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 *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 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. I hope my 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, 2011). Similarly, much of the planning and control algorithms originate in American and Soviet defence-funded research (Nilsson, 1984; Thrun et al., 2006).

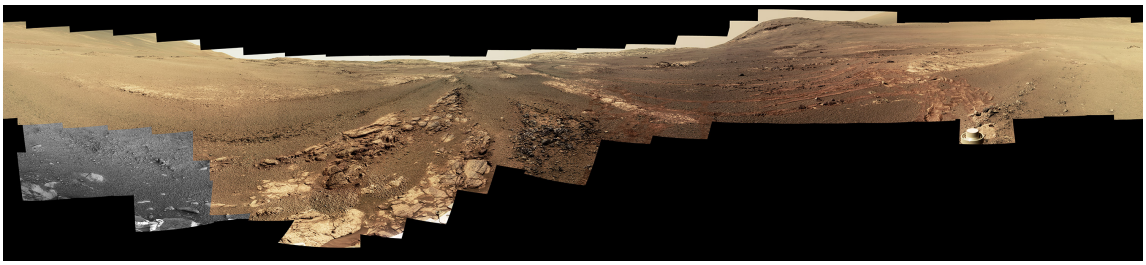
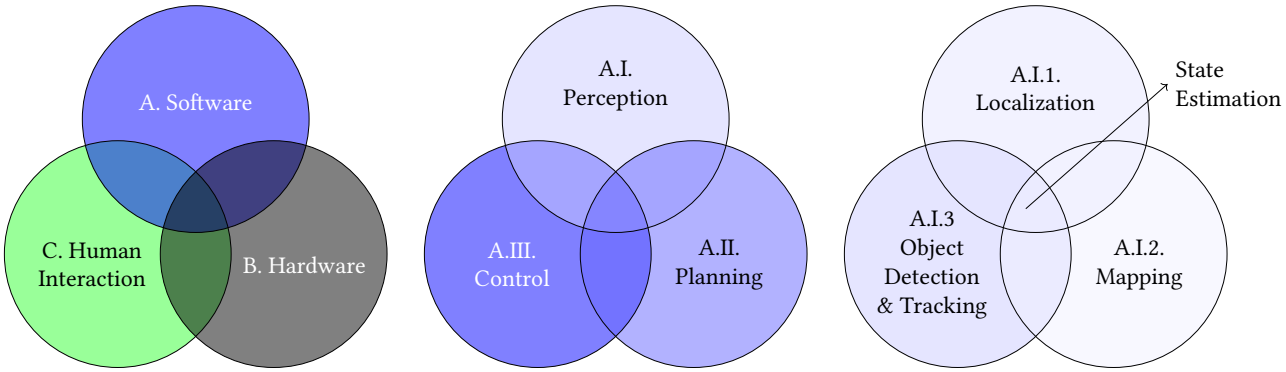


Figure 1.2: The last 360 degree panorama taken by the PanCam apparatus of the Mars Exploration Rover, *Opportunity*, at its final resting place on Mars, the western rim of the Endeavour Crater. Contact with *Opportunity* was lost shortly after this was captured, due to a severe dust storm (credit: NASA/JPL-Caltech/Cornell/ASU).



(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., 2013)).

Despite decades of research, mobile autonomy as a field still has nebulous demarcations between subfields. I 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

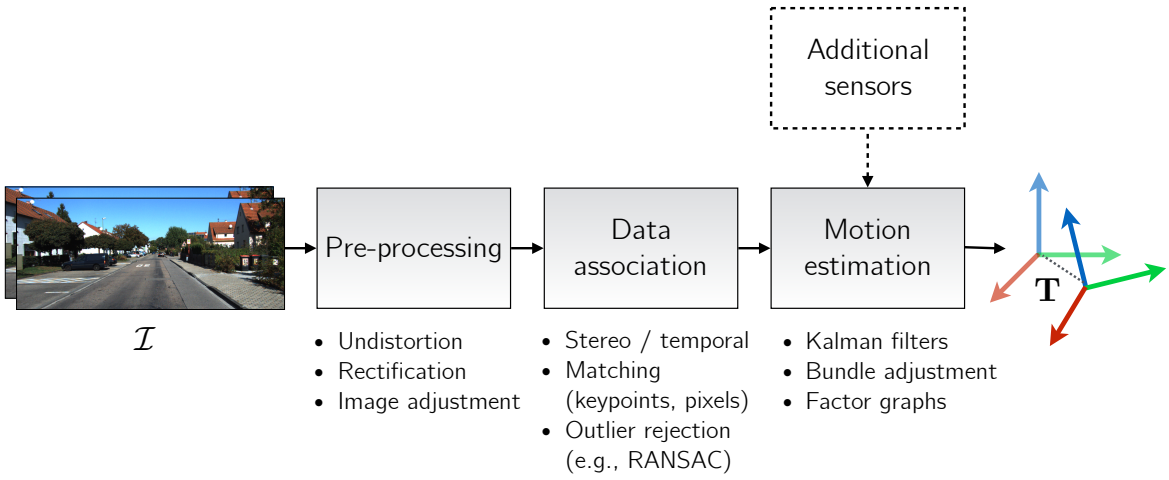


Figure 1.4: A 'classical' visual odometry pipeline consists of several distinct components that have interpretable inputs and outputs.

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 and 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).

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. A typical VO pipeline is illustrated in Figure 1.4.

Modern visual egomotion estimation pipelines (e.g., Leutenegger et al. (2015); Cvišić and Petrović (2015); Tsotsos et al. (2015)) 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., 2014), typically relying on loss functions that use direct pixel intensities. There are also those that fuse the two (Forster et al., 2014) modalities into a semi-direct approach.

In the last five years, a significant part of the 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 re-localization (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 existing 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 mini-

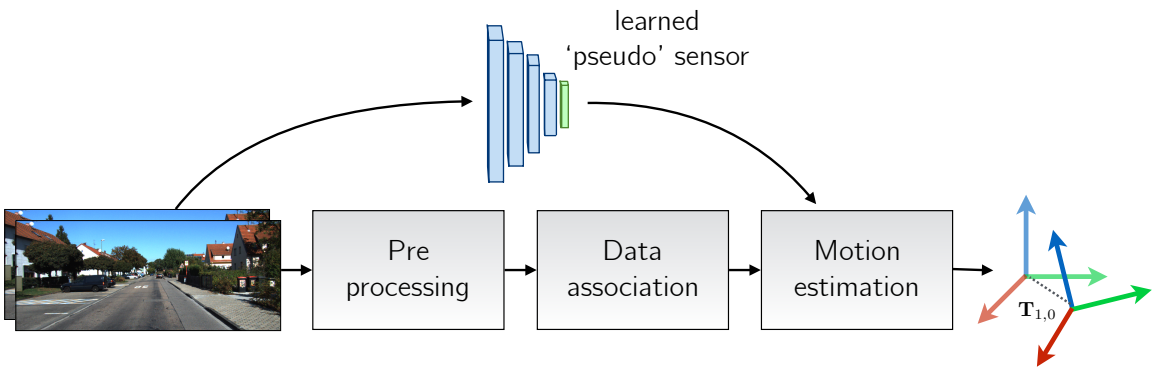


Figure 1.5: A learned *pseudo-sensor* extracts latent information from the same data stream.

mize 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. Furthermore , there is new evidence that separating different tasks into interpretable components (e.g., optical flow, scene segmentation) works better than end-to-end learning for action (Zhou et al., 2019). Table 1.1 summarizes the benefits of using deep models (as opposed to classical pipelines) along several salient criteria.

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., 2016b))
Robustness	Empirically generalizable (Zhou et al., 2019)	Highly dependant on training data
Flexibility	Limited by ingenuity of designer	Limited by training data

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, and to retain the benefits of classical state estimation pipelines while leveraging the representational power of modern data-drive learning techniques, we introduce the paradigm of the *learned pseudo-sensor*. Instead of completely replacing the classical pipeline, herein we present four ways in which machine learning can be used to train a hyper-parametric model 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, **??**), 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 baseline 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 (**??**) 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

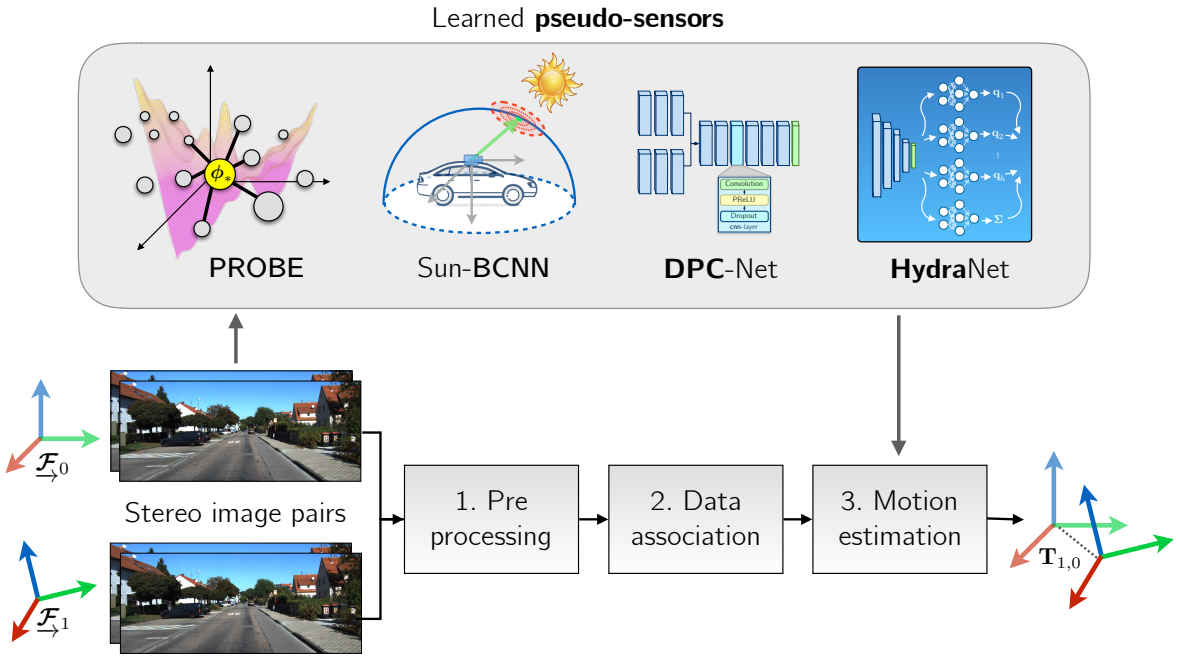


Figure 1.6: This dissertation details four examples of *pseudo-sensors* that improve 'classical' egomotion estimation through data-driven learning.

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. (2016a). 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 read-

ily 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.

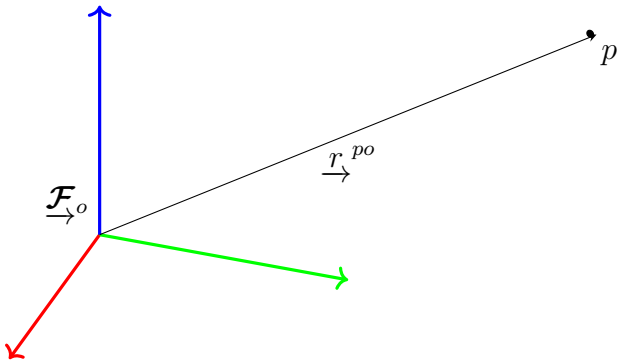


Figure 2.1: A position vector expressed in a coordinate frame.

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 $\mathbf{T}_{v,i}$, and $\mathbf{C}_{v,i}$[1], respectively.

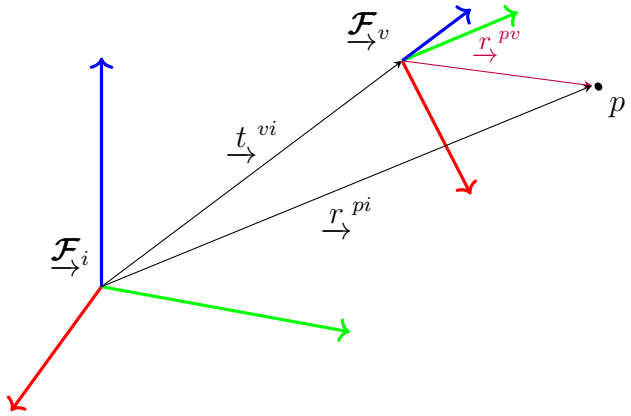


Figure 2.2: Two common references frames used throughout this thesis.

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

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

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

$$\begin{bmatrix} \mathbf{r}_v^{p,i} \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{C}_{v,i} & \mathbf{t}_v^{i,v} \\ \mathbf{0}^T & 1 \end{bmatrix}}_{\mathbf{T}_{v,i}} \begin{bmatrix} \mathbf{r}_i^{p,i} \\ 1 \end{bmatrix} \tag{2.17}$$

which is numerically equivalent to

$$\mathbf{r}_v^{p,i} = \mathbf{C}_{v,i}\mathbf{r}_i^{p,i} + \mathbf{t}_v^{i,v} \tag{2.18}$$

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 (2014) 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, $\boldsymbol{\Sigma}$.

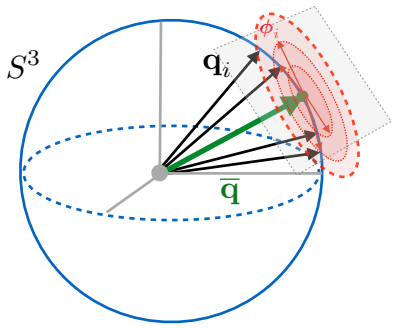


Figure 2.3: We can define uncertainty in the left tangent space of a mean element of a Lie group (here illustrated for unit quaternions).

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

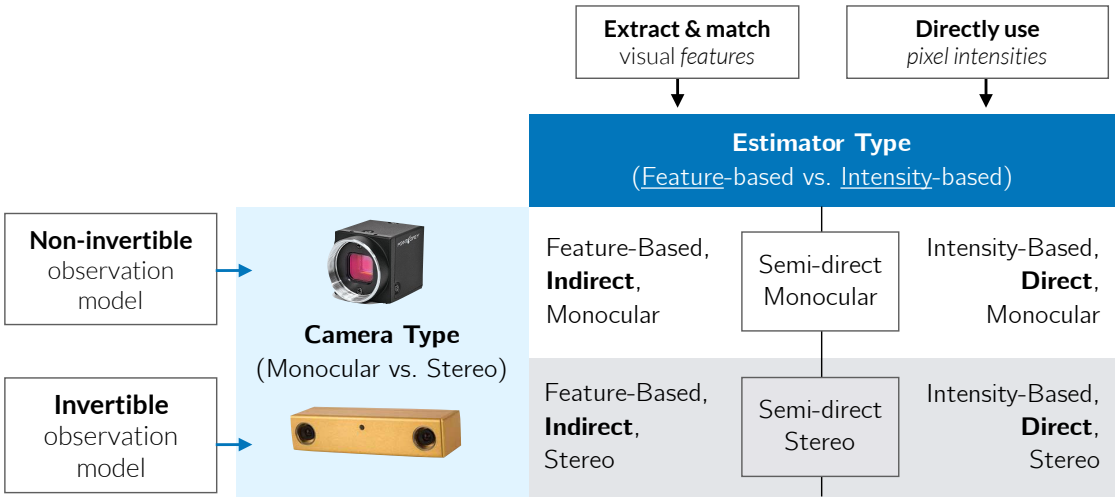


Figure 3.1: A taxonomy of different types of visual odometry.

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 (2011) 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.

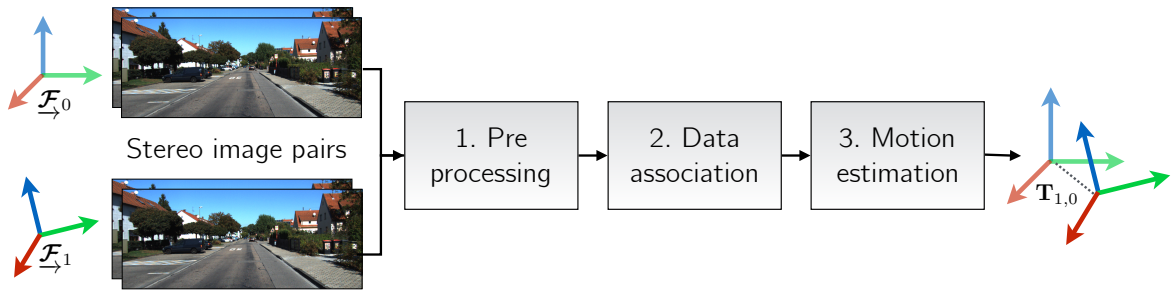
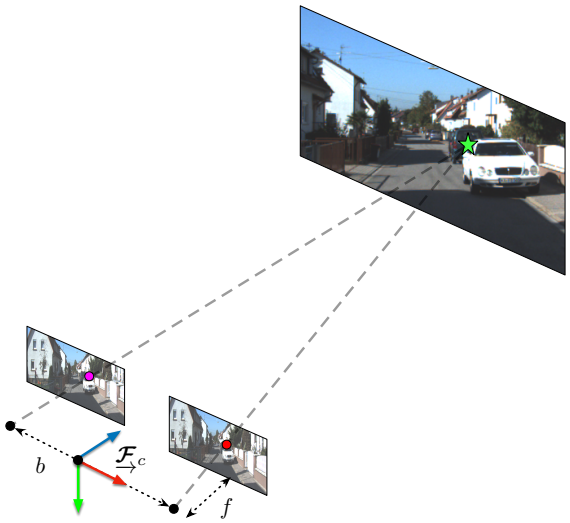
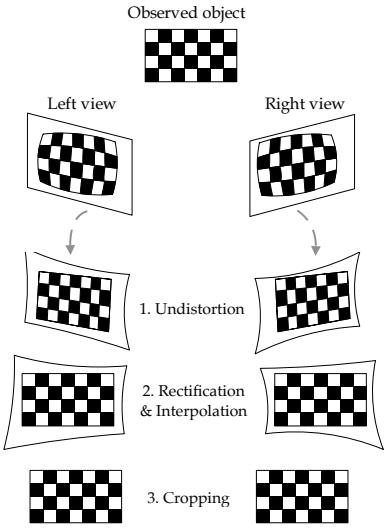


Figure 3.2: A 'classical' stereo visual odometry pipeline consists of several distinct components that have interpretable inputs and outputs.

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

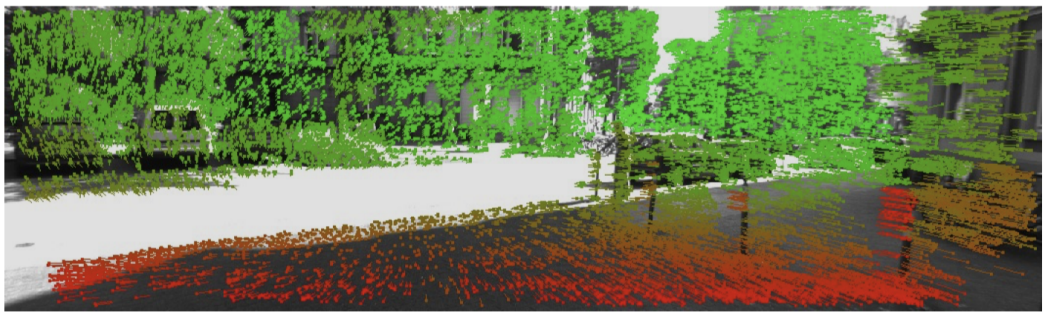


Figure 3.4: Feature tracking using `libviso2`, taken from Geiger et al. (2011a). Colours correspond to depth.

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, 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

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

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 Motion solution

To compute $\mathbf{T} \in \mathrm{SE}(3)$, the rigid transform between two subsequent stereo camera poses, we assume we have a set of N_t visual landmarks in each stereo pair at a time instance, t.

We triangulate landmarks in the first camera frame, $\mathbf{y}_{i,t}$, and re-project them into the second frame, $\mathbf{y}'_{i,t}$. We model errors due to sensor noise and quantization as a Gaussian distribution in image space with a covariance $\mathbf{\Sigma}_{\mathbf{y}_{i,t}}$ that may change for each feature or may be constant, $\mathbf{\Sigma}_{\mathbf{y}_{i,t}} = \mathbf{\Sigma}_{\mathbf{y}}$,

$$p(\mathbf{y}'_{i,t}|\mathbf{y}_{i,t}, \mathbf{T}, \mathbf{\Sigma}_{\mathbf{y}_{i,t}}) = \mathcal{N}\left(\mathbf{e}_{i,t}(\mathbf{T}); \mathbf{0}, \mathbf{\Sigma}_{\mathbf{y}_{i,t}}\right), \tag{3.3}$$

where

$$\mathbf{e}_{i,t} = \mathbf{y}'_{i,t} - f(\mathbf{T}f^{-1}(\mathbf{y}_{i,t})). \tag{3.4}$$

The maximum likelihood transform, $\mathbf{T}^*$, is then given by

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

This is a nonlinear least squares problem, and can be solved iteratively using standard techniques. During iteration n, we represent the transform as the product of an estimate $\mathbf{T}^{(n)} \in \mathrm{SE}(3)$ and a perturbation $\delta\boldsymbol{\xi} \in \mathbb{R}^6$ represented in exponential coordinates:

$$\mathbf{T} = \exp\left(\delta\boldsymbol{\xi}^{\wedge}\right)\overline{\mathbf{T}}. \tag{3.6}$$

Linearizing the transform for small perturbations $\delta\boldsymbol{\xi}$ yields a linear least-squares problem:

$$\mathcal{L}(\delta\boldsymbol{\xi}) = \frac{1}{2}\sum_{i=1}^{N_t} \left(\mathbf{e}_{i,t}^{(n)} - \mathbf{J}_{i,t}^{(n)}\delta\boldsymbol{\xi}\right)^T \mathbf{\Sigma}_{\mathbf{y}_{i,t}}^{-1} \left(\mathbf{e}_{i,t}^{(n)} - \mathbf{J}_{i,t}^{(n)}\delta\boldsymbol{\xi}\right) \tag{3.7}$$

Here, $\mathbf{J}_{i,t}^{(n)}$ is the Jacobian matrix of the reprojection error. **ToDo: add this to the appendix**

Rearranging, we see the minimizing perturbation is the solution to a linear system of equations:

$$\delta\boldsymbol{\xi}^{(n)} = \left(\sum_{i=1}^{N_t} \mathbf{J}_{i,t}^T \mathbf{R}_{,}^{-1} \mathbf{J}_{i,t}\right)^{-1} \sum_{i=1}^{N_t} \mathbf{J}_{i,t}^T \mathbf{\Sigma}_{\mathbf{y}_{i,t}}^{-1} \mathbf{e}_{i,t}^{(n)}. \tag{3.8}$$

We then update the estimated transform and proceed to the next iteration,

$$\mathbf{T}^{(n+1)} = \text{Exp}\left(\delta\boldsymbol{\xi}^{(n)}\right)\mathbf{T}^{(n)}. \tag{3.9}$$

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. (3.9) until we see a relative change in the squared error of less than one percent after an update.

3.3 Robust Estimation

Since eq. (3.7) 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, $\rho(\cdot)$,

$$\mathbf{T}^* = \underset{\mathbf{T}\in\text{SE}(3)}{\text{argmin}} \sum_{i=1}^{N_t} \rho\left(\mathbf{e}_{i,t}{}^T \boldsymbol{\Sigma}_{\mathbf{y}_{i,t}}^{-1}\mathbf{e}_{i,t}\right) = \underset{\mathbf{T}\in\text{SE}(3)}{\text{argmin}} \sum_{i=1}^{N_t} \rho(\epsilon_i), \tag{3.10}$$

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

$$\rho(\epsilon) = \begin{cases} \frac{c^2}{2}\log\left(1+\frac{\epsilon^2}{c^2}\right) & \text{Cauchy,} \\ \frac{1}{2}\frac{\epsilon^2}{c^2+\epsilon^2} & \text{Geman-McClure (Geman et al., 1992),} \\ \begin{cases} \frac{\epsilon^2}{2} & \text{if } \|\epsilon\| < c \\ c\|\epsilon\| - \frac{c^2}{2} & \text{if } \|\epsilon\| \geq c \end{cases} & \text{Huber (Huber, 1964).} \end{cases} \tag{3.11}$$

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

Conclusion

> Of what a strange nature is knowledge! It clings to a mind when it has once seized on it like a lichen on a rock.
>
> — Mary Shelley, *Frankenstein; or, The Modern Prometheus*

And thus, we have reached the end. This dissertation has dealt with the development of a general framework for improving the performance of model-based visual odometry pipelines through learned probabilistic pseudo-sensors that extract difficult-to-model latent information. We presented four examples of such *pseudo-sensors*. We close with a final summary of the contributions and list of publications, a discussion of potential future work, and some concluding remarks.

4.1 Summary of Contributions

4.1.1 Predictive Robust Estimation

We began with a pseudo-sensor that used a heteroscedastic noise model to enable predictively robust estimation. PROBE and its follow-up work, PROBE-GK, contributed

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

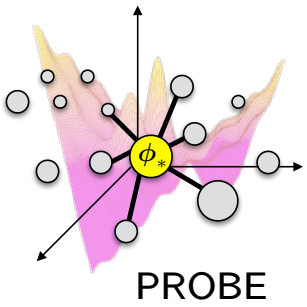


Figure 4.1: PROBE (??).

2. two different approaches to constructing the robust algorithm: one baseds on k-nearest neighbours, and one 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.

A total of three publications associated with PROBE,

- Peretroukhin, V., Vega-Brown, W., Roy, N., and Kelly, J. (2016b). PROBE-GK: Predictive robust estimation using generalized kernels. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'16)*, pages 817–824, Stockholm, Sweden

- 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.

4.1.2 Sun BCNN

With Sun-BCNN, we applied learned *pseudo-sensors* to the problem of illumination direction in outdoor environments. In sum, the novel contributions were:

1. the application of a Bayesian CNN to the problem of sun direction estimation, incorporating the resulting covariance estimates into a visual odometry pipeline;

2. an empirical demonstration that a Bayesian CNN with dropout layers after each convolutional and fully-connected layer can achieve state-of-the-art accuracy at test time;

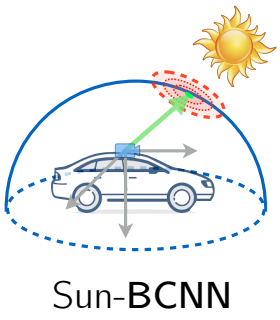


Figure 4.2: Sun BCNN (**??**).

3. a loss function that incorporated a 3D unit-length sun direction vector, appropriate for full 6-DOF pose estimation;

4. experimental results on over 30 km of visual navigation data in urban (Geiger et al., 2012) and planetary analogue (Furgale et al., 2012) environments;

5. an investigation into the sensitivity of the Bayesian CNN-based sun estimate to cloud cover, camera and environment changes, and measurement parameterization; and

6. open-source software[1].

Sun-BCNN and its origin in learned sun sensors have three associated publications,

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

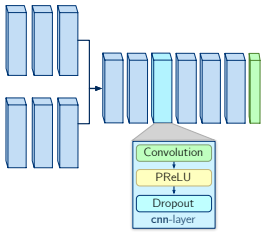
- Peretroukhin, V., Clement, L., and Kelly, J. (2017). Reducing drift in visual odometry by inferring sun direction using a bayesian convolutional neural network. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'17)*, pages 2035–2042, Singapore

- 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.

4.1.3 Deep Pose Corrections

Next, we generalized the results of Sun-BCNN to learn full six degree-of-freedom corrections for a particular egomotion pipeline and a given environment with DPC-Net. Our contributions included

1. the formulation of a novel deep corrective approach to egomotion estimation,



DPC-Net

Figure 4.3: DPC-Net (**??**).

[1] https://github.com/utiasSTARS/sun-bcnn-vo.

2. a novel cost function for deep $\mathrm{SE}(3)$ regression that naturally balances translation and rotation errors, and

3. an open-source implementation of DPC-Net in `PyTorch`[2].

DPC-Net was published in a journal publication,

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

4.1.4 Deep Probabilistic Inference of $\mathrm{SO}(3)$ with HydraNet

Finally, we applied the lessons of DPC-Net and Sun-BCNN to learning only rotation estimates through a network structure that incorporated both aleatoric and epistemic uncertainty which can be fused with classical pipelines through pose graph optimization. With this work, we contributed

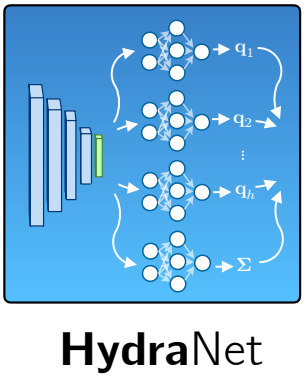


Figure 4.4: HydraNet (**??**).

1. a deep network structure we call *HydraNet* that built on prior work Lakshminarayanan et al. (2017); Osband et al. (2016a) to produce meaningful uncertainties over unconstrained targets,

2. a loss formulation and mathematical framework that extends HydraNet to means and covariances of the rotation group $\mathrm{SO}(3)$,

3. and open source code for $\mathrm{SO}(3)$ regression[3].

HydraNet was published in a refereed workshop paper,

- 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.

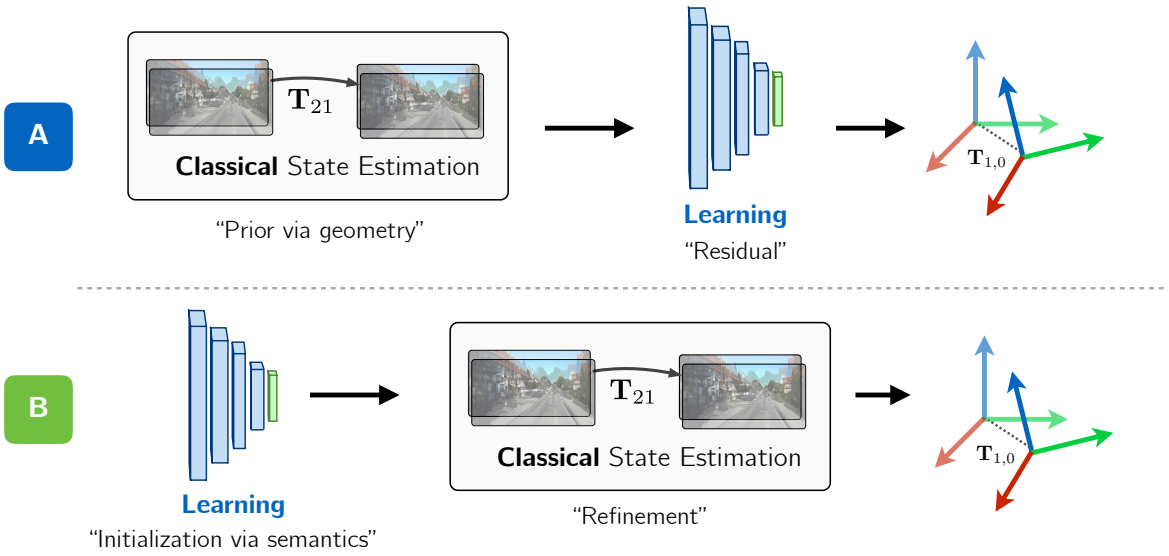


Figure 4.5: Two different ways to incorporate learning with classical pipelines. The first is by using pipelines as a *prior* which can then be corrected by learned approaches, while the second is use learning as an initialization which can then be *refined* by classical techniques.

4.2 Future Work

There are many avenues for future work. For instance, although the fusion of pipelines with pseudo-sensors can significantly improve localization performance in a given environment, there are few guarantees that the final estimates are accurate and consistent.

A potential thread of future work addresses this deficiency by developing more tightly-coupled perception systems that can be 'certified' to produce globally-optimal solutions while still being robust to adverse environmental effects. To do this, we propose that another way to fuse learned models with classical optimization techniques. Instead of treating the learning as a way to incorporate residuals (as much of this thesis is), we propose that we use learning as initializations, or priors, that are more invariant to effects like large view-point changes and have the ability to generalize to large-scale environments (Figure 4.5). By incorporating these learned 'priors' into an optimization framework, we can leverage recently-developed theory on convex relaxations (e.g., that presented in Rosen et al. (2019)) to ensure that the final localization and mapping results are 'optimal', in the sense that they are the global minima of a maximum-likelihood-based loss, and 'safe', in the sense that they provide consistent uncertainty estimates even in the presence of unmodelled effects.

[2]See https://github.com/utiasSTARS/dpc-net.
[3]https://github.com/utiasSTARS/so3_learning.

4.3 Final Remarks

The German Philosopher Georg Hegel had a particular dialectic method that involved a triad: a thesis, antithesis and synthesis. Somewhat curiously, this *thesis* has been an attempt at a *synthesis* of the thesis posed by classical visual egomotion, and the antithesis posed by data-driven end-to-end learning techniques. My hope is that the synthesis proposed by the paradigm of learned *pseudo-sensors* will prove to be a fruitful one within the field of state-estimation, and for the broader robotics community at large.

Post Scriptum

> Machinery that gives us abundance has left us in want. Our knowledge has made us cynical. Our cleverness, hard and unkind. We think too much, and feel too little. More than machinery, we need humanity.
>
> Charlie Chaplin, *The Great Dictator*

Appendices

Appendix A

Left and Middle Perturbations

A.1 Identities

$$\operatorname{Exp}\left(\left(\boldsymbol{\xi}+\delta\boldsymbol{\xi}\right)\right) \approx \operatorname{Exp}\left(\left(\boldsymbol{\mathcal{J}}\delta\boldsymbol{\xi}\right)\right)\operatorname{Exp}\left(\boldsymbol{\xi}\right), \tag{A.1}$$

$$\begin{aligned}\log\left(\mathbf{T}_1\mathbf{T}_2\right)^{\vee} &= \log\left(\operatorname{Exp}\left(\boldsymbol{\xi}_1\right)\operatorname{Exp}\left(\boldsymbol{\xi}_2\right)\right)^{\vee} \\ &\approx \begin{cases} \boldsymbol{\mathcal{J}}(\boldsymbol{\xi}_2)^{-1}\boldsymbol{\xi}_1+\boldsymbol{\xi}_2 & \text{if } \boldsymbol{\xi}_1 \text{ small} \\ \boldsymbol{\xi}_1+\boldsymbol{\mathcal{J}}(-\boldsymbol{\xi}_1)^{-1}\boldsymbol{\xi}_2 & \text{if } \boldsymbol{\xi}_2 \text{ small.} \end{cases}\end{aligned} \tag{A.2}$$

A.2 Perturbing $\mathrm{SE}(3)$

Consider the quantity,

$$\mathbf{T}_{ba} = \mathbf{T}_{bi}{\mathbf{T}_{ai}}^{-1} \tag{A.3}$$

where $\underrightarrow{\mathcal{F}}_i$ is some inertial frame.

A.2.1 Left Perturbation

Separating $\mathbf{T}_{ba}$ into a mean component, $\overline{\mathbf{T}}_{ba}$, and a small left perturbation,

$$\mathbf{T}_{ba} = \operatorname{Exp}\left(\delta\boldsymbol{\xi}^l_{ba}\right)\overline{\mathbf{T}}_{ba} = \operatorname{Exp}\left(\delta\boldsymbol{\xi}^l_{ba}\right)\operatorname{Exp}\left(\overline{\boldsymbol{\xi}}_{ba}\right) \tag{A.4}$$

Applying a logarithm to both sides,

$$\log\left(\mathbf{T}_{ba}\right)^{\vee} = \log\left(\operatorname{Exp}\left(\delta\boldsymbol{\xi}^l_{ba}\right)\operatorname{Exp}\left(\overline{\boldsymbol{\xi}}_{ba}\right)\right)^{\vee} \tag{A.5}$$

Using Equation (A.2),

$$\log\left(\mathbf{T}_{ba}\right)^{\vee} \approx \overline{\boldsymbol{\xi}}_{ba} + \boldsymbol{\mathcal{J}}_{ba}^{-1}\delta\boldsymbol{\xi}_{ba}^{l} \tag{A.6}$$

where $\boldsymbol{\mathcal{J}}_{ba} \triangleq \boldsymbol{\mathcal{J}}(\overline{\boldsymbol{\xi}}_{ba})$. This is exactly Equation 6.104 in Sean Anderson's thesis.

A.2.2 Middle Perturbation

Now consider the middle perturbation,

$$\mathbf{T}_{ba} = \operatorname{Exp}\left(\overline{\boldsymbol{\xi}}_{ba} + \delta\boldsymbol{\xi}_{ba}^{m}\right) \tag{A.7}$$

Immediately, we can take the logarithm of both sides and see that,

$$\log\left(\mathbf{T}_{ba}\right)^{\vee} = \overline{\boldsymbol{\xi}}_{ba} + \delta\boldsymbol{\xi}_{ba}^{m}, \tag{A.8}$$

where we now observe that $\delta\boldsymbol{\xi}_{ba}^{l} \approx \boldsymbol{\mathcal{J}}_{ba}\delta\boldsymbol{\xi}_{ba}^{m}$.

A.3 DPC $\mathrm{SE}(3)$ Loss

Using the notation in this document, the DPC derivation requires an expression for $\delta\boldsymbol{\xi}_b$, and assumes that $\boldsymbol{\xi}_a$ is constant.

A.3.1 Middle Perturbation

Consider,

$$\mathbf{T}_{ba} = \mathbf{T}_b{\mathbf{T}_a}^{-1} \tag{A.9}$$

where we have dropped the i frame for clarity. Middle perturbing $\mathbf{T}_{ba}$ and $\mathbf{T}_b$ and keeping $\mathbf{T}_a$ fixed (i.e., $\mathbf{T}_a = \overline{\mathbf{T}}_a$).

$$\operatorname{Exp}\left(\overline{\boldsymbol{\xi}}_{ba} + \delta\boldsymbol{\xi}_{ba}^{m}\right) = \operatorname{Exp}\left(\overline{\boldsymbol{\xi}}_{b} + \delta\boldsymbol{\xi}_{b}^{m}\right)\overline{\mathbf{T}}_a^{-1} \tag{A.10}$$

Using Equation (A.1) twice,

$$\operatorname{Exp}\left(\left(\boldsymbol{\mathcal{J}}_{ba}\delta\boldsymbol{\xi}_{ba}^{m}\right)\right)\operatorname{Exp}\left(\overline{\boldsymbol{\xi}}_{ba}\right) = \operatorname{Exp}\left(\left(\boldsymbol{\mathcal{J}}_{b}\delta\boldsymbol{\xi}_{b}^{m}\right)\right)\operatorname{Exp}\left(\overline{\boldsymbol{\xi}}_{b}\right)\overline{\mathbf{T}}_a^{-1} \tag{A.11}$$

Collecting terms, we have

$$\operatorname{Exp}\left(\left(\boldsymbol{\mathcal{J}}_{ba}\delta\boldsymbol{\xi}_{ba}^{m}\right)\right)\overline{\mathbf{T}}_{ba}=\operatorname{Exp}\left(\left(\boldsymbol{\mathcal{J}}_{b}\delta\boldsymbol{\xi}_{b}^{m}\right)\right)\overline{\mathbf{T}}_{b}\overline{\mathbf{T}}_{a}^{-1} \tag{A.12}$$

Right multiplying by $\overline{\mathbf{T}}_{ba}^{-1}$, we are left with

$$\operatorname{Exp}\left(\left(\boldsymbol{\mathcal{J}}_{ba}\delta\boldsymbol{\xi}_{ba}^{m}\right)\right)=\operatorname{Exp}\left(\left(\boldsymbol{\mathcal{J}}_{b}\delta\boldsymbol{\xi}_{b}^{m}\right)\right) \tag{A.13}$$

or

$$\boldsymbol{\mathcal{J}}_{ba}\delta\boldsymbol{\xi}_{ba}^{m}=\boldsymbol{\mathcal{J}}_{b}\delta\boldsymbol{\xi}_{b}^{m}. \tag{A.14}$$

Solving for $\delta\boldsymbol{\xi}_{ba}^{m}$,

$$\delta\boldsymbol{\xi}_{ba}^{m}=\boldsymbol{\mathcal{J}}_{ba}^{-1}\boldsymbol{\mathcal{J}}_{b}\delta\boldsymbol{\xi}_{b}^{m}. \tag{A.15}$$

Now inserting Equation (A.15) into Equation (A.8),

$$\log\left(\mathbf{T}_{ba}\right)^{\vee}\approx\overline{\boldsymbol{\xi}}_{ba}+\boldsymbol{\mathcal{J}}_{ba}^{-1}\boldsymbol{\mathcal{J}}_{b}\delta\boldsymbol{\xi}_{b}^{m} \tag{A.16}$$

This is exactly Equation 13 in the DPC-Net paper:

$$g(\boldsymbol{\xi}+\delta\boldsymbol{\xi})\approx\boldsymbol{\mathcal{J}}(g(\boldsymbol{\xi}))^{-1}\boldsymbol{\mathcal{J}}(\boldsymbol{\xi})\delta\boldsymbol{\xi}+g(\boldsymbol{\xi}). \tag{A.17}$$

A.3.2 Left Perturbation

Using the left perturbation, we can repeat the procedure of relating $\delta\boldsymbol{\xi}_{ba}^{l}$ and $\delta\boldsymbol{\xi}_{b}^{l}$ (by perturbing $\mathbf{T}_{ba}$ and $\mathbf{T}_{b}$ and keeping $\mathbf{T}_{a}$ fixed (i.e., $\mathbf{T}_{a}=\overline{\mathbf{T}}_{a}$).

$$\operatorname{Exp}\left(\delta\boldsymbol{\xi}_{ba}^{l}\right)\overline{\mathbf{T}}_{ba}=\operatorname{Exp}\left(\delta\boldsymbol{\xi}_{b}^{l}\right)\overline{\mathbf{T}}_{b}\overline{\mathbf{T}}_{a}^{-1} \tag{A.18}$$

from which we see immediately that $\operatorname{Exp}\left(\delta\boldsymbol{\xi}_{ba}^{l}\right)=\operatorname{Exp}\left(\delta\boldsymbol{\xi}_{b}^{l}\right)$ and therefore,

$$\delta\boldsymbol{\xi}_{ba}^{l}=\delta\boldsymbol{\xi}_{b}^{l} \tag{A.19}$$

Now using Equation (A.6), we have

$$\log\left(\mathbf{T}_{ba}\right)^{\vee}\approx\overline{\boldsymbol{\xi}}_{ba}+\boldsymbol{\mathcal{J}}_{ba}^{-1}\delta\boldsymbol{\xi}_{b}^{l} \tag{A.20}$$

A.3.3 Summary

Using the left perturbation, we have

$$\log\left(\mathbf{T}_{ba}\right)^{\vee} \approx \overline{\boldsymbol{\xi}}_{ba} + \boldsymbol{\mathcal{J}}_{ba}^{-1}\delta\boldsymbol{\xi}_{b}^{l} \tag{A.21}$$

Using the centre/middle perturbation, we have

$$\log\left(\mathbf{T}_{ba}\right)^{\vee} \approx \overline{\boldsymbol{\xi}}_{ba} + \boldsymbol{\mathcal{J}}_{ba}^{-1}\boldsymbol{\mathcal{J}}_{b}\delta\boldsymbol{\xi}_{b}^{m} \tag{A.22}$$

And we see the same earlier expression relating left and middle perturbations,

$$\delta\boldsymbol{\xi}^{l} \approx \boldsymbol{\mathcal{J}}\delta\boldsymbol{\xi}^{m} \tag{A.23}$$

A.3.4 Reconciliation

Consider the two update rules:

$$\mathbf{T}_{b} \leftarrow \operatorname{Exp}\left(\delta\boldsymbol{\xi}_{b}^{l}\right)\overline{\mathbf{T}}_{b} \tag{A.24}$$

$$\mathbf{T}_{b} \leftarrow \operatorname{Exp}\left((\overline{\boldsymbol{\xi}}_{b} + \delta\boldsymbol{\xi}_{b}^{m})\right) \tag{A.25}$$

But using Equation (A.1), the middle update becomes,

$$\operatorname{Exp}\left((\overline{\boldsymbol{\xi}}_{b} + \delta\boldsymbol{\xi}_{b}^{m})\right) \approx \operatorname{Exp}\left((\boldsymbol{\mathcal{J}}_{b}\delta\boldsymbol{\xi}_{b}^{m})\right)\operatorname{Exp}\left(\overline{\boldsymbol{\xi}}_{b}\right) = \operatorname{Exp}\left((\boldsymbol{\mathcal{J}}_{b}\delta\boldsymbol{\xi}_{b}^{m})\right)\overline{\mathbf{T}}_{b} = \operatorname{Exp}\left(\delta\boldsymbol{\xi}_{b}^{l}\right)\overline{\mathbf{T}}_{b} \tag{A.26}$$

So the middle perturbation does not require us to keep the mean in the group (as long as we avoid any degeneracies).

Appendix B

Supplementary HydraNet Details

B.1 Experiments

B.1.1 One-dimensional regression

For each uncertainty extraction, we used a four layer neural network (with 20 units per layer) with a Scaled Exponential Linear Unit (SELU). For the dropout method, we added dropout layers (with a small dropout probability, $p = 0.03$, to account for the small network size as recommended by Gal and Ghahramani (2016)). We performed 50 forward passes through the network, and computed the mean and variance of the outputs to determine the prediction and uncertainty estimate. For the ensemble bootstrap method, we trained ten separate models on bootstrapped samples of the training data. For HydraNet, we used the first two layers as the body, and branched the final two layers into ten heads. One additional head was created that directly regressed an uncertainty estimate.

Every model in this experiment was trained for 3000 epochs using stochastic gradient descent with momentum, using minibatch sizes of 50 (refer to Table B.1 for specific hyperparameters). We repeated training 100 times, and recorded the test-time negative log likelihood for each method at each repetition.

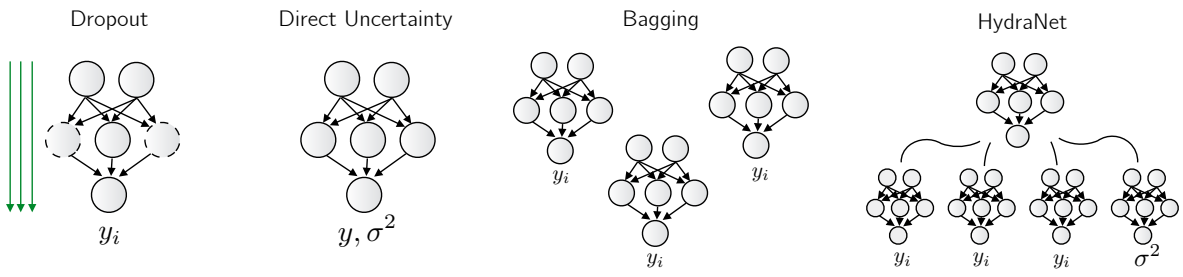


Figure B.1: Different scalable approaches to neural network uncertainty.

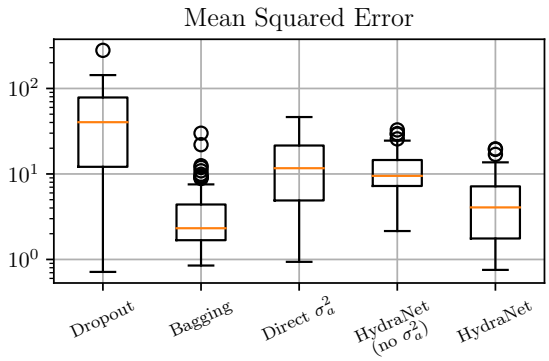


Figure B.2: Mean squared errors for the different probabilistic regression models in 1D.

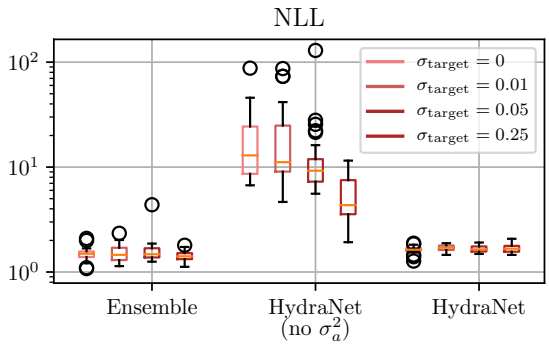
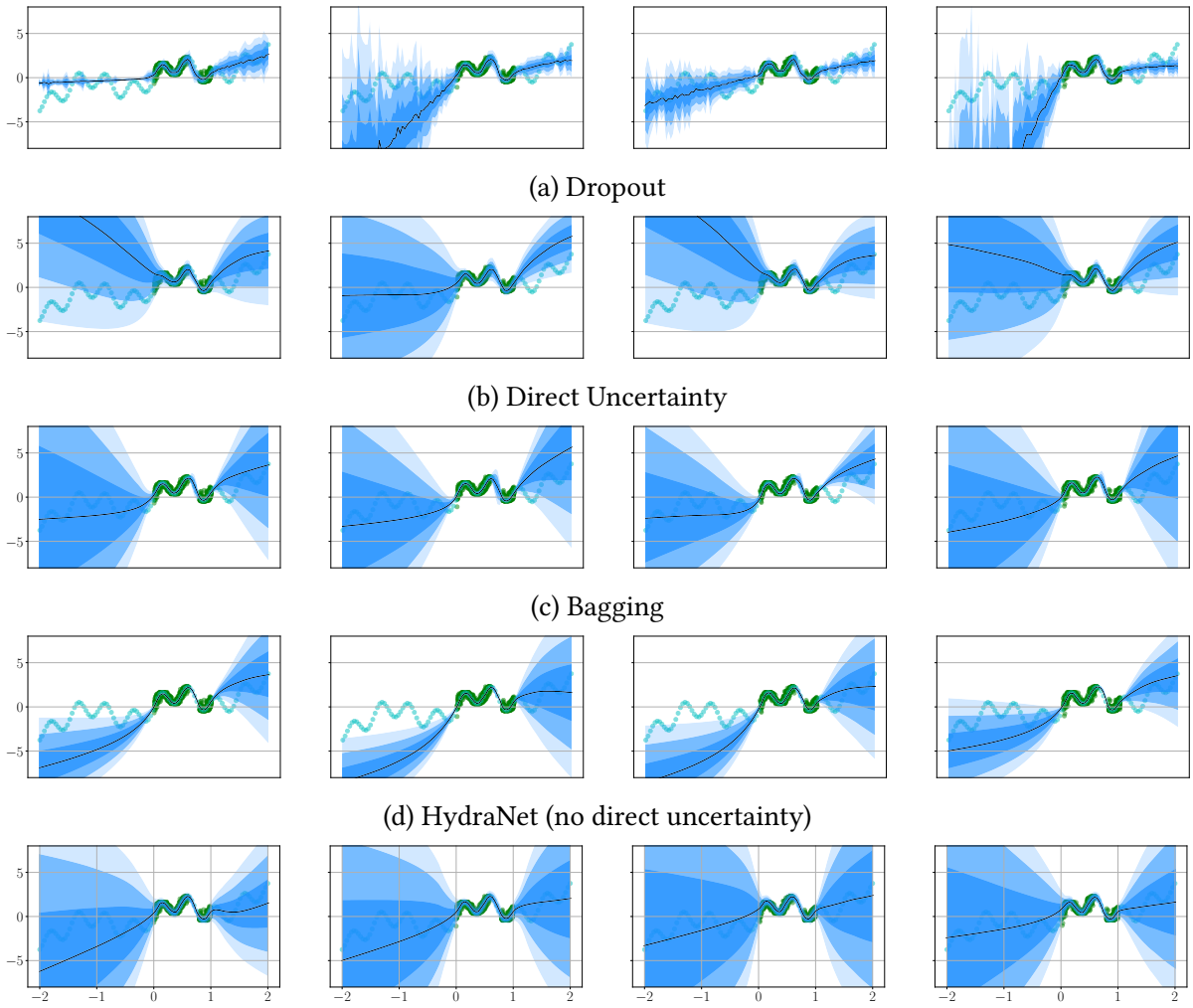


Figure B.3: For the 1D experiment, we experimented with adding zero-mean Gaussian additive noise to the regression targets in an attempt to promote diversity amongst the outputs. We found that while this improved the uncertainty estimates gleaned from the HydraNet heads alone (what we call epistemic uncertainty) it made little difference once we included aleatoric uncertainty.

We present three additional figures here that were not included in the main paper. Figure B.4 presents four representative samples from the 100 repetitions for each method, and Figure B.2 presents mean squared errors for each method. The last figure, Figure B.3, details the effects of adding zero mean Gaussian noise to the regression targets during training. We experimented with this approach to try and promote more diversity amongst the HydraNet heads within training data. We found, however, that although this does improve the negative log likelihoods for HydraNet with only epistemic uncertainty (i.e., the sample variance over the head outputs), its benefits were non-existent for the full HydraNet approach. Namely, since the full HydraNet approach uses an NLL loss, the network tended to account for target noise by enlarging the aleatoric uncertainty rather than overfitting each head to a specific target.



(a) Dropout

(b) Direct Uncertainty

(c) Bagging

(d) HydraNet (no direct uncertainty)

(e) HydraNet

Figure B.4: A comparison of different ways to extract uncertainty from deep networks. Each shade of blue represents one standard deviation σ produced by the model.

B.1.2 Hemisphere world

For this experiment, we created a synthetic world with a 6×6 grid of landmarks, each spaced one meter apart. Our monocular camera resided on a hemisphere (of radius 25 meters) from the centre of the landmark grid. The camera sensor was 500×500 pixels, with a principal point in the middle of the sensor and a focal length of 500 pixels. We added zero-mean Gaussian noise of unit pixel variance to each landmark projection.

The network consisted of five residual blocks, each containing a fully connected layer and a ReLU non-linearity. For each camera location, we projected all 36 landmarks onto the image plane, added noise, and then stored 72 image coordinates as training or test input.

Table B.1: Hyper-parameters for 1D training.

Uncertainty Method	Learning Rate	Momentum	Dropout (%)
Dropout	0.05	0.5	3
Direct Regression	0.0001	0	0
Bagging	0.01	0.9	0
HydraNet (no direct uncertainty)	0.01	0.9	0
HydraNet	0.01	0.1	0

B.1.3 7-Scenes

Figure B.5 presents regression results on all seven scenes from the 7-scenes dataset. Our model consisted of a `resnet34` body (pre-trained, but not frozen) with 25+1 heads in the same structure as the synthetic experiment. We used the Adam optimizer with a learning rate of 5×10^{-5} for all scenes, and trained each model for 15 epochs, selecting the one with the lowest negative log likelihood.

B.1.4 KITTI

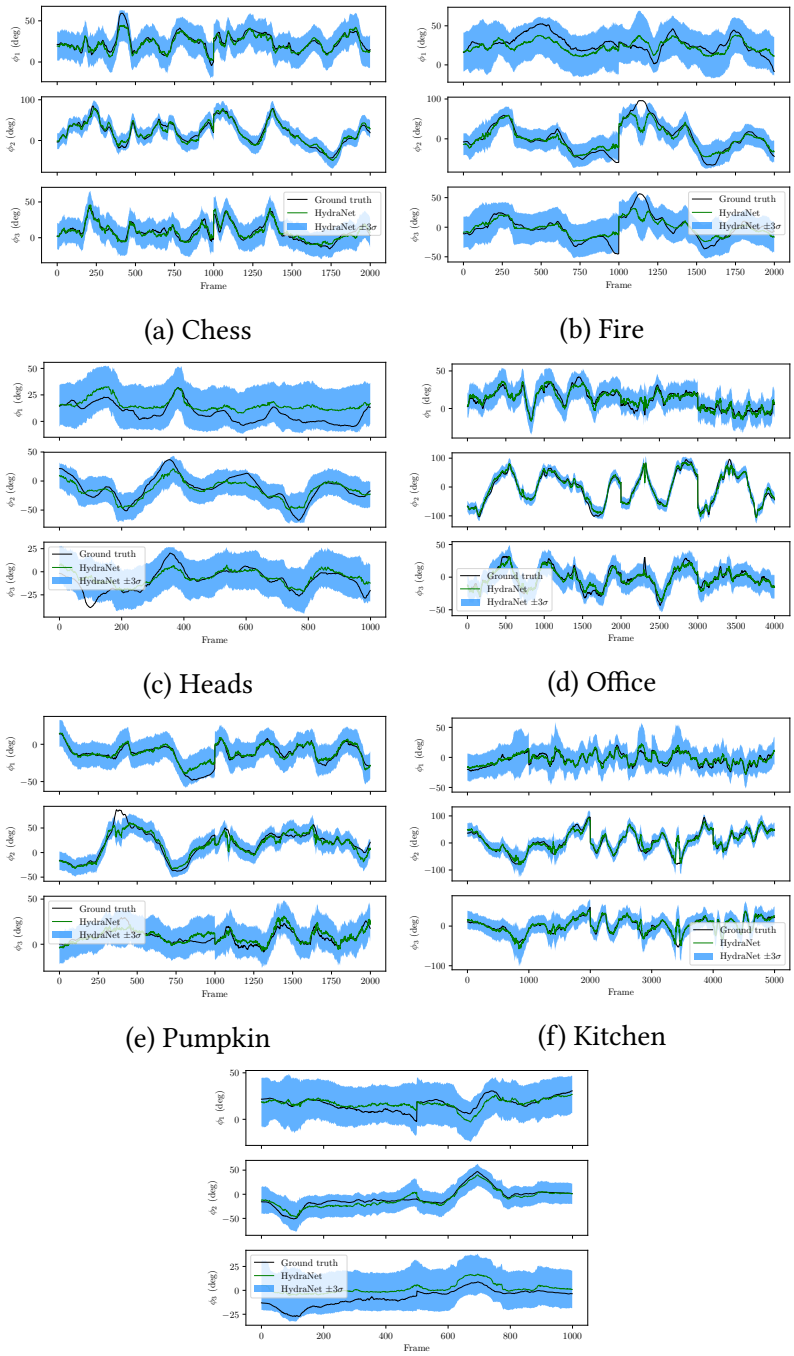
Network details

Our custom convolutional network was built using PyTorch as follows:

```
self.cnn = torch.nn.Sequential(
    convunit(2, 64),
    convunit(64, 128),
    convunit(128, 256),
    convunit(256, 512),
    convunit(512, 1024),
    convunit(1024, 1024),
    convunit(1024, 1024)
)
```

with each `conv_unit` defined as,

```
def convunit(in, out, ks=3, st=2, pad=1):
        return torch.nn.Sequential(
            torch.nn.Conv2d(in, out,
              kernelsize=ks,
              stride=st,
              padding=pad),
            torch.nn.BatchNorm2d(out),
            torch.nn.ReLU()
        )
```



(a) Chess

(b) Fire

(c) Heads

(d) Office

(e) Pumpkin

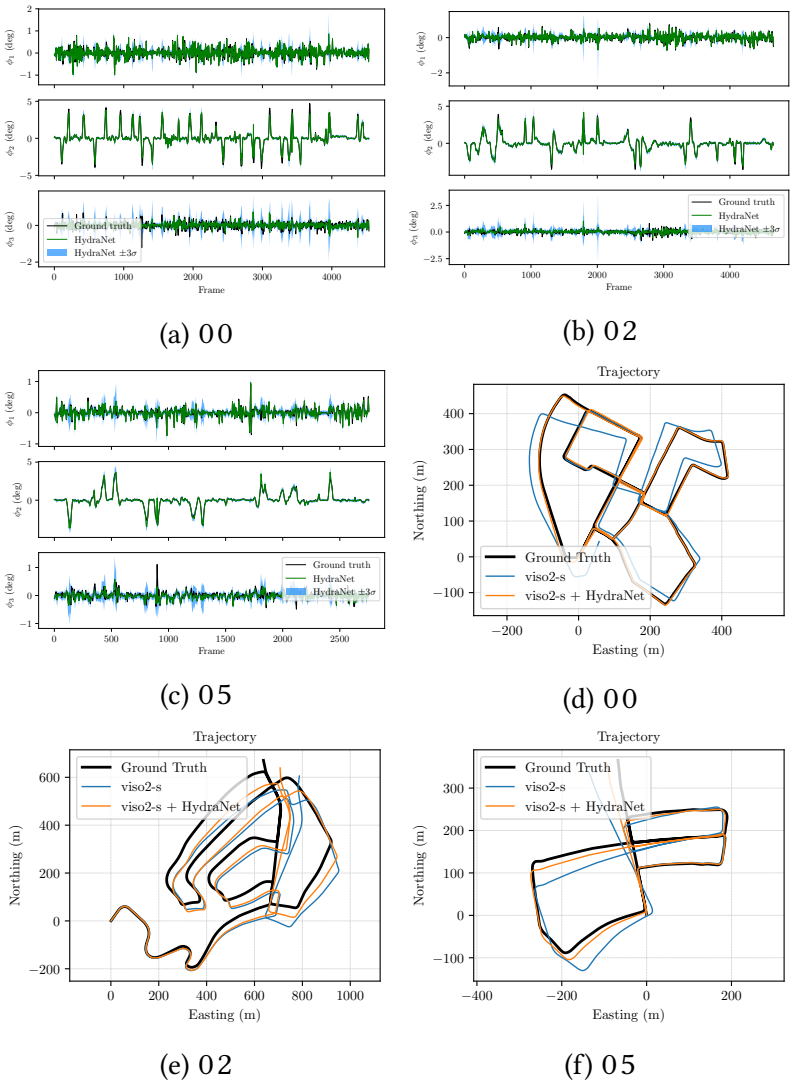
(f) Kitchen

(g) Stairs

Figure B.5: Probabilistic regression plots for all seven datasets from the 7-Scenes dataset.

and the head structure being identical to both of the previous experiments. Our two-dimensional flow image was constructed using OpenCV with the function `calcOpticalFlowFarneback()` from two RGB images converted to grayscale. We trained the network using the Adam op-

timizer, with a learning rate of 5×10^{-5} and no pre-training. We found that augmenting the dataset with rotation targets and inputs that represented both the forward and reverse temporal pairs improved generalization.



(a) 00 (b) 02

(c) 05 (d) 00

(e) 02 (f) 05

Figure B.6: KITTI frame-to-frame rotation probabilistic regression for sequences 00, 02 and 05. Top-down trajectory plots show localization improvements after fusion with a classical stereo visual odometry pipeline.

Keyframe direct VO

To compare with an accurate classical method, we used a state-of-the-art dense direct stereo visual odometry method based off of keyframes and minimization of pixel re-projection error. This method is similar to the dense method presented in Peretroukhin and Kelly (2018) and largely based on the ideas in **?**. We plan to release this pipeline after the double-blind review process.

Pose graph relaxation

As discussed in the main paper, we fused our probabilistic rotation regression with classical stereo visual odometry using pose graph relaxation implemented with the help of a Python-based factor graph library which we will publicize after the review process. Using that framework, we solved

$$\mathbf{T}_{1,w}^{*}, \mathbf{T}_{2,w}^{*} = \underset{\mathbf{T}_{1,w}, \mathbf{T}_{2,w} \in \mathrm{SE}(3)}{\operatorname{argmin}} \mathcal{L}(\hat{\mathbf{T}}_{2,1}, \hat{\mathbf{C}}_{2,1}) \tag{B.1}$$

$$= \underset{\mathbf{T}_{1,w}, \mathbf{T}_{2,w} \in \mathrm{SE}(3)}{\operatorname{argmin}} \boldsymbol{\xi}_{1,2}^{T} \boldsymbol{\Sigma}_{\text{vo}}^{-1} \boldsymbol{\xi}_{1,2} + \boldsymbol{\phi}_{1,2}^{T} \boldsymbol{\Sigma}_{\text{hn}}^{-1} \boldsymbol{\phi}_{1,2} \tag{B.2}$$

where

$$\boldsymbol{\xi}_{1,2} = \operatorname{Log}\left(\left(\mathbf{T}_{2,w} \mathbf{T}_{1,w}^{-1} \right) \hat{\mathbf{T}}_{2,1}^{-1} \right), \tag{B.3}$$

$$\boldsymbol{\phi}_{1,2} = \operatorname{Log}\left(\left(\mathbf{C}_{2,w} \mathbf{C}_{1,w}^{T} \right) \hat{\mathbf{C}}_{2,1}^{T} \right), \tag{B.4}$$

and $\hat{\mathbf{T}}_{2,1}$, $\boldsymbol{\Sigma}_{\text{vo}}$ and $\hat{\mathbf{C}}_{2,1}$, $\boldsymbol{\Sigma}_{\text{hn}}$ were provided by our classical estimator and the HydraNet network respectively. Note that $\boldsymbol{\Sigma}_{\text{hn}} \in \mathbb{R}^{3\times3} \geq 0$ while $\boldsymbol{\Sigma}_{\text{vo}} \in \mathbb{R}^{6\times6} \geq 0$. We also overload the logarithm function, $\operatorname{Log}(\cdot)$ to represent both $\mathrm{SE}(3)$ and $\mathrm{SO}(3)$ logarithmic maps as necessary. To account for gauge freedom, we fixed the first transformation to identity, $\mathbf{T}_{1,w} = \mathbf{1}$, and initialized $\mathbf{T}_{2,w}$ to $\hat{\mathbf{T}}_{2,1}$. After convergence, we composed the final frame-to-frame estimate as $\mathbf{T}_{2,1}^{*} = \mathbf{T}_{2,w}^{*} \left(\mathbf{T}_{1,w}^{*} \right)^{-1} = \mathbf{T}_{2,w}^{*}$.

Additional results

We present additional regression and fused trajectory results for KITTI odometry benchmark sequences 00, 02 and 05 in Figure B.6. We trained each test sequence with the remainder of the sequences in the dataset, mimicking a cross-validation approach.

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. (2014). Associating uncertainty with three-dimensional poses for use in estimation problems. *IEEE Trans. Rob.*, 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.

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.

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.

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. (2014). SVO: Fast semi-direct monocular visual odometry. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 15–22.

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

Furgale, P., Carle, P., Enright, J., and Barfoot, T. D. (2012). The devon island rover navigation dataset. *Int. J. Rob. Res.*, 31(6):707–713.

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. (2013). Vision meets robotics: The KITTI dataset. *Int. J. Rob. Res.*, 32(11):1231–1237.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition, (CVPR)*, pages 3354–3361.

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.

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.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 6402–6413. Curran Associates, Inc.

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.

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.

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 Roy, B. V. (2016a). Deep exploration via bootstrapped DQN. *CoRR*, abs/1602.04621.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. (2016b). 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. (2016a). PROBE-GK: Predictive robust estimation using generalized kernels. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 817–824.

Peretroukhin, V., Vega-Brown, W., Roy, N., and Kelly, J. (2016b). PROBE-GK: Predictive robust estimation using generalized kernels. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'16)*, pages 817–824, Stockholm, Sweden.

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.

Rosen, D. M., Carlone, L., Bandeira, A. S., and Leonard, J. J. (2019). SE-Sync: A certifiably correct algorithm for synchronization over the special euclidean group. *Int. J. Rob. Res.*, 38(2-3):95–125.

Scaramuzza, D. and Fraundorfer, F. (2011). Visual odometry [tutorial]. *IEEE Robot. Autom. 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., 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. (2015). 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. 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.

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