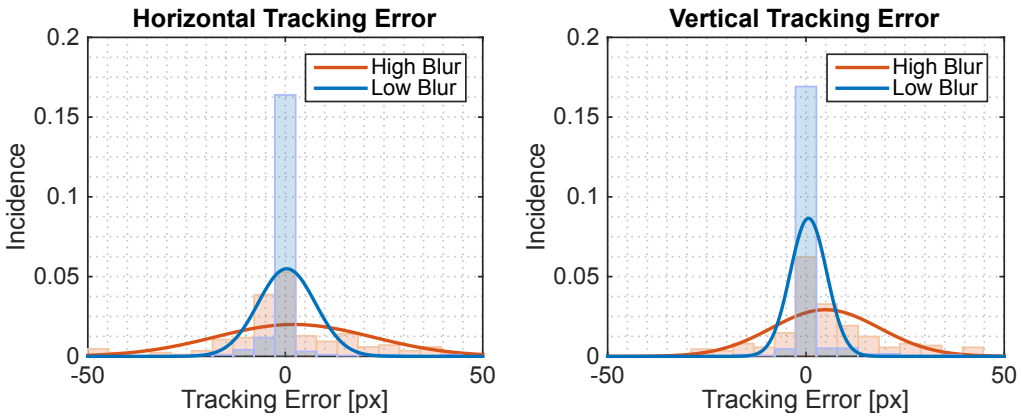


Figure 4.4: Effect of blur on reprojection and tracking error for the slow-then-fast checkerboard dataset. We distinguish between high and low blur by thresholding the blur metric Crete et al. (2007). The variance in both errors increases with blur.

covariance matrix by a function of the blur metric.

4.5.4 Optical flow variance

To detect moving objects, we compute a score for each feature based on the ratio of the variance in optical flow vectors in a small region around the feature to the variance in flow vectors of a larger region. Intuitively, if the flow variance in the small region differs significantly from that in the larger region, we might expect the feature in question to belong to a moving object, and we would therefore like to trust the feature less. Since we consider only the variance in optical flow vectors, we expect this predictor to be reasonably invariant to scene geometry.

We compute this optical flow variance score according to

$$\log\left(\frac{\bar{\sigma}_s^2}{\bar{\sigma}_l^2}\right), \tag{4.32}$$

where $\bar{\sigma}_s^2, \bar{\sigma}_l^2$ are the means of the variance of the vertical and horizontal optical flow vector components in the small and large regions respectively. Figure 4.5 shows sample results of this scoring procedure for two images in the KITTI dataset. Our optical flow variance score generally picks out moving objects such as vehicles and cyclists in diverse scenes.

4.5.5 Image frequency composition

Reliable feature tracking is often difficult in textureless or self-similar environments due to low feature counts and false matches. We detect textureless and self-similar image regions by computing the Fast Fourier Transform (FFT) of each image and analyzing its frequency com-

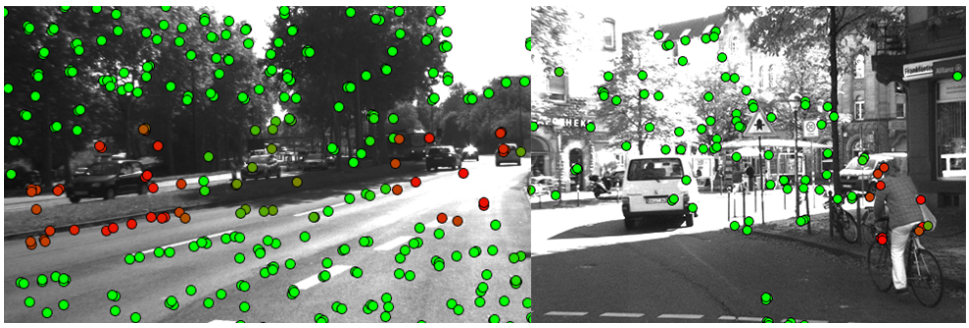


Figure 4.5: The optical flow variance predictor can help in detecting moving objects. Red circles correspond to higher values of the optical flow variance score (i.e., features more likely to belong to a moving object).

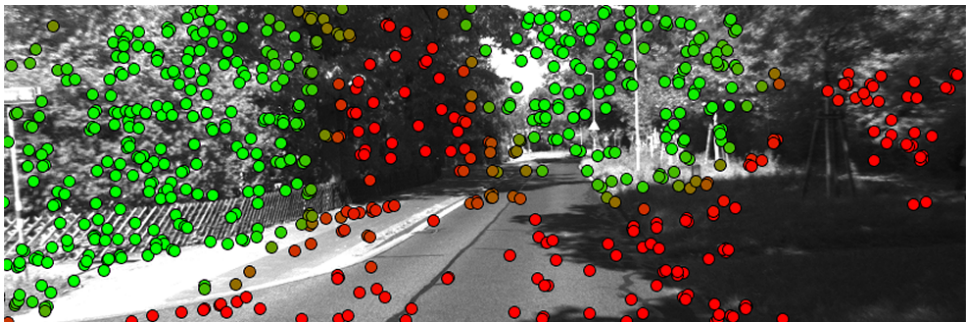


Figure 4.6: A high-frequency predictor can distinguish between regions of high and low texture such as foliage and shadows. Green indicates higher values.

position. For each feature, we compute a coefficient for the low- and high-frequency regimes of the FFT. Figure 4.6 shows the result of the high-frequency version of this predictor on a sample image from the KITTI dataset. Our high-frequency predictor effectively distinguishes between textureless regions (e.g., shadows and roads) and texture-rich regions (e.g., foliage).

4.6 Experiments

To validate PROBE-GK, we used three types of data: synthetic simulations, the KITTI dataset, and our own experimental data collected at the University of Toronto.

4.6.1 Simulation

Monte-Carlo Verification

To begin, we verified that PROBE-GK can predict increasingly accurate estimates of the true error covariance as more training data is added. We developed a basic simulation environment consisting of a large amount of point landmarks being observed by a stereo camera. In

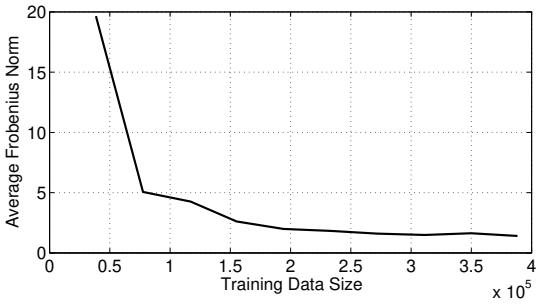


Figure 4.7: Mean Frobenius norm of the error between the estimated and true noise covariance as a function of training data size. The norm tends to zero as training data is added which indicates that PROBE-GK is learning the correct covariances.

our simulation, the camera traversed a single step in one direction, and recorded empirical reprojection errors based on ground truth poses. We simulated additive Gaussian noise on image coordinates, and used Monte Carlo simulations (propagating the additive noise through Equation (3.10)) to estimate the true covariances. Figure 4.7 shows the mean Frobenius norm (as defined in Barfoot and Furgale (2014b)) between the covariances estimated by PROBE-GK and the true covariances for a test trial. The mean norm tends to zero as more landmarks are added, indicating that PROBE-GK does learn the correct covariances.

Synthetic

Next, we formulated a synthetic dataset wherein a stereo camera traverses a circular path observing 2000 randomly distributed point features. We added Gaussian noise to each of the ideal projected pixel co-ordinates for visible landmarks at every step. We varied the noise variance as a function of the vertical pixel coordinate of the feature in image space. In addition, a small subset of the landmarks received an error term drawn from a uniform distribution to simulate the presence of outliers. The prediction space was composed of the vertical and horizontal pixel locations in each of the stereo cameras.

We simulated independent training and test traversals, where the camera moved for 30 and 60 seconds respectively (at a forward speed of 3 metres per second for final path lengths of 90 and 180 meters). Figure 4.9 and Table 4.1 document the qualitative and quantitative comparisons of PROBE-GK (trained with and without ground-truth) against two baseline stereo odometry frameworks. Both baseline estimators were implemented based on **??**. The first utilized fixed covariances for all reprojection errors, while the second used a modified robust cost (i.e. M-estimation) based on Student's t weighting, with $\nu = 5$ (as suggested in Kerl et al. (2013)). These benchmarks served as baseline estimators (with and without robust costs) that used fixed covariance matrices and did not include a predictive component.

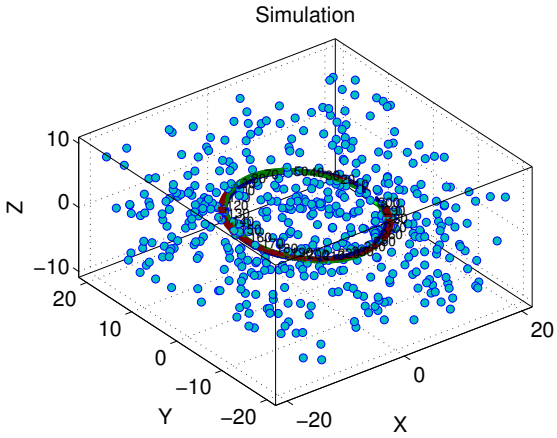


Figure 4.8: Our synthetic world. A stereo camera rig moves through a world with 2000 point features.

Using PROBE-GK with ground truth data for training, we significantly reduced both the translation and rotational Average Root Mean Squared Error (ARMSE) by approximately 50%. In our synthetic data, the Expectation Maximization approach was able to achieve nearly identical results to the ground-truth-aided model within 5 iterations.

4.6.2 KITTI

To evaluate PROBE-GK on real environments, we trained and tested several models on the KITTI Vision Benchmark suite (Geiger et al., 2012, 2013b), a series of datasets collected by a car outfitted with a number of sensors driven around different parts of Karlsruhe, Germany. Within the dataset, ground truth pose information is provided by a high grade inertial navigation unit which also fuses measurements from differential GPS. Raw data is available for different types of environments through which the car was driving; for our work, we focused on the city, residential and road categories (Figure 4.10). From each category, we chose two separate trials for training and testing.

Figures 4.11 to 4.13 show typical results; Table 4.1 presents a quantitative comparison. PROBE GK-GT produced significant reductions in ARMSE, reducing translational ARMSE by as much as 80%. In contrast, GK-EM showed more modest improvements; this is unlike our synthetic experiments, where both GK-EM and GK-GT achieved similar performance. We are still actively exploring why this is the case; we note that although our simulated data is drawn from a mixture of Gaussian distributions, the underlying noise distribution for real data may be far more complex. With no ground truth, EM has to jointly optimize the camera poses and sensor uncertainty. It is unclear whether this is feasible in the general case with no ground truth information.

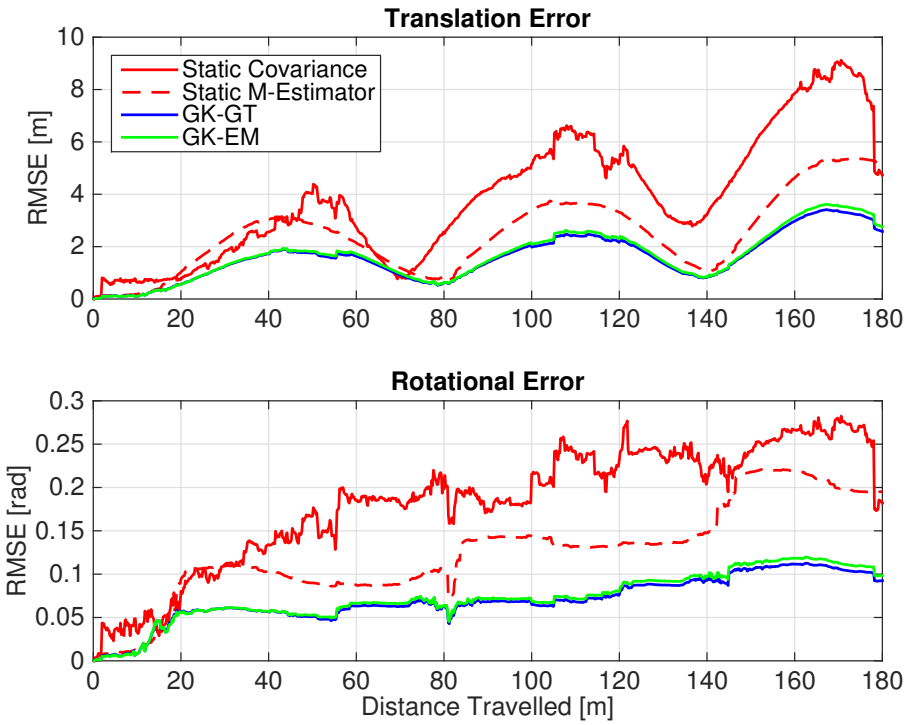


Figure 4.9: A comparison of translational and rotational Root Mean Square Error on simulated data (RMSE) for four different stereo-visual odometry pipelines: two baseline bundle adjustment procedures with and without a robust Student's t cost with a fixed and hand-tuned covariance and degrees of freedom (M-Estimation), a robust bundle adjustment with covariances learned from ground truth with algorithm 1 (GK-GT), and a robust bundle adjustment using covariances learned without ground truth using expectation maximization, with algorithm 3 (GK-EM). Note in this experiment, the RMSE curves for GK-GT and GK-EM very nearly overlap. The overall translational and rotational ARMSE values are shown in Table 4.1.

Figure 4.10: The KITTI dataset contains three different environments. We validate PROBE-GK by training on each type and testing against a baseline stereo visual odometry pipeline.

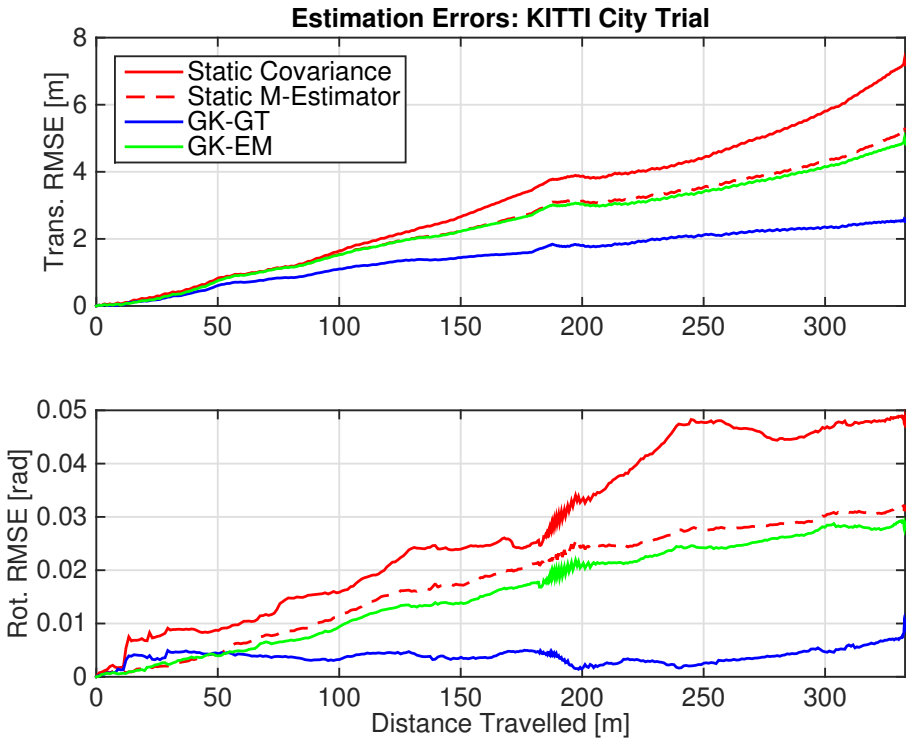


Figure 4.11: RMSE comparison of stereo odometry estimators evaluated on data from the city category in the KITTI dataset. See Table 4.1 for a quantitative summary.

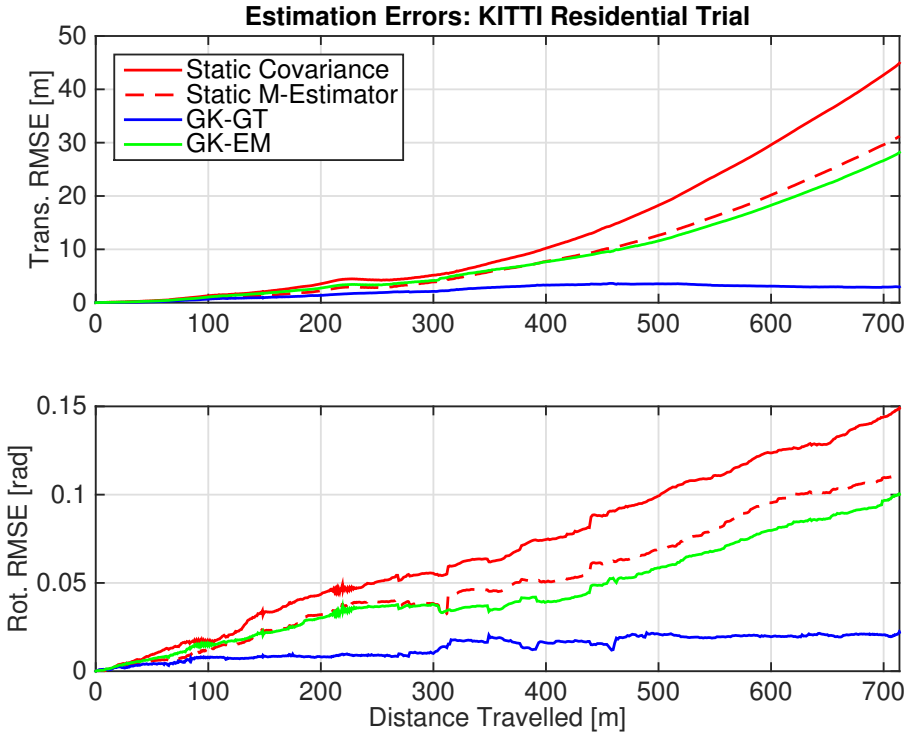


Figure 4.12: RMSE comparison of stereo odometry estimators evaluated on data from the residential category in the KITTI dataset.

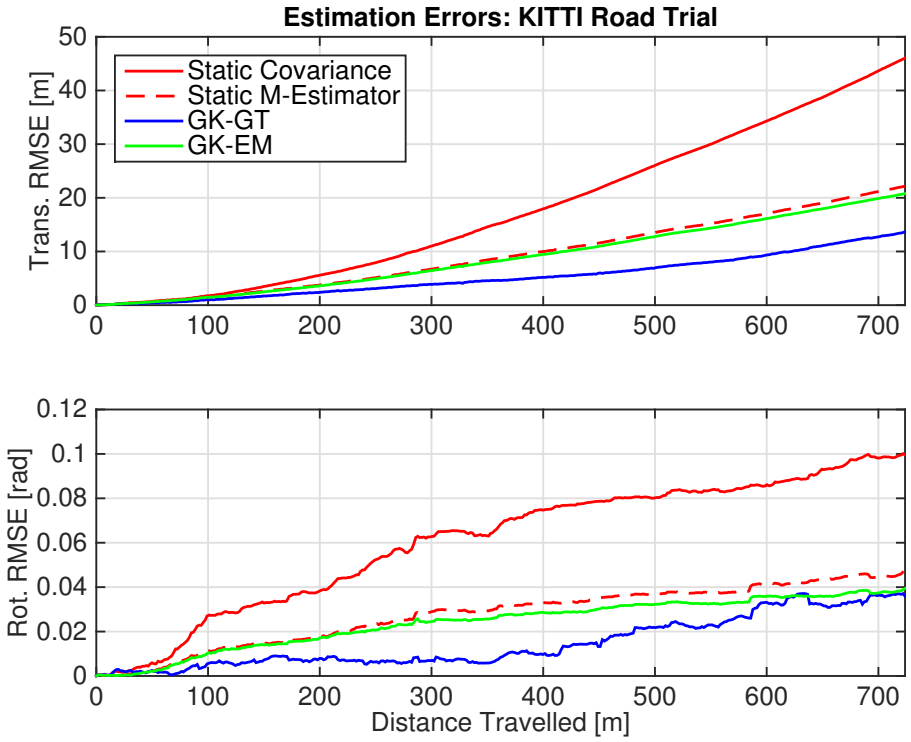


Figure 4.13: RMSE comparison of stereo odometry estimators evaluated on data from the road category in the KITTI dataset.

Table 4.1: Comparison of average root mean squared errors (ARMSE) for rotational and translational components. Each trial is trained and tested from a particular category of raw data from the synthetic and KITTI datasets.

		Trans. ARMSE [m]				Rot. ARMSE [rad]			
	Length [m]	Fixed Covar.	Static M-Estimator	GK-GT	GK-EM	Fixed Covar.	Static M-Estimator	GK-GT	GK-EM
Synthetic	180	3.87	2.49	1.59	1.66	0.18	0.13	0.070	0.073
City	332.9	3.84	2.99	1.69	2.87	0.032	0.021	0.0046	0.018
Residential	714.1	13.48	9.37	1.97	8.80	0.068	0.050	0.013	0.044
Road	723.8	17.69	9.38	5.24	8.87	0.060	0.027	0.015	0.024

Further, we observe that the performance of PROBE-GK depends on the similarity of the training data to the final test trials. A characteristic training dataset was important for consistent improvements on test trials.

4.6.3 UTIAS

To further investigate the capability of our EM approach, we evaluated PROBE-GK on experimental data collected at the University of Toronto Institute for Aerospace Studies (UTIAS). For this experiment, we drove a Clearpath Husky rover outfitted with an Ashtech DG14 Differential GPS, and a PointGrey XB3 stereo camera around the MarsDome (an indoor Mars analog testing environment) at UTIAS (Figure 4.14) for five trials of a similar path. Each trial

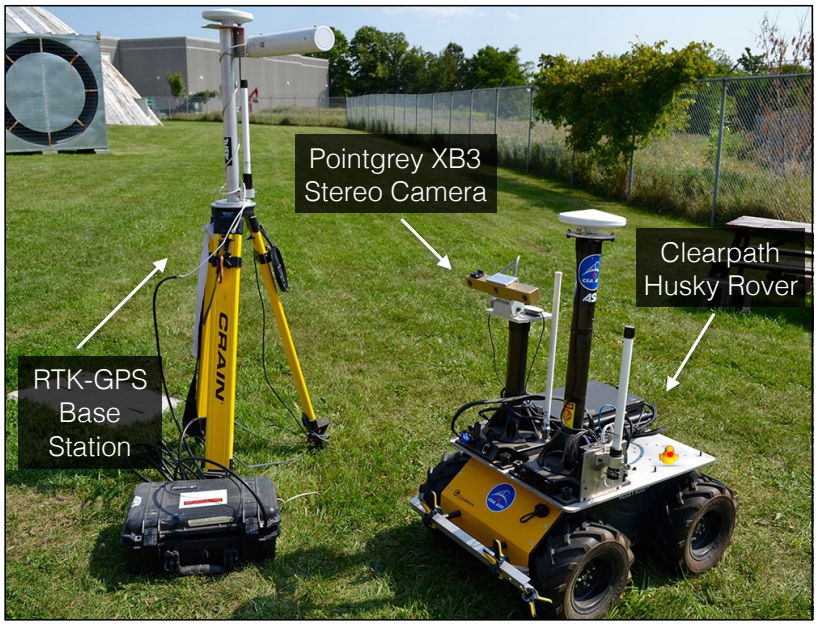


Figure 4.14: Our experimental apparatus: a Clearpath Husky rover outfitted with a PointGrey XB3 stereo camera and a differential GPS receiver and base station.

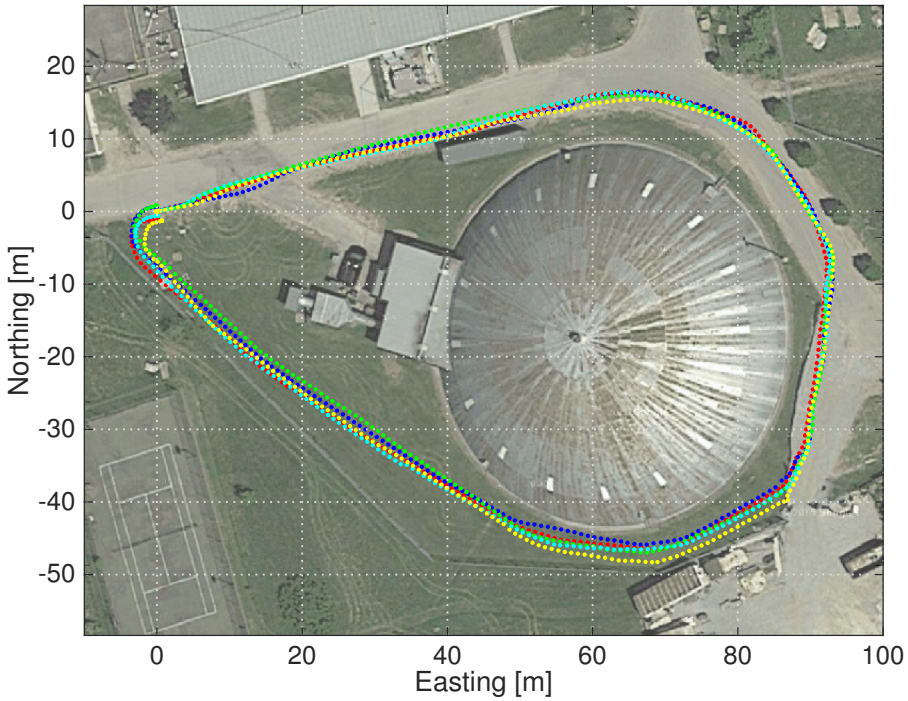


Figure 4.15: GPS ground truth for 5 experimental trials collected near the UTIAS Mars Dome. Each trial is approximately 250 m long.

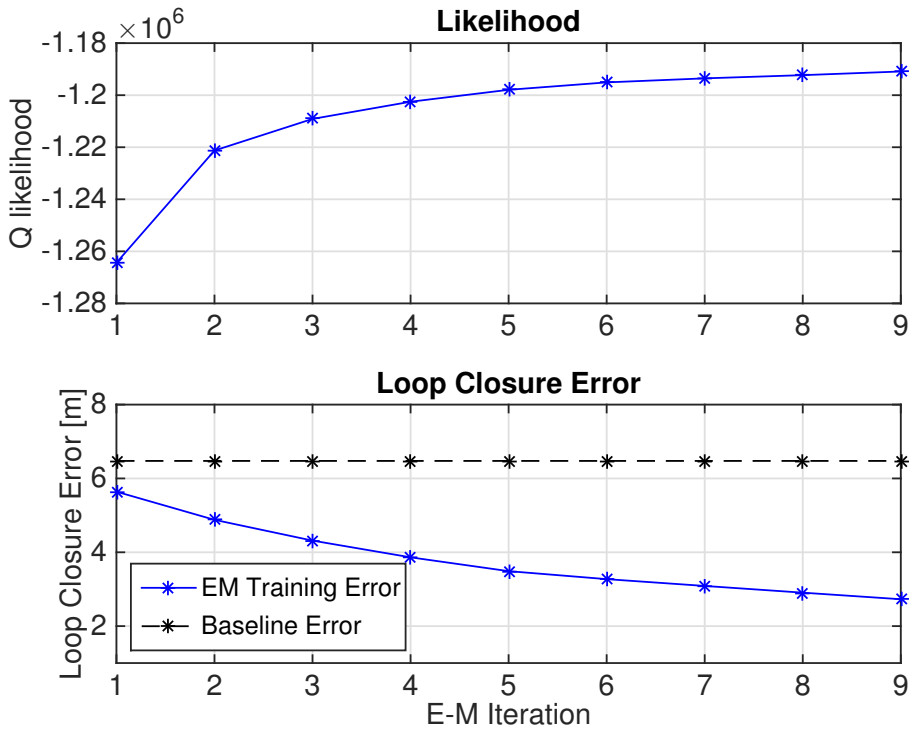


Figure 4.16: Training without ground truth using PROBE-GK-EM on a 250.2m path around the Mars Dome at UTIAS. The likelihood of the data increases with each iteration, and the loop closure error decreases, improving significantly from a baseline static M-estimator.

was approximately 250 m in length and we made an effort to align the start and end points of each loop. We used the wide baseline (25 cm) of the XB3 stereo camera to record the stereo images. The approximate trajectory for all 5 trials, as recorded by GPS, is shown in Figure 4.15. Note that the GPS data was not used during training, and only recorded for reference.

For the prediction space in our experiments, we mimicked the KITTI experiments, omitting inertial magnitudes as no inertial data was available. We trained PROBE-GK without ground truth, using the Expectation Maximization approach. Figure 4.16 shows the likelihood and loop closure error as a function of EM iteration.

The EM approach indeed produced significant error reductions on the training dataset after just a few iterations. Although it was trained with no ground truth information, our PROBE-GK model was used to produce significant reductions in the loop closure errors of the remaining 4 test trials. This reinforced our earlier hypothesis: the EM method works well when the training trajectory more closely resembles the test trials (as was the case in this experiment). Table 4.2 lists the statistics for each test.

Table 4.2: Comparison of loop closure errors for 4 different experimental trials with and without a learned PROBE-GK-EM model.

		Loop Closure Error [m]	
Trial	Path Length [m]	PROBE-GK-EM	Static M-Estimator
2	250.3	3.88	8.07
3	250.5	3.07	6.64
4	205.4	2.81	7.57
5	249.9	2.34	7.75

4.7 Summary

Predictive Robust Estimation (PROBE) applied Generalized Kernel estimation to improve on the uncorrelated and static Gaussian error models typically employed in stereo odometry. By building a non-parametric predictive model for the density of reprojection errors, we derived a robust least squares objective whose parameters were predicted based on training data. In summary, this chapter contributed

1. a probabilistic model for indirect stereo visual odometry, leading to a predictive robust algorithm for inference on that model,

2. an efficient approach to constructing the robust algorithm based on Generalized Kernel (GK) estimation,

3. a procedure for training our model using pairs of stereo images with known relative transforms, and

4. an iterative, expectation-maximization approach to train our GK model when the relative ground truth egomotion was unavailable.

Appendices

Appendix A

PROBE: K-NN

A.1 Theory

In our initial exploratory work, we explored the idea of scaling With Predictive ROBust Estimation, we aim to improve localization accuracy in the presence of such effects by building a model of the uncertainty in the affected visual observations. We learn the model in an offline training procedure and then use it online to predict the uncertainty of incoming observations as a function of their location in a predefined *prediction space*. Our model can be learned in completely unknown environments with frequent or infrequent ground truth data.

The primary contributions of this research are a flexible framework for learning the quality of visual features with respect to navigation estimates, and a straightforward way to incorporate this information into a navigation pipeline. On its own, PROBE can produce more accurate estimates than a binary outlier rejection scheme like Random Sample Consensus (RANSAC) because it can simultaneously reduce the influence of outliers while intelligently weighting inliers. PROBE reduces the need to develop finely-tuned uncertainty models for complex sensors such as cameras, and better accounts for the effects observed in complex, dynamic scenes than typical fixed-uncertainty models. While we present PROBE in the context of visual feature-based navigation, we stress that it is not limited to visual measurements and could also be applied to other sensor modalities.

The aim of PROBE is to learn a model for the quality of visual features, with the goal of reducing the impact of deleterious visual effects such as moving objects, motion blur, and shadows on navigation estimates. Feature quality is characterized by a scalar weight, β_i, for each visual feature in an environment. To compute β_i we define a prediction space (similar to Vega-Brown et al. (2013)) that consists of a set of visual-inertial predictors computed from the local image region around the feature and the inertial state of the vehicle (Section 4.5 details our choice of predictors). We then scale the image covariance of each feature by β_i during

the non-linear optimization.

In a similar manner to M-estimation, PROBE achieves robustness by varying the influence of certain measurements. However, in contrast to robust cost functions that weight measurements based purely on estimation error, PROBE weights measurements based on their assessed quality.

To learn the model, we require training data that consists of a traversal through a typical environment with some measure of ground truth for the path, but not for the visual features themselves. Like many machine learning techniques, we assume that the training data is representative of the test environments in which the learned model will be used.

We learn the quality of visual features *indirectly* through their effect on navigation estimates. We define high quality features as those that result in estimates that are close to ground truth. Our framework is flexible enough that we do not require ground truth at every image and we can learn the model based on even a single loop closure error.

A.1.1 Mathematical Formulation

To solve for the relative egomotion, $\mathbf{T}_t$, between two camera frames, $\underrightarrow{\mathcal{F}}_{c_0}$ and $\underrightarrow{\mathcal{F}}_{c_1}$, we follow technique described in Section 3.2.3 to convert stereo observations into point-clouds and then solve for the maximum likelihood $\mathrm{SE}(3)$ transformation. We associate with each match $\{\mathbf{y}_{i,c_0}, \mathbf{y}_{i,c_1}\}$ a vector of *predictors*, $\boldsymbol{\phi}_{i,t}$. We compute the covariance as a function of these predictors, so that $\mathbf{R}_{i,c_0} = \mathbf{R}_{i,c_1} = \mathbf{R}_{i,t} = \mathbf{R}(\boldsymbol{\phi}_{i,t})$, and we use the same covariance function for features in both frames[1],

$$\mathbf{y}_{i,c_0} \sim \mathcal{N}\left(\bar{\mathbf{y}}_{i,c_0}, \mathbf{R}_{i,t}\right) = \mathcal{N}\left(\bar{\mathbf{y}}_{i,c_0}, \mathbf{R}(\boldsymbol{\phi}_{i,t})\right) \tag{A.1}$$

$$\mathbf{y}_{i,c_1}, \sim \mathcal{N}\left(\bar{\mathbf{y}}_{i,c_1}, \mathbf{R}_{i,t}\right) = \mathcal{N}\left(\bar{\mathbf{y}}_{i,c_1}, \mathbf{R}(\boldsymbol{\phi}_{i,t})\right). \tag{A.2}$$

This then builds the following weighted least squares objective,

$$\mathbf{T}_t^* = \underset{\mathbf{T}\in\mathrm{SE}(3)}{\operatorname{argmin}} \sum_{i=1}^{N_t} \mathbf{e}_i(\mathbf{T}_t)^T \boldsymbol{\Sigma}_{i,t}^{-1} \mathbf{e}_i(\mathbf{T}_t). \tag{A.3}$$

where $\boldsymbol{\Sigma}_{i,t}$ is now given by,

$$\boldsymbol{\Sigma}_{i,t} = \mathbf{D}\mathbf{G}_{i,c_1}\mathbf{R}(\boldsymbol{\phi}_{i,t})\mathbf{G}_{i,c_1}^T\mathbf{D}^T + \mathbf{D}\mathbf{T}_t\mathbf{G}_{i,c_0}\mathbf{R}(\boldsymbol{\phi}_{i,t})\mathbf{G}_{i,c_0}^T\mathbf{T}_t^T\mathbf{D}^T \tag{A.4}$$

[1]We conjecture that this is reasonable in a VO setup, where images change minimally between consecutive frames.

We build a model for $\mathbf{R}(\boldsymbol{\phi}_{i,t})$ as,

$$\mathbf{R}(\boldsymbol{\phi}_{i,c}) = \beta(\boldsymbol{\phi}_{i,t})\bar{\mathbf{R}}, \tag{A.5}$$

with

$$\beta(\boldsymbol{\phi}_{i,c}) = \left(\frac{1}{\epsilon_{\text{avg}} K} \sum_{k=1}^{K} \epsilon_k \right)^{\gamma}, \ \epsilon_k \in \text{K-NN}(\boldsymbol{\phi}_{i,t}), \tag{A.6}$$

where $\{\epsilon_k\}_{k=1}^{K}$ are K egomotion errors that are 'nearest' to $\boldsymbol{\phi}_{i,c}$, ϵ_{avg} is an average error, $\bar{\mathbf{R}}$ is a baseline *nominal* covariance, and $\gamma > 1$ is a hyper-parameter designed to exaggerate the effect of small changes in position error.

A.1.2 Training

Training proceeds by traversing the training path, selecting a subset of visual features at each step, and using them to compute an incremental position estimate. By comparing the estimated position to the ground truth position, we compute the translational Root Mean Squared Error (RMSE), ϵ, and store it at each feature's position in the prediction space. The full algorithm is summarized in Algorithm 4.

Algorithm 4 Train PROBE based on a dataset ($\mathcal{D}$) of pairs of input sensor data ($\mathcal{I}_s$) and ground truth egomotion ($\mathbf{T}_s$).

function BUILDPROBEMODEL($\mathcal{D}$)
  for $l \leftarrow [1, ..., N_{iter}]$ **do**
    for all $\mathcal{I}_s$, $\mathbf{T}_s$ in $\mathcal{D}$ **do**
      $\mathcal{F} \leftarrow$ EXTRACTFEATURES($\mathcal{I}_s$)
      $\{f_1, \ldots, f_J\} \leftarrow$ SAMPLE($\mathcal{F}$)
      $\hat{\mathbf{T}} \leftarrow$ COMPUTETRANSFORM($\{f_1, \ldots, f_J\}$)
      $\epsilon \leftarrow$ ERROR($\hat{\mathbf{T}}, \mathbf{T}_s$)
      $\{\boldsymbol{\phi}_{s,1}, \ldots, \boldsymbol{\phi}_{s,J}\} \leftarrow$ PREDICTOR($\{f_1, \ldots, f_J\}$)
      Insert $\{\boldsymbol{\phi}_{s,1}, \ldots, \boldsymbol{\phi}_{s,J}\}$ into $\mathcal{M}$ and store ϵ at all J locations
    end for
  end for
  return $\mathcal{M}$
end function

A.1.3 Evaluation

To use the PROBE model in a test environment, we compute the location of each observed visual feature in our prediction space, and then compute its relative weight β_i as a function

Figure A.1: Three types of environments in the KITTI dataset, as well as 2 types of environments at the University of Toronto. We use one trial from each category to train and then evaluate separate trials in the same category.

of its K nearest neighbours in the training set. For efficiency, the K nearest neighbours are found using a k-d tree. The final scaling factor β_i is a function of the mean of the α values corresponding to the K nearest neighbours, normalized by ϵ_{avg}, the mean α value of the entire training set.

Algorithm 5 Compute scalar covariance factors, β_i, for a set of stereo feature tracks (and IMU data), $\mathcal{F}$, given a PROBE model $\mathcal{M}$.

function USEPROBE($\mathcal{M}$, $\mathcal{F}$, γ)
  $\epsilon_{\text{avg}} \leftarrow$ AVERAGEERROR($\mathcal{M}$)
  for all f_i in $\mathcal{F}$ **do**
    $\boldsymbol{\phi}_i \leftarrow$ PREDICTOR(f_i)
    $\epsilon_1, ..., \epsilon_K \leftarrow$ FINDKNN($\boldsymbol{\phi}_i, K, \mathcal{M}$)
    $\beta_i \leftarrow \left(\frac{1}{\epsilon_{\text{avg}} K} \sum_{k=1}^{K} \epsilon_k \right)^{\gamma}$
  end for
  return $\boldsymbol{\beta} = \{\beta_i\}$
end function

The value of K can be determined through cross-validation, and in practice depends on the size of the training set and the environment. The computation of β_i is designed to map small differences in learned α values to scalar weights that span several orders of magnitude. An appropriate value of γ can be found by searching through a set range of candidate values and choosing the value that minimizes the average RMSE (ARMSE) on the training set.

A.2 Experiments

Table A.1: Comparison of translational Average Root Mean Square Error (ARMSE) and Final Translational Error on the KITTI dataset.

			Nominal RANSAC (99% outlier rejection)		Aggressive RANSAC (99.99% outlier rejection)		PROBE	
Trial	Type	Path Length	ARMSE	Final Error	ARMSE	Final Error	ARMSE	Final Error
26_drive_0051	City [1]	251.1 m	4.84 m	12.6 m	3.30 m	8.62 m	3.48 m	8.07 m
26_drive_0104	City [1]	245.1 m	0.977 m	4.43 m	0.850 m	3.46 m	1.19 m	3.61 m
29_drive_0071	City [1]	234.0 m	5.44 m	30.3 m	5.44 m	30.4 m	3.03 m	12.8 m
26_drive_0117	City [1]	322.5 m	2.29 m	9.07 m	2.29 m	9.07 m	2.76 m	9.08 m
30_drive_0027	Residential [1, †]	667.8 m	4.22 m	12.2 m	4.30 m	10.6 m	3.64 m	4.57 m
26_drive_0022	Residential [2]	515.3 m	2.21 m	3.99 m	2.66 m	6.09 m	3.06 m	4.99 m
26_drive_0023	Residential [2]	410.8 m	1.64 m	8.20 m	1.77 m	8.27 m	1.71 m	8.13 m
26_drive_0027	Road [3]	339.9 m	1.63 m	8.75 m	1.63 m	8.65 m	1.40 m	7.57 m
26_drive_0028	Road [3]	777.5 m	4.31 m	16.9 m	3.72 m	13.1 m	3.92 m	13.2 m
30_drive_0016	Road [3]	405.0 m	4.56 m	19.5 m	3.33 m	14.6 m	2.76 m	13.9 m
UTIAS Outdoor	Snowy parking lot	302.0 m	7.24 m	10.1 m	7.02 m	10.6 m	6.85 m	6.09 m
UTIAS Indoor	Lab interior	32.83 m	—	0.854 m	—	0.738 m	—	0.617 m

[1] Trained using sequence 09_26_drive_0005. [2] Trained using sequence 09_26_drive_0046. [3] Trained using sequence 09_26_drive_0015.
[†] This residential trial was evaluated with a model trained on a sequence from the city category because of several moving vehicles that were better represented in that training dataset.

Figure A.2: Our four-wheeled skid-steered Clearpath Husky rover equipped with Skybotix VI-Sensor and Ashtech DGPS antenna used to collect the outdoor UTIAS dataset.

Bibliography

Alcantarilla, P. F. and Woodford, O. J. (2016). Noise models in feature-based stereo visual odometry.

Altmann, S. L. (1989). Hamilton, rodrigues, and the quaternion scandal. *Math. Mag.*, 62(5):291–308.

Barfoot, T. D. (2017). *State Estimation for Robotics*. Cambridge University Press.

Barfoot, T. D. and Furgale, P. T. (2014a). Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Trans. Rob.*, 30(3):679–693.

Barfoot, T. D. and Furgale, P. T. (2014b). Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Trans. Robot.*, 30(3):679–693.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., and Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the Robust-Perception age. *IEEE Trans. Rob.*, 32(6):1309–1332.

Clement, L., Peretroukhin, V., and Kelly, J. (2017). Improving the accuracy of stereo visual odometry using visual illumination estimation. In Kulic, D., Nakamura, Y., Khatib, O., and Venture, G., editors, *2016 International Symposium on Experimental Robotics*, volume 1 of *Springer Proceedings in Advanced Robotics*, pages 409–419. Springer International Publishing, Berlin Heidelberg. Invited to Journal Special Issue.

Crete, F., Dolmiere, T., Ladret, P., and Nicolas, M. (2007). The blur effect: perception and estimation with a new no-reference perceptual blur metric. In *Human vision and electronic imaging XII*, volume 6492, page 64920I. International Society for Optics and Photonics.

Cvišić, I. and Petrović, I. (2015). Stereo odometry based on careful feature selection and tracking. In *Proc. European Conf. on Mobile Robots (ECMR)*, pages 1–6.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38.

DeTone, D., Malisiewicz, T., and Rabinovich, A. (2016). Deep image homography estimation.

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

Fitzgibbon, A. W., Robertson, D. P., Criminisi, A., Ramalingam, S., and Blake, A. (2007). Learning priors for calibrating families of stereo cameras. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pages 1–8.

Florez, S. A. R. (2010). *Contributions by vision systems to multi-sensor object localization and tracking for intelligent vehicles*. PhD thesis.

Forster, C., Pizzoli, M., and Scaramuzza, D. (2014a). SVO: Fast semi-direct monocular visual odometry. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 15–22.

Forster, C., Pizzoli, M., and Scaramuzza, D. (2014b). SVO: Fast semi-direct monocular visual odometry. In *Proc. IEEE Int. Conf. Robot. Automat.(ICRA)*, pages 15–22. IEEE.

Furgale, P. (2011). *Extensions to the Visual Odometry Pipeline for the Exploration of Planetary Surfaces*. PhD thesis.

Furgale, P., Rehder, J., and Siegwart, R. (2013). Unified temporal and spatial calibration for multi-sensor systems. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1280–1286.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proc. Int. Conf. Mach. Learning (ICML)*, pages 1050–1059.

Garg, R., Carneiro, G., and Reid, I. (2016). Unsupervised CNN for single view depth estimation: Geometry to the rescue. In *European Conf. on Comp. Vision*, pages 740–756. Springer.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013a). Vision meets robotics: The KITTI dataset. *Int. J. Rob. Res.*, 32(11):1231–1237.

Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013b). Vision meets robotics: The KITTI dataset. *Int. Journal Robot. Research (IJRR)*.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

Geiger, A., Ziegler, J., and Stiller, C. (2011a). StereoScan: Dense 3D reconstruction in real-time. In *Proc. Intelligent Vehicles Symp. (IV)*, pages 963–968. IEEE.

Geiger, A., Ziegler, J., and Stiller, C. (2011b). StereoScan: Dense 3D reconstruction in real-time. In *Proc. IEEE Intelligent Vehicles Symp. (IV)*, pages 963–968.

Geman, S., McClure, D. E., and Geman, D. (1992). A nonlinear filter for film restoration and other problems in image processing. *CVGIP: Graphical models and image processing*, 54(4):281–289.

Grewal, M. S. and Andrews, A. P. (2010). Applications of kalman filtering in aerospace 1960 to the present [historical perspectives]. *IEEE Control Syst. Mag.*, 30(3):69–78.

Handa, A., Bloesch, M., Pătrăucean, V., Stent, S., McCormac, J., and Davison, A. (2016). gvnn: Neural network library for geometric computer vision. In *Computer Vision – ECCV 2016 Workshops*, pages 67–82. Springer, Cham.

Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, pages 73–101.

Irani, M. and Anandan, P. (2000). About direct methods. In *Vision Algorithms: Theory and Practice*, pages 267–277. Springer.

Kendall, A. and Cipolla, R. (2016). Modelling uncertainty in deep learning for camera relocalization. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 4762–4769.

Kerl, C., Sturm, J., and Cremers, D. (2013). Robust odometry estimation for RGB-D cameras. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 3748–3754.

Lambert, A., Furgale, P., Barfoot, T. D., and Enright, J. (2012). Field testing of visual odometry aided by a sun sensor and inclinometer. *J. Field Robot.*, 29(3):426–444.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Leutenegger, S., Lynen, S., Bosse, M., Siegwart, R., and Furgale, P. (2015). Keyframe-based visual–inertial odometry using nonlinear optimization. *Int. J. Rob. Res.*, 34(3):314–334.

Lucas, B. D. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

MacTavish, K. and Barfoot, T. D. (2015). At all costs: A comparison of robust cost functions for camera correspondence outliers. In *Proc. Conf. on Comp. and Robot Vision (CRV)*, pages 62–69.

Maimone, M., Cheng, Y., and Matthies, L. (2007). Two years of visual odometry on the mars exploration rovers. *J. Field Robot.*, 24(3):169–186.

Mayor, A. (2019). *Gods and Robots*. Princeton University Press.

Melekhov, I., Ylioinas, J., Kannala, J., and Rahtu, E. (2017). Relative camera pose estimation using convolutional neural networks. In *Proc. Int. Conf. on Advanced Concepts for Intel. Vision Syst.*, pages 675–687. Springer.

Nilsson, N. J. (1984). Shakey the robot. Technical report, SRI INTERNATIONAL MENLO PARK CA.

Oliveira, G. L., Radwan, N., Burgard, W., and Brox, T. (2017). Topometric localization with deep learning.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. (2016). Deep exploration via bootstrapped DQN. In *Proc. Advances in Neural Inform. Process. Syst. (NIPS)*, pages 4026–4034.

Peretroukhin, V., Clement, L., Giamou, M., and Kelly, J. (2015a). PROBE: Predictive robust estimation for visual-inertial navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'15)*, pages 3668–3675, Hamburg, Germany.

Peretroukhin, V., Clement, L., and Kelly, J. (2015b). Get to the point: Active covariance scaling for feature tracking through motion blur. In *Proceedings of the IEEE International Conference on Robotics and Automation Workshop on Scaling Up Active Perception*, Seattle, Washington, USA.

Peretroukhin, V., Clement, L., and Kelly, J. (2017). Reducing drift in visual odometry by inferring sun direction using a bayesian convolutional neural network. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'17)*, pages 2035–2042, Singapore.

Peretroukhin, V., Clement, L., and Kelly, J. (2018). Inferring sun direction to improve visual odometry: A deep learning approach. *International Journal of Robotics Research*.

Peretroukhin, V. and Kelly, J. (2018). DPC-Net: Deep pose correction for visual localization. *IEEE Robotics and Automation Letters*.

Peretroukhin, V., Vega-Brown, W., Roy, N., and Kelly, J. (2016). PROBE-GK: Predictive robust estimation using generalized kernels. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 817–824.

Peretroukhin, V., Wagstaff, B., and Kelly, J. (2019). Deep probabilistic regression of elements of so(3) using quaternion averaging and uncertainty injection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'19) Workshop on Uncertainty and Robustness in Deep Visual Learning*, pages 83–86, Long Beach, California, USA.

Redfield, S. (2019). A definition for robotics as an academic discipline. *Nature Machine Intelligence*, 1(6):263–264.

Scaramuzza, D. and Fraundorfer, F. (2011a). Visual odometry [tutorial]. *IEEE Robot. Autom. Mag.*, 18(4):80–92.

Scaramuzza, D. and Fraundorfer, F. (2011b). Visual odometry [tutorial]. *IEEE Robot. Automat. Mag.*, 18(4):80–92.

Sola, J. (2017). Quaternion kinematics for the error-state kalman filter. *arXiv preprint arXiv:1711.02508*.

Solà, J., Deray, J., and Atchuthan, D. (2018). A micro lie theory for state estimation in robotics.

Sünderhauf, N. and Protzel, P. (2007). Stereo odometry: a review of approaches. *Chemnitz University of Technology Technical Report*.

Sünderhauf, N., Shirazi, S., Dayoub, F., Upcroft, B., and Milford, M. (2015). On the performance of ConvNet features for place recognition. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*, pages 4297–4304.

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. (2006). Stanley: The robot that won the DARPA grand challenge. *J. Field Robotics*, 23(9):661–692.

Tsotsos, K., Chiuso, A., and Soatto, S. (2015a). Robust inference for visual-inertial sensor fusion. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 5203–5210.

Tsotsos, K., Chiuso, A., and Soatto, S. (2015b). Robust inference for visual-inertial sensor fusion. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 5203–5210.

Umeyama, S. (1991). Least-Squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(4):376–380.

Vega-Brown, W., Bachrach, A., Bry, A., Kelly, J., and Roy, N. (2013). CELLO: A fast algorithm for covariance estimation. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 3160–3167.

Vega-Brown, W. and Roy, N. (2013). CELLO-EM: Adaptive sensor models without ground truth. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, pages 1907–1914.

Vega-Brown, W. R., Doniec, M., and Roy, N. G. (2014). Nonparametric Bayesian inference on multivariate exponential families. In *Proc. Advances in Neural Information Proc. Syst. (NIPS) 27*, pages 2546–2554.

Zhang, G. and Vela, P. (2015). Optimally observable and minimal cardinality monocular SLAM. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 5211–5218.

Zhou, B., Krähenbühl, P., and Koltun, V. (2019). Does computer vision matter for action?