

ON LEARNING PSEUDO-SENSORS TO IMPROVE VISUAL EGOMOTION ESTIMATION

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

© Copyright 2019 by Valentin Peretroukhin

Abstract

On learning pseudo-sensors to improve visual egomotion estimation

Valentin Peretroukhin

Doctor of Philosophy

Graduate Department of Institute for Aerospace Studies

University of Toronto

2019

The ability to estimate *egomotion* 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 information may be intermittent or unavailable. Visual egomotion estimation, also known as *visual odometry*, has become ubiquitous in mobile robotics due to the availability of high-quality, compact, and inexpensive cameras that capture rich representations of the world. To remain computationally tractable, ‘classical’ visual odometry pipelines make simplifying assumptions that, while permitting reliable operation in ideal conditions, often lead to systematic error. In this dissertation, we present several ways in which conventional pipelines can be improved through the addition of a *pseudo-sensor*. In this context, a *pseudo-sensor* is a data-driven model that infers latent information that is difficult to extract through analytic measurement models. By fusing the output of such pseudo-sensors with traditional pipelines, we retain the performance of conventional techniques in nominal conditions while leveraging modern high-capacity data-driven models to improve uncertainty quantification, correct for systematic bias, and improve robustness to deleterious effects. We validate our approach on 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 supervisor¹, the perennial love of my family and friends², and the limitless patience of my lab mates³. Thank you all.

¹as well as all of my collaborators and academic mentors (special thanks to Lee)

²especially the support and encouragement of Elyse

³in humouring my insatiable need for debate and banter (special thanks to Lee)

Contents

1	Introduction	2
1.1	A Visual <i>Pipeline</i>	4
1.2	The Pseudo-Sensor	7
1.3	Original Contributions	8
2	Learned Probabilistic Sun Sensor	12
2.1	Background	12
2.1.1	Regularization and Dropout	12
2.1.2	Bayesian Modelling and Variational Inference	12
2.1.3	Monte Carlo Dropout as Approximate Variational Inference	14
3	Mathematical Foundations	16
3.1	Coordinate Frames	16
3.2	Rotations	17
3.2.1	Unit Quaternions	19
3.2.2	Topology	19
3.3	Spatial Transforms	20
3.3.1	Applying Transforms	21
3.4	Perturbations	22
3.5	Uncertainty through Injection	23
3.6	Deep Learning	24
3.6.1	Feed-forward Neural Networks	24
3.6.2	Convolutional Neural Networks	25
3.6.3	Supervised Training	26
3.6.4	Practical Considerations	27
4	Classical Visual Odometry	27
4.1	A Taxonomy of VO	28

4.2	Canonical VO Pipeline	28
4.2.1	Preprocessing	28
4.2.2	Data Association	29
4.2.3	Maximum Likelihood Motion Solution	31
4.3	Robust Estimation	33
4.4	Outstanding Issues	35
5	Predictive Robust Estimation	37
5.1	Introduction	37
5.2	Motivation	38
5.3	Related Work	39
5.4	Predictive Robust Estimation for VO	40
5.4.1	Bayesian Noise Model for Visual Odometry	40
5.4.2	Generalized Kernels	41
5.4.3	Generalized Kernels for Visual Odometry	42
5.4.4	Inference without ground truth	45
5.5	Prediction Space	47
5.5.1	Angular velocity and linear acceleration	48
5.5.2	Local image entropy	48
5.5.3	Blur	48
5.5.4	Optical flow variance	50
5.5.5	Image frequency composition	51
5.6	Experiments	51
5.6.1	Simulation	52
5.6.2	KITTI	53
5.6.3	UTIAS	56
5.7	Summary	59
5	Learned Probabilistic Sun Sensor	53
5.1	Introduction	53
5.2	Motivation	54
5.3	Related Work	56
5.4	Background	59
5.4.1	Neural Networks for Parametric Learning	59
5.4.2	Convolutional Neural Networks	59
5.4.3	Regularization and Dropout	59
5.4.4	Dropout as Variational Inference	59

5.5	Sun-Aided Stereo Visual Odometry	59
5.5.1	Observation Model	59
5.5.2	Sliding Window Bundle Adjustment	60
5.6	Orientation Correction	61
5.7	Indirect Sun Detection using a Bayesian Convolutional Neural Network	62
5.7.1	Cost Function	63
5.7.2	Uncertainty Estimation	63
5.7.3	Implementation and Training	64
5.8	Simulation Experiments	65
5.9	Urban Driving Experiments: The KITTI Odometry Benchmark	73
5.9.1	Sun-BCNN Test Results	76
5.9.2	Visual Odometry Experiments	77
5.10	Planetary Analogue Experiments: The Devon Island Rover Navigation Dataset	78
5.10.1	Sun-BCNN Test Results	81
5.10.2	Visual Odometry Experiments	82
5.11	Sensitivity Analysis	84
5.11.1	Cloud Cover	84
5.11.2	Model Generalization	87
5.11.3	Mean and Covariance Computation	90
5.12	Summary	92
6	Learned Pose Corrections	92
6.1	Introduction	92
6.2	Motivation	93
6.3	Related Work	94
6.4	System Overview: Deep Pose Correction	95
6.4.1	Loss Function: Correcting SE(3) Estimates	97
6.4.2	Loss Function: SE(3) Covariance	97
6.4.3	Loss Function: SE(3) Jacobians	98
6.4.4	Loss Function: Correcting SO(3) Estimates	100
6.4.5	Pose Graph Relaxation	100
6.5	Experiments	101
6.5.1	Training & Testing	101
6.5.2	Estimators	103
6.5.3	Evaluation Metrics	104
6.6	Results & Discussion	110

6.6.1	Correcting Sparse Visual Odometry	110
6.6.2	Distorted Images	111
6.7	Summary	111
7	Learned Probabilistic Rotations	112
7.1	Introduction	112
7.2	Motivation	113
7.3	Related work	114
7.4	Approach	115
7.4.1	Why Rotations?	115
7.4.2	Probabilistic Regression	116
7.4.3	Deep Probabilistic $\text{SO}(3)$ Regression	118
7.4.4	Loss Function	120
7.5	Experiments	122
7.5.1	Uncertainty Evaluation: Synthetic Data	122
7.5.2	Absolute Orientation: 7-Scenes	124
7.5.3	Relative Rotation: KITTI Visual Odometry	124
7.6	Summary	130
8	Conclusion	131
8.1	Summary of Contributions	132
8.1.1	Predictive Robust Estimation	132
8.1.2	Sun BCNN	132
8.1.3	Deep Pose Corrections	133
8.1.4	Deep Probabilistic Inference of $\text{SO}(3)$ with HydraNet	133
8.2	Future Work	133
8.3	Final Remarks	134
8.4	Coda: In Search of Elegance	135
Appendices		138
A	PROBE: Isotropic Covariance Models through K-NN	139
A.1	Introduction	139
A.2	Theory	139
A.3	Training	140
A.4	Testing	141
A.5	Experiments	142

B Visual Odometry Implementation Details	144
B.1 Overview	144
B.2 Solution with Robust Loss	145
B.3 Deriving the Necessary Jacobians	146
Bibliography	148

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 column vectors in homogeneous coordinates.
- \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.
- $\underline{\mathcal{F}}_a$: 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^{cb} : A vector (resp. homogenous coordinates) from point b to point c (denoted by the superscript) and expressed in $\underline{\mathcal{F}}_a$ (denoted by the subscript).
- \mathbf{C}_{ab} : The 3×3 rotation matrix that transforms vectors from $\underline{\mathcal{F}}_b$ to $\underline{\mathcal{F}}_a$: $\mathbf{p}_a^{cb} = \mathbf{C}_{ab}\mathbf{p}_b^{cb}$.
- \mathbf{T}_{ba} : The 4×4 transformation matrix that transforms homogeneous points from $\underline{\mathcal{F}}_a$ to $\underline{\mathcal{F}}_b$: $\mathbf{p}_b^{cb} = \mathbf{T}_{ba}\mathbf{p}_a^{ca}$.

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*

In this dissertation, we present a general approach to improve visual egomotion estimation for mobile autonomous platforms. By *egomotion estimation*, we refer to the process of computing the motion of a rigid body relative to a fixed coordinate frame using measurements from sensors attached to the body (hence *egomotion*¹).

Remark (Autonomy through history). Mobile *automata* have been a part of human culture since antiquity. In ancient India, the king Ajatashatru was said to use *bhuta vahana yanta* ('spirit movement machines') to protect the relics of Gautama Buddha after his death in the fourth century BCE. According to Burmese legend, the bhuta vahana yanta of Ajatashatru were made with stolen secrets from a group of Greco-Roman 'roboticists' named the *yantakara*. The methods of the yantakara were closely guarded, and mechanical assassins were said to pursue those who attempted to disseminate them (Mayor, 2019).

In the millennia since, mobile automata were relegated to isolated demonstrations (e.g., the programmable cart of Hero of Alexandria or the 'autonomous' knight of Leonardo da Vinci) or to imagined forms in cautionary tales (e.g., Mary Shelley's *Frankenstein*). Although the

¹To the best of the author's knowledge, this term originates in experimental psychology (Warren, 1976), and was also referred to as *passive navigation* in the context of camera motion (Bruss and Horn, 1983).

secrets of the yantakara may never be rediscovered, the ancient pursuit of helpful automata has found new life towards the turn of the twenty-first century and yielded machinery and algorithms that show great promise in aiding humanity.

One way to estimate egomotion is to measure rates (e.g., linear or rotational velocities) and integrate them to compute changes in position and orientation. This method, referred to as *dead reckoning*, was used by marine navigators of ancient times to determine the longitude of a ship at sea. Unlike latitude, which navigators could determine by measuring the altitude of the noon Sun, or the altitude of Polaris (the North Star) at night, longitude was not reliably computed from celestial measurements until the development of the marine chronometer in the 18th century (Barfoot, 2017). As a result, early seafarers could only *dead reckon* east-west motion by relying on estimates of local water currents, and by measuring the ship's magnetic heading and its speed relative to water (through a tool called a *chip log*, Figure 1.1).

In a similar process, early aviators computed egomotion through magnetic heading, airspeed, and an estimate of prevailing winds. While all dead-reckoning-based egomotion estimates exhibit unbounded error growth, these early methods were particularly inaccurate and required regular corrections through known landmarks.² In the mid twentieth century, the goals of intercontinental flight and space exploration necessitated the development of more accurate egomotion sensors (e.g., gimballed inertial sensors including accelerometers and gyroscopes) and an associated set of estimation techniques that could compute egomotion without human intervention (e.g., the Kalman filter (Grewal and Andrews, 2010)).

By the late twentieth century, unmanned Lunar and Martian exploration motivated the development of a new approach to egomotion estimation. Although ground vehicles like extra-planetary rovers could infer egomotion through *wheel odometry* (integrating wheel rates and

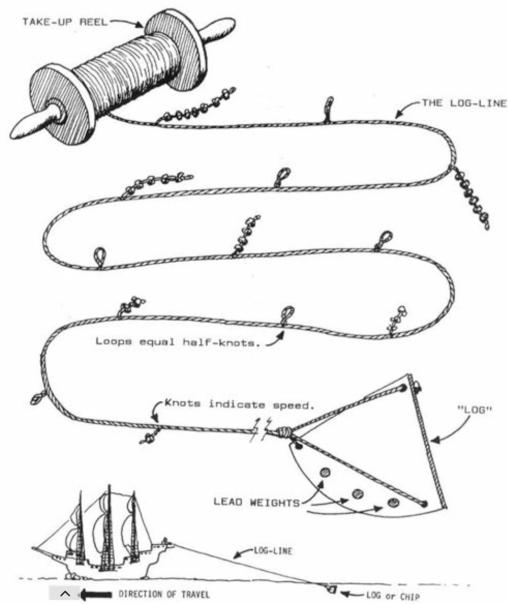


Figure 1.1: A *chip log* was a tool used to measure ship speed. The *log* or *chip* was tossed into the water, and speed was measured by the amount of *knots* that unravelled in a set time interval (credit: ocean-motion.org).

²Perhaps the most famous example of dead-reckoning error was made by Christopher Columbus in 1492. He believed he had reached the *Indies* (modern Indonesia) but had really arrived in the Bahamas.

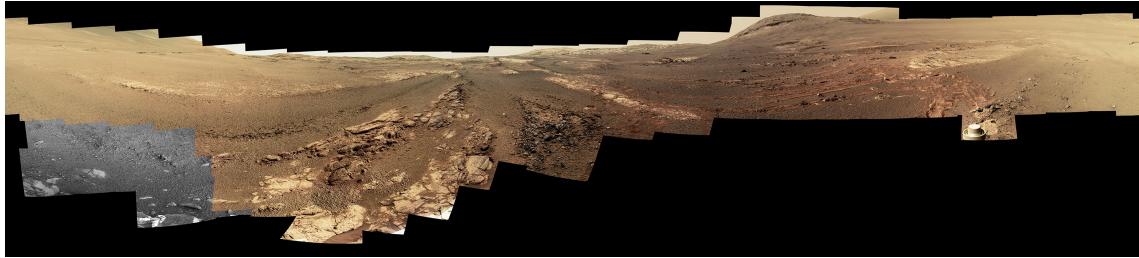


Figure 1.3: The last 360 degree panorama of the rocky Martian surface taken by the Pancam apparatus of the Mars Exploration Rover, *Opportunity*, at its final resting place, the western rim of the Endeavour Crater (*credit: NASA/JPL-Caltech/Cornell/ASU*).

using wheel orientation to drive a kinematic model), this approach was highly inaccurate on surfaces that induce wheel slip (e.g., the rock and sand covered surface of Mars, Figure 1.3). To address this, a number of researchers in the 1980s developed the technique of *visual odometry* (Scaramuzza and Fraundorfer, 2011) (or VO), as a way to infer egomotion from sequentially-collected images. The mathematical basis of VO was closely tied to the technique of photogrammetric *bundle adjustment* (Triggs et al., 2000) which originates from 19th century photogrammetry (Albertz, 2007).

In the nearly four decades since the first development of VO, the proliferation of compact, relatively-inexpensive, high-resolution cameras has made vision-based sensing techniques ubiquitous in mobile autonomous applications. In addition to computing egomotion, camera data can be used to build detailed maps of an environment (which can be computed in tandem with egomotion through a technique called Simultaneous Localization and Mapping, or SLAM (Cadena et al., 2016)), as well as detect, track and avoid other objects (Figure 1.2).

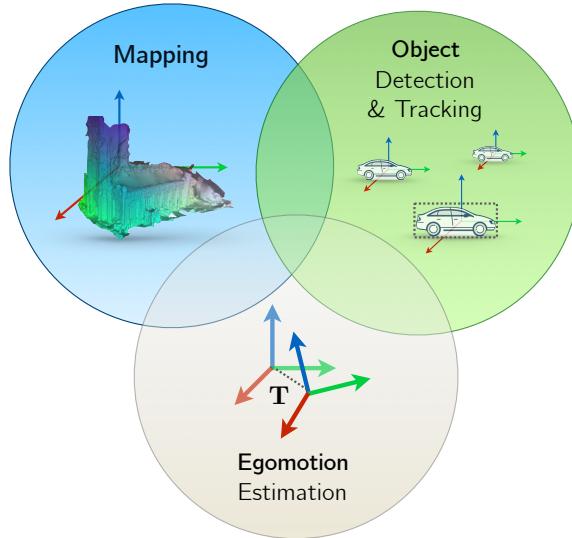


Figure 1.2: Visual egomotion estimation can be useful to create maps and to detect and track other objects.

1.1 A Visual Pipeline

Central to *classical* visual odometry algorithms (which, in this context, refers to the bulk of VO research published during what Cadena et al. (2016) call the *classical* and *algorithmic-analysis* ages of VO and SLAM research between 1986 and 2015) is the idea of a processing pipeline. A pipeline consists of several connected

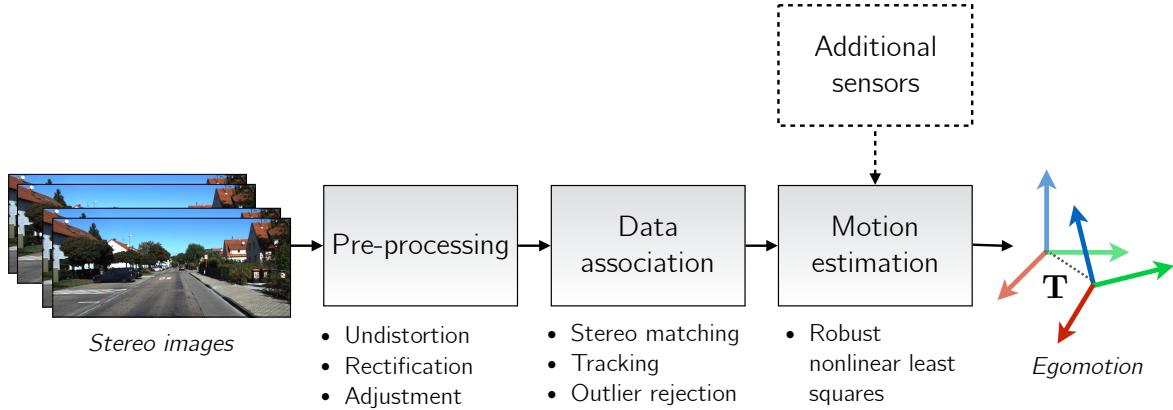


Figure 1.4: A ‘classical’ visual odometry pipeline consists of several distinct components with interpretable inputs and outputs. We list common examples of each component.

computational ‘blocks’ or ‘stages’ that have interpretable inputs and outputs. By carefully processing information contained within raw sensor data, pipelines facilitate the construction of complex state estimation architectures that can fuse visual observations with other sensors of varied modality to create maps and models of the external world and infer the egomotion of a mobile platform within it. In this dissertation, we will largely deal with the improvement of a canonical visual odometry pipeline— we illustrate its major components in Figure 1.4.

Broadly, VO solutions based on the idea of hand-crafted pipelines (Leutenegger et al., 2015; Cvišić and Petrović, 2015; Tsotsos et al., 2015; Alcantarilla and Woodford, 2016; Forster et al., 2014; Wang et al., 2017a; Engel et al., 2018) have achieved impressive localization accuracy within a variety of settings. They have been used to compute the egomotion of self-driving cars through urban roads, and track autonomous drones through aggressive flying maneuvers.

Building on momentum from the computer vision community, a significant part of the visual state estimation literature has also considered replacing classical pipelines with parametric modelling through deep convolutional neural networks (CNNs) and data-driven learning. Although initially developed for image classification (LeCun et al., 2015), CNN-based measurement models have been applied to numerous problems in visual state estimation including homography estimation (DeTone et al., 2016), single image depth reconstruction (Garg et al., 2016), camera re-localization (Kendall and Cipolla, 2016), and place recognition (Sünderhauf et al., 2015). A number of recent CNN-based approaches have also tackled the problem of visual egomotion estimation, often purporting to obviate the need for classical visual localization pipelines by learning pose changes *end-to-end*, without requiring intermediate outputs (Melekhov et al., 2017; Handa et al., 2016; Oliveira et al., 2017).

In light of these new approaches, debate has emerged within the robotics and computer vision communities regarding the extent to which data-driven parametric models should replace pipelines. Although classical approaches can achieve high accuracy under nominal conditions, deep data-driven networks have the potential to improve upon them in two respects.

First, owing to their representational power, deep parametric networks have the potential to learn bespoke data associations that are robust to (1) large viewpoint changes, (2) moving objects, and (3) self-similar visual textures (e.g., indoor walls, grass, sky, etc.). Such robust, invariant data associations have been the holy-grail of hand-crafted pipelines, and traditional data association methods must be carefully tuned to work well in a given setting ([Schonberger et al., 2017](#)).

Second, deep parametric regression of egomotion has the potential to more efficiently use dense high-dimensional visual data. To remain computationally tractable, classical VO pipelines typically take one of two approaches. First, some pipelines ([Leutenegger et al., 2015; Cvišić and Petrović, 2015](#)) choose to indirectly summarize visual data by extracting and matching a set of sparsely-distributed salient *features*. These features may be robust to some of the effects mentioned above, but, by construction, they discard large portions of visual data that may inform more accurate egomotion estimates. Alternatively, other methods ([Forster et al., 2014; Wang et al., 2017a; Engel et al., 2018](#)) may instead choose to match photometric intensity directly through an assumption of photometric consistency. This relatively simple schema permits dense data association but can be a poor assumption in the presence of large viewpoint changes or non-Lambertian surfaces. Further, this approach yields a high number of data associations and produces a highly non-convex objective that requires care to optimize.

In contrast, deep networks built on basic image operations with modern computationally-efficient implementations have the potential to avoid the pitfalls of both of these approaches while remaining tractable (assuming appropriate hardware). Despite this potential, current end-to-end learning techniques for egomotion estimation have a number of disadvantages (see Table 1.1). They often generalize poorly to new environments, come with few analytical guarantees, provide only point estimates of latent parameters, and do not allow for intermediate representations that have been shown to improve generalization performance on visual tasks ([Zhou et al., 2019](#)). Indeed, according to one benchmark, the most accurate visual egomotion approach at the time of writing³ remains a classical pipeline based on carefully selected sparse features.

³Based on the KITTI Odometry benchmark leaderboard at http://www.cvlibs.net/datasets/kitti/eval_odometry.php.

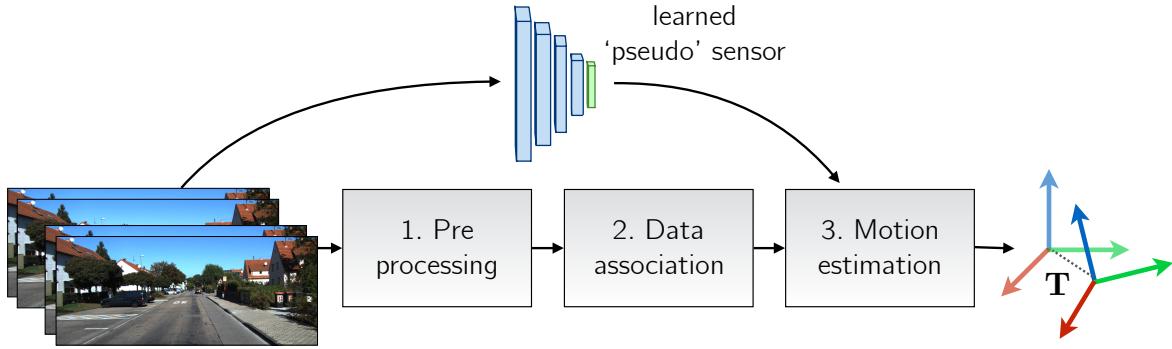


Figure 1.5: A learned *pseudo-sensor* extracts latent information from the same data stream.

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

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

1.2 The Pseudo-Sensor

As mobile autonomy enters the robust-perception age ([Cadena et al., 2016](#)), classical pipelines that work in limited contexts will need to be adapted and augmented to ensure they can operate over longer time-periods, and through challenging environments. However, replacing these performant pipelines completely with learned approaches, is in our view, unnecessary. Instead, we propose the paradigm of the *pseudo-sensor* (Figure 1.5). A *pseudo-sensor* is a machine-learning-based (hyper-)parametric model that extracts latent information from sensor data that is difficult to model analytically. Pseudo-sensors leverage the representational

power of modern data-driven learning techniques to extract useful quantities that make existing classical pipelines more consistent and accurate in a given environment. In this dissertation, we present four such pseudo-sensors. In each case, the output of the pseudo-sensor is fused with a classical visual odometry pipeline to produce better motion estimates. To accomplish this fusion, we rely on two approaches. The first (Predictive Robust Estimation, or PROBE, Chapter 5), uses the paradigm of a pseudo-sensor to build a heteroscedastic noise model from extant sensor data. By predicting uncertainty information, PROBE effectively re-scales a robust loss function to better account for deleterious visual effects. The second approach (used by Sun-BCNN, DPC-Net, and HydraNet, Chapters 5 to 7 respectively) produces geometric quantities (probabilistic estimates of an illumination source, SE(3) corrections to existing egomotion estimates, and independent probabilistic rotation estimates, respectively), that can be fused with the original egomotion estimate through pose graph optimization.

1.3 Original Contributions

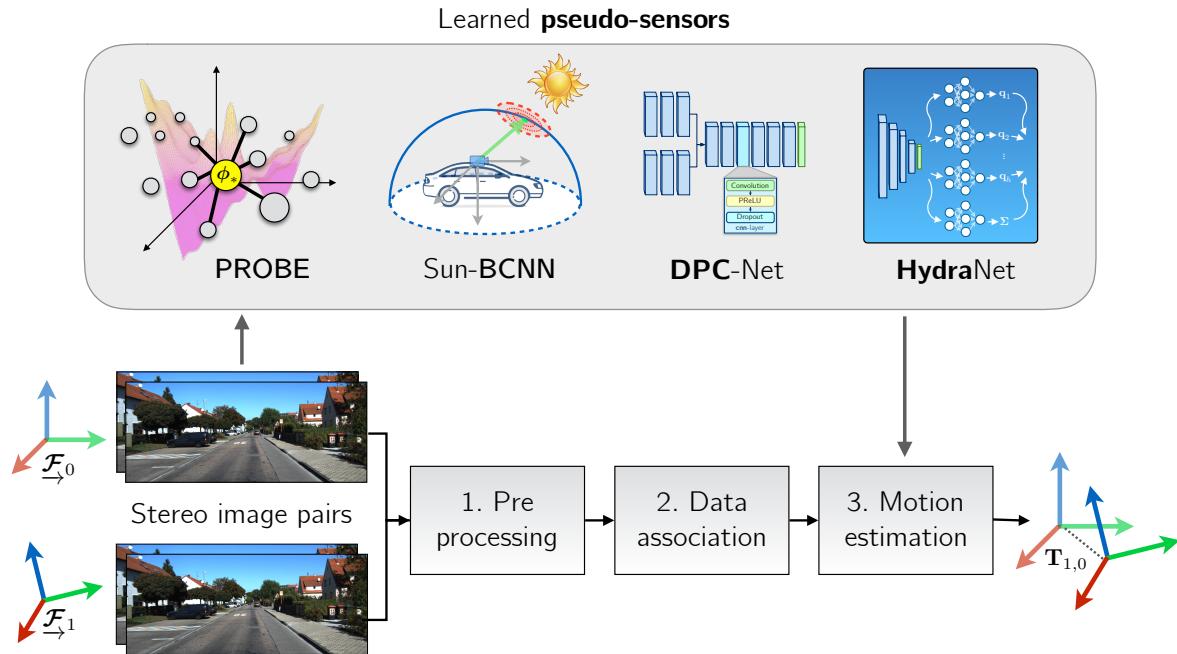


Figure 1.6: This dissertation details four examples of *pseudo-sensors* that improve 'classical' egomotion estimation through data-driven learning.

This dissertation consists of several published contributions under the umbrella of *learned pseudo-sensors* that improve a canonical visual egomotion pipeline. Before detailing each pseudo-sensor, we present some mathematical foundations (Chapter 3) and a common base-

line for a feature-based stereo visual odometry pipeline (Chapter 4) 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 5, Appendix A) builds a predictive model for observation uncertainty from training data. To do this, we collaborate with William Vega-Brown at MIT to adapt the technique of Generalized Kernel (GK) estimation ([Vega-Brown et al., 2014](#)) to visual odometry. Generalized Kernel estimation allows us to build an efficient Bayesian model for stereo tracking uncertainty. By setting a prior on covariance, we derive a ‘robust’ objective that can be *predictively* scaled to improve the accuracy and consistency of a feature-based stereo visual odometry pipeline. PROBE is associated with three publications listed below. The first two publications (and Appendix A) explore useful predictors for uncertainty and build a non-Bayesian isotropic covariance model. The latter publication presents the Bayesian GK approach.

- 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. Sun-BCNN: Learned sun sensor

Sun-BCNN (Chapter 5) is a virtual sun sensor based on a Bayesian Convolutional Neural Network (BCNN) that was developed in collaboration with Lee Clement. Much like a dedicated hardware sun sensor, Sun-BCNN infers a probabilistic estimate of the direction of the sun that can be used to inject orientation information into an egomotion pipeline. However, unlike dedicated sensors, Sun-BCNN requires no additional hardware and can predict both a mean and uncertainty from a single RGB image. It is associated with three publications listed below. The first publication consists of initial

exploratory work on virtual sun sensors, while the second presents the BCNN formulation. The final publication is an extended journal article that summarizes the results of the prior two conference publications and adds experiments from data collected in the Canadian High Arctic and around Oxford, UK, as well as investigating the effect of cloud cover and transfer learning.

- 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*, 37(9):996–1016.

3. DPC-Net: Learned pose corrections

Deep Pose Correction (Chapter 6) is an approach to improving egomotion estimates through SE(3) pose corrections learned with deep networks. As part of this work, we derive a novel loss function based on Lie theory that permits the learning six degree-of-freedom pose residuals in a supervised learning framework. After training, our Deep Pose Correction Network (DPC-Net) predicts low-rate, ‘small’ *corrections* that can be fused with egomotion estimates from a canonical pipeline. DPC-Net does not require any modification to an existing pipeline, and can learn to correct multi-faceted errors from estimator bias, sensor mis-calibration or environmental effects. It is associated with one journal publication listed below.

- Peretroukhin, V. and Kelly, J. (2018). DPC-Net: Deep pose correction for visual localization. *IEEE Robotics and Automation Letters*, 3(3):2424–2431.

4. HydraNet: Learned probabilistic rotation estimation

Finally, HydraNet (Chapter 7) is a multi-headed network structure that can regress probabilistic estimates of rotation (elements of the matrix Lie group, SO(3)) that account for both aleatoric and epistemic uncertainty. HydraNet builds upon results from

both Sun-BCNN and DPC that show that correcting rotation is critical to accurate egomotion estimation. Towards this end, HydraNet is designed to produce well-calibrated notions of uncertainty over $\text{SO}(3)$ that facilitate fusion with classical egomotion pipelines through 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 $\text{SO}(3)$ 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

Learned Probabilistic Sun Sensor

He stepped down, avoiding any long look at her as one avoids long looks at the sun, but seeing her as one sees the sun, without looking.

Leo Tolstoy, *Anna Karenina*

2.1 Background

2.1.1 Regularization and Dropout

Recall from Section 3.6.4 that dropout (Figure 5.4) was originally proposed as a regularization technique that improves the generalization performance of deep networks.

2.1.2 Bayesian Modelling and Variational Inference

Our goal is to generate a function $f(\mathbf{x}; \boldsymbol{\pi})$ that maps inputs \mathbf{x} to outputs \mathbf{y} a function of parameters $\boldsymbol{\pi}$. Recall from Chapter 5, that in Bayesian inference, we define a *prior* on the space of parameters $p(\boldsymbol{\pi})$ and a *likelihood function* that defines how likely we are to observe \mathbf{y} given an input \mathbf{x} and parameters $\boldsymbol{\pi}$, $p(\mathbf{y}|\mathbf{x}, \boldsymbol{\pi})$. For example for regression problems, we may define the likelihood

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\pi}) = \mathcal{N}(f(\mathbf{x}; \boldsymbol{\pi}), \tau^{-1}\mathbf{1}) \quad (2.1)$$

where τ is a model *precision*. Given a dataset $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\} = \{\{\mathbf{x}_1, \dots, \mathbf{x}_N\}, \{\mathbf{y}_1, \dots, \mathbf{y}_N\}\}$, the Bayesian formulation seeks a posterior over the parameters $\boldsymbol{\pi}$:

$$p(\boldsymbol{\pi}|\mathcal{D}) = \frac{p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\pi})p(\boldsymbol{\pi})}{p(\mathbf{Y}|\mathbf{X})} \quad (2.2)$$

where we may think of this posterior as evaluating the sentence ‘the likelihood of the observed outputs given a particular set of parameters (weighed by a prior) compared to how likely they are in general’. This latter ‘in general’ quantity is the marginal likelihood (also known as the model *evidence*) over the space of parameters,

$$p(\mathbf{Y}|\mathbf{X}) = \int_{\boldsymbol{\pi}} p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\pi})p(\boldsymbol{\pi})d\boldsymbol{\pi}. \quad (2.3)$$

For a given test input \mathbf{x}^* , we can then *infer* a predictive distribution over the space of outputs by integrating over the posterior of model parameters:

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \int_{\boldsymbol{\pi}} p(\mathbf{y}^*|\mathbf{x}^*, \boldsymbol{\pi})p(\boldsymbol{\pi}|\mathcal{D})d\boldsymbol{\pi}. \quad (2.4)$$

For most problems, evaluating the posterior (Equation (2.2)) analytically is not problems. Instead, we can employ the technique of *variational inference* (Gal, 2016).

Variational Inference

In variational inference, we define a simpler variational density $q(\boldsymbol{\pi}; \boldsymbol{\theta})$ based on the auxiliary parameters $\boldsymbol{\theta}$. We then seek to minimize the Kullback-Leibler (KL) divergence between $q(\boldsymbol{\pi}; \boldsymbol{\theta})$ and the true posterior $p(\boldsymbol{\pi}|\mathcal{D})$:

$$\boldsymbol{\theta}^* = \operatorname{argmin}_{\boldsymbol{\theta}} \text{KL}(q(\boldsymbol{\pi}; \boldsymbol{\theta}) || p(\boldsymbol{\pi}|\mathcal{D})), \quad (2.5)$$

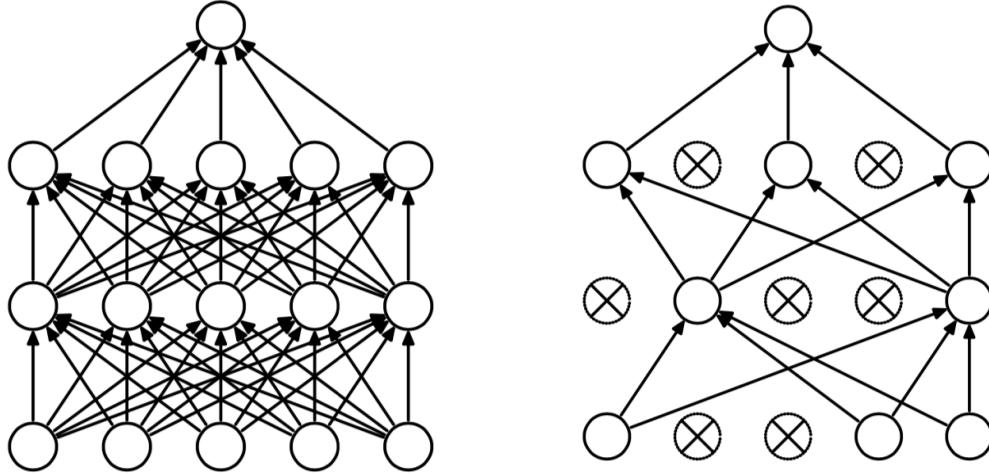
which then allows us to compute the approximate predictive distribution as

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) \approx \int_{\boldsymbol{\pi}} p(\mathbf{y}^*|\mathbf{x}^*, \boldsymbol{\pi})q(\boldsymbol{\pi}; \boldsymbol{\theta}^*)d\boldsymbol{\pi}. \quad (2.6)$$

One can show that minimizing KL divergence (Equation (2.5)) is the same as maximizing the evidence lower bound (ELBO)¹

$$\boldsymbol{\theta}^* = \operatorname{argmax}_{\boldsymbol{\theta}} \int_{\boldsymbol{\pi}} \log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\pi})q(\boldsymbol{\pi}; \boldsymbol{\theta})d\boldsymbol{\pi} - \text{KL}(q(\boldsymbol{\pi}; \boldsymbol{\theta}) || p(\boldsymbol{\pi})). \quad (2.7)$$

¹KL divergence and the ELBO differ by a constant that is not dependent on the auxiliary variables.



(a) Standard fully-connected neural network. (b) A neural network with dropout applied.

Figure 2.1: The technique of *dropout* stochastically removes the contribution of certain neurons to regularize learning. Figures from [Srivastava et al. \(2014\)](#).

Intuitively, maximizing the first term above (referred to as the expected log likelihood) ensures that $q(\pi; \theta)$ explains the dataset well, while minimizing the latter term regularizes the optimization so that $q(\pi; \theta)$ does not stray too far away from a prior.

2.1.3 Monte Carlo Dropout as Approximate Variational Inference

A key insight of Bayesian Neural Networks ([Gal, 2016](#); [Gal and Ghahramani, 2016b](#)) is the training objective of a neural network with dropout resembles the ELBO optimization of Equation (2.7) with a careful choice of $q(\pi; \theta)$ such that a particular condition holds (Eq. 3.12, [Gal \(2016\)](#)).

Namely, consider training a neural network with dropout and with *weight decay* (as is commonly done). The loss function for supervised training in this scenario can be written as

$$\mathcal{L}_{\text{dropout}}(\pi) = \underbrace{\frac{1}{N} \sum_{i=1}^N \mathcal{E}(\mathbf{y}_i, \tilde{\mathbf{y}}_i)}_{\text{error function}} + \underbrace{\lambda \sum_{\ell=1}^L (\|\mathbf{W}_\ell\|_2^2 + \|\mathbf{b}_\ell\|_2^2)}_{\text{weight decay}}, \quad (2.8)$$

where \mathbf{y}_i is the target output and $\tilde{\mathbf{y}}_i = NN(\mathbf{x}_i)$ is a stochastic sample of the network output.

Now consider that we define our auxiliary parameters for q to be the neural network

weight matrices, $\boldsymbol{\theta} = \{\mathbf{W}_\ell\}_{\ell=1}^L$, and treat the parameters of

$$q(\boldsymbol{\pi}; \boldsymbol{\theta}) = \text{diag}(\{b_{i\ell}\}) \mathbf{W}_\ell \quad (2.9)$$

$$b_{i\ell} \sim \text{Bernoulli}(p_i) \quad (2.10)$$

Namely we can use a stochastic estimator such that:

$$p(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D}) \approx \int_{\boldsymbol{\pi}} p(\mathbf{y}^* | \mathbf{x}^*, \boldsymbol{\pi}) q(\boldsymbol{\pi}; \boldsymbol{\theta}^*) d\boldsymbol{\pi} \quad (2.11)$$

$$\approx \frac{1}{T} \sum_{t=1}^T p(\mathbf{y}^* | \mathbf{x}^*, \tilde{\boldsymbol{\pi}}_t) \quad (2.12)$$

where $\tilde{\boldsymbol{\pi}}_t$ is a sample of our trained network with dropout applied at test time.

Chapter 3

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

3.1 Coordinate Frames

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

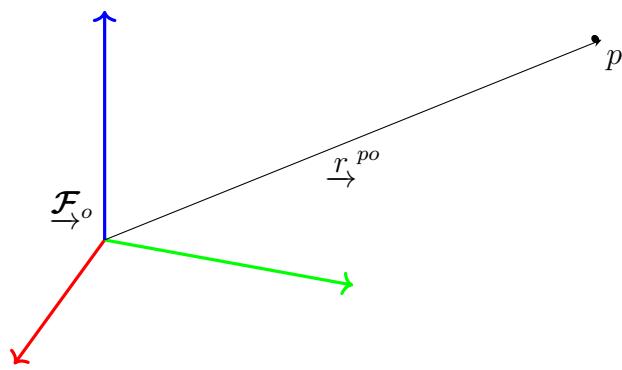


Figure 3.1: A position vector expressed in a coordinate frame.

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

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

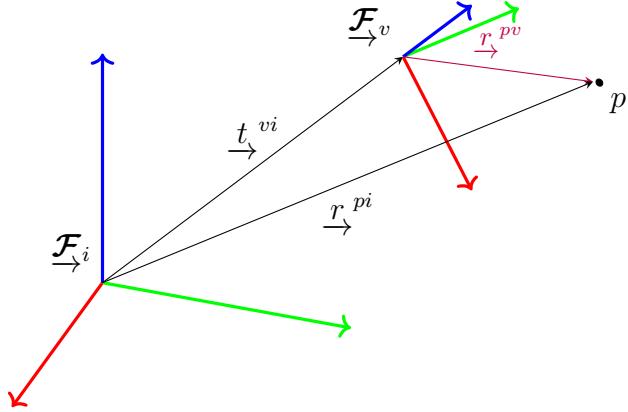


Figure 3.2: Two common references frames used throughout this thesis.

3.2 Rotations

The rotation matrix \mathbf{C} is a member of the matrix Lie group² $\text{SO}(3)$ (the Special Orthogonal group). We can define it as follows:

$$\text{SO}(3) = \{\mathbf{C} \in \mathbb{R}^{3 \times 3} \mid \mathbf{C}^T \mathbf{C} = \mathbf{1}, \det \mathbf{C} = 1\}. \quad (3.1)$$

Active and Passive Representations

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 dissertation 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

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

²A Lie group is a group that is also a differentiable manifold. See Barfoot (2017) for more details.

exponential map³ from three axis-angle parameters, $\phi = \phi\mathbf{a}$, $\phi \in \mathbb{R}$, $\mathbf{a} \in S^2$, to a rotation matrix, \mathbf{C} :

$$\mathbf{C} = \text{Exp}(\phi) = \exp(\phi^\wedge) = \sum_{n=0}^{\infty} \frac{1}{n!} (\phi^\wedge)^n \quad (3.2)$$

$$= \cos \phi \mathbf{1} + (1 - \cos \phi) \mathbf{a} \mathbf{a}^T + \sin \phi \mathbf{a}^\wedge, \quad (3.3)$$

where the wedge operator $(\cdot)^\wedge$ ⁴ is defined as

$$\mathbf{a}^\wedge = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}^\wedge = \begin{bmatrix} 0 & -a_2 & a_1 \\ a_2 & 0 & -a_0 \\ -a_1 & a_0 & 0 \end{bmatrix}. \quad (3.4)$$

Equation (3.3) is often referred to 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 (3.2) is surjective, we can define an inverse map if we restrict its domain to $0 \leq \phi < \pi$:

$$\phi = \text{Log}(\mathbf{C}) = \log(\mathbf{C})^\vee = \frac{\phi(\mathbf{C} - \mathbf{C}^T)^\vee}{2 \sin \phi}, \quad (3.5)$$

where $\phi = \arccos\left(\frac{\text{tr}(\mathbf{C}) - 1}{2}\right)$ and the *vee* operator, $(\cdot)^\vee : \mathbb{R}^{3 \times 3} \rightarrow \mathbb{R}^3$, is defined as the unique inverse of the wedge operator $(\cdot)^\wedge$. Note Equation (3.5) is undefined at both $\phi = 0$ and at $\phi = \pi$. In the former case, we can use a small-angle approximation and define

$$\text{Log}(\mathbf{C}) \approx (\mathbf{C} - \mathbf{1})^\vee \text{ when } \phi \approx 0. \quad (3.6)$$

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

³We follow Solà et al. (2018) and also define *capitalized* map for notational clarity.

⁴This operator is sometimes also expressed as $(\cdot)^\times$ or $[\cdot]_\times$ and is known as the skew-symmetric operator.

3.2.1 Unit Quaternions

We can also represent a rotation with unit quaternion, \mathbf{q} . A unit quaternion consists of a scalar value q_ω , and a three-dimensional vector component, \mathbf{q}_v :

$$\mathbf{q} = \begin{bmatrix} q_\omega \\ \mathbf{q}_v \end{bmatrix} \in S^3, \quad (\|\mathbf{q}\| = 1). \quad (3.7)$$

Unit quaternions also form a Lie group ([Solà et al., 2018](#)) and lie on a three-dimensional unit sphere within \mathbb{R}^4 . As with rotation matrices, we can define a surjective map from three parameters to the group itself,

$$\mathbf{q} = \text{Exp}(\phi) = \begin{bmatrix} \cos(\phi/2) \\ \mathbf{a} \sin(\phi/2) \end{bmatrix}. \quad (3.8)$$

By inspection, we can see that both \mathbf{q} and $-\mathbf{q}$ represent the same axis-angle pair, $\{\phi, \mathbf{a}\}$. As a result, unit quaternions represent a *double cover* of $\text{SO}(3)$ and we must be careful to account for this when using them as a rotation parametrization. In particular, when computing the logarithmic map,

$$\phi = \text{Log}(\mathbf{q}) = 2\mathbf{q}_v \frac{\arctan(\|\mathbf{q}_v\|, q_\omega)}{\|\mathbf{q}_v\|}, \quad (3.9)$$

we must account for the double cover by replacing \mathbf{q} with $-\mathbf{q}$ if q_ω is negative. Also note that as with rotation matrices Equation (3.9) is undefined when $\phi = 0$, but, importantly, we do not face any issues when $\phi = \pi$ due to the half-angle. In the former case, we can again rely on small angle approximations to define:

$$\text{Log}(\mathbf{q}) \approx \frac{\mathbf{q}_v}{q_\omega} \left(1 - \frac{\|\mathbf{q}_v\|^2}{3q_\omega^2} \right) \quad \text{when } \phi \approx 0. \quad (3.10)$$

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

3.2.2 Topology

Topologically, $\text{SO}(3)$ is *diffeomorphic*⁵ to the real projective space, \mathbb{RP}^3 , the space of all lines passing through the origin in \mathbb{R}^4 ([Hartley et al., 2013](#)). As a result, any global n -parametrization of $\text{SO}(3)$ will incur some cost. If we use rotation matrices ($n = 9$), we need to ensure or-

⁵A diffeomorphism is a smooth invertible function that maps one differentiable manifold to another.

thonormality and that $\det \mathbf{C} = 1$. With unit quaternions ($n = 4$), we need to account for the unit-norm constraint and take note of the double cover. Parametrizations with $n = 3$ (like Euler angles and axis-angle parameters) will be bounded, but unconstrained. However, due to the topological structure of $\text{SO}(3)$ all three-parameter parametrizations will not be invertible for certain rotations. With Euler angles, one has to be wary of *gimbal lock*, wherein two angles become indeterminate from each other. For axis-angle parameters, it is not possible to uniquely represent rotations whose angle is π .

Remark ($\text{SO}(3)$ Topology). To see why this is the case, consider that according to Euler's theorem, we can represent any element in $\text{SO}(3)$ by the axis-angle pair $\{\mathbf{a}, \phi\}$, $\mathbf{a} \in S^2$, and $0 \leq \phi \leq \pi$. We can partially represent this space by the closed ball of radius π in \mathbb{R}^3 using the combined axis-angle coordinates $\phi = \phi\mathbf{a}$. However, at the boundary ($\phi = \pi$), we must account for the fact that rotations represented by $\{\mathbf{a}, \pi\}$ are identical to those represented by $\{-\mathbf{a}, \pi\}$. In other words, we must *identify* all antipodal points, ϕ and $-\phi$ when $\|\phi\| = \pi$. This closed 3-ball with identified antipodal points on its boundary is topologically equivalent to the 3-sphere (S^3) with its antipodal points identified. In turn, this space is equivalent to \mathbb{RP}^3 since any line passing through the origin in \mathbb{R}^4 can be mapped to two unit normals, $\pm \mathbf{n} \in S^3$. This *identification* makes rotation representation particularly tricky in \mathbb{R}^3 and clearly explains why unit quaternions, $\mathbf{q} \in S^3$, are a *double* cover of $\text{SO}(3)$, since we must add the relation $\mathbf{q} = -\mathbf{q}$ to make these two spaces equivalent. We direct the reader to [Hartley et al. \(2013\)](#) who use the *gnomonic* projection to provide further geometric intuition for why $\text{SO}(3)$ is a projective space.

Accordingly, in this dissertation, we parametrize rotations as the constrained quantities, \mathbf{q} or \mathbf{C} . When dealing with perturbations about a given rotation (e.g., to compute updates to a state, or to propagate uncertainty), we use *small* rotations, $\delta\mathbf{C}$ or $\delta\mathbf{q}$, parametrized using three unconstrained parameters that we can assume are a one-to-one mapping to elements in $\text{SO}(3)$ (since, for small rotations, $\|\phi\| \ll \pi$).

3.3 Spatial Transforms

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

$$\text{SE}(3) = \{\mathbf{T} = \begin{bmatrix} \mathbf{C} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \mid \mathbf{C} \in \text{SO}(3), \mathbf{t} \in \mathbb{R}^3\}. \quad (3.11)$$

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

$$\mathbf{T} = \text{Exp}(\boldsymbol{\xi}) = \exp(\boldsymbol{\xi}^\wedge) = \sum_{n=0}^{\infty} \frac{1}{n!} (\boldsymbol{\xi}^\wedge)^n \quad (3.12)$$

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

$$\boldsymbol{\xi}^\wedge \triangleq \begin{bmatrix} \boldsymbol{\rho} \\ \phi \end{bmatrix}^\wedge = \begin{bmatrix} \boldsymbol{\phi}^\wedge & \boldsymbol{\rho} \\ \mathbf{0}^T & 0 \end{bmatrix}. \quad (3.13)$$

In practice, we can evaluate the exponential map through the Euler-Rodriguez formula (Equation (3.3)) and by computing the left-Jacobian of $\text{SO}(3)$, \mathbf{J} ,

$$\mathbf{T} = \text{Exp}\left(\begin{bmatrix} \boldsymbol{\rho} \\ \phi \end{bmatrix}\right) = \begin{bmatrix} \mathbf{C}(\phi) & \mathbf{J}(\phi)\boldsymbol{\rho} \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad (3.14)$$

where

$$\mathbf{J}(\phi) = \frac{\sin \phi}{\phi} \mathbf{1} + (1 - \frac{\sin \phi}{\phi}) \mathbf{a} \mathbf{a}^T + \frac{1 - \cos \phi}{\phi} \mathbf{a}^\wedge. \quad (3.15)$$

3.3.1 Applying Transforms

Applying our notation for coordinate frames (and referring back to Section 3.1), a transform, \mathbf{T}_{vi} can be expressed as

$$\mathbf{T}_{vi} = \begin{bmatrix} \mathbf{C}_{vi} & \mathbf{t}_v^{iv} \\ \mathbf{0}^T & 1 \end{bmatrix}. \quad (3.16)$$

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

$$\mathbf{r}_v^{pv} = \mathbf{T}_{vi} \mathbf{r}_i^{pi}, \quad (3.17)$$

or

$$\begin{bmatrix} \mathbf{r}_v^{pv} \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{C}_{vi} & \mathbf{t}_v^{iv} \\ \mathbf{0}^T & 1 \end{bmatrix}}_{\mathbf{T}_{vi}} \begin{bmatrix} \mathbf{r}_i^{pi} \\ 1 \end{bmatrix}, \quad (3.18)$$

which is numerically equivalent to

$$\mathbf{r}_v^{pv} = \mathbf{C}_{vi} \mathbf{r}_i^{pi} + \mathbf{t}_v^{iv}. \quad (3.19)$$

3.4 Perturbations

When solving optimization problems that involve rotations or rigid-body transforms, it is often useful to consider a small *perturbation* about an operating point. 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, $\bar{\mathbf{C}} \triangleq \text{Exp}(\bar{\boldsymbol{\phi}})$, in three different ways:

$$\mathbf{C} = \begin{cases} \text{Exp}(\delta\boldsymbol{\phi}^\ell) \bar{\mathbf{C}} & \text{left perturbation,} \\ \text{Exp}(\bar{\boldsymbol{\phi}} + \delta\boldsymbol{\phi}^m) & \text{middle perturbation,} \\ \bar{\mathbf{C}} \text{Exp}(\delta\boldsymbol{\phi}^r) & \text{right perturbation.} \end{cases} \quad (3.20)$$

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

$$\text{Exp}((\boldsymbol{\phi} + \delta\boldsymbol{\phi}^m)) \approx \text{Exp}(\mathbf{J}(\boldsymbol{\phi})\delta\boldsymbol{\phi}^m) \text{Exp}(\boldsymbol{\phi}). \quad (3.21)$$

From this it follows that $\delta\boldsymbol{\phi}^\ell \approx \mathbf{J}(\boldsymbol{\phi})\delta\boldsymbol{\phi}^m$ and elucidates why \mathbf{J} is called the *left Jacobian*.

In this dissertation, we will use the left and middle perturbations when appropriate. Using small angle approximations, the Euler-Rodriguez formula (Equation (3.3)) yields $\text{Exp}(\delta\boldsymbol{\phi}) \approx \mathbf{1} + \delta\boldsymbol{\phi}^\wedge$, which allows us to write the useful formula for the left perturbation:

$$\mathbf{C} = (\mathbf{1} + (\delta\boldsymbol{\phi}^\ell)^\wedge)\bar{\mathbf{C}}. \quad (3.22)$$

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

$$\mathbf{T} = \begin{cases} \text{Exp}(\delta\boldsymbol{\xi}^\ell) \bar{\mathbf{T}} & \text{left perturbation,} \\ \text{Exp}(\bar{\boldsymbol{\xi}} + \delta\boldsymbol{\xi}^m) & \text{middle perturbation,} \\ \bar{\mathbf{T}} \text{Exp}(\delta\boldsymbol{\xi}^r) & \text{right perturbation.} \end{cases} \quad (3.23)$$

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

$$\text{Exp}((\boldsymbol{\xi} + \delta\boldsymbol{\xi}^m)) \approx \text{Exp}((\mathcal{J}(\boldsymbol{\xi})\delta\boldsymbol{\xi}^m)) \text{Exp}(\boldsymbol{\xi}), \quad (3.24)$$

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

$$\mathcal{J}(\xi) \triangleq \begin{bmatrix} \mathbf{J}(\phi) & \mathbf{Q}(\xi) \\ \mathbf{0} & \mathbf{J}(\phi) \end{bmatrix}, \quad (3.25)$$

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

$$\mathbf{T} = (\mathbf{1} + (\delta\xi^\ell)^\wedge)\bar{\mathbf{T}}. \quad (3.26)$$

To derive locally linear systems from sets of rigid-body transforms, or ‘poses’, we can apply Equation (3.26). To update an operating point, we solve for $\delta\xi^\ell$ and then use the constraint-sensitive update $\mathbf{T} \leftarrow \text{Exp}(\delta\xi^\ell)\bar{\mathbf{T}}$.

Finally, we note that we will often drop the perturbation superscripts $(\cdot)^\ell$ and $(\cdot)^m$ after defining the perturbation scheme.

3.5 Uncertainty through Injection

We can also use perturbation theory to implicitly define uncertainty on constrained manifolds (see [Barfoot and Fur-gale \(2014\)](#) for a thorough discussion).

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

$$\mathbf{T} = \text{Exp}(\delta\xi)\bar{\mathbf{T}}. \quad (3.27)$$

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

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

$$\mathbf{C} = \text{Exp}(\delta\phi)\bar{\mathbf{C}}, \quad (3.28)$$

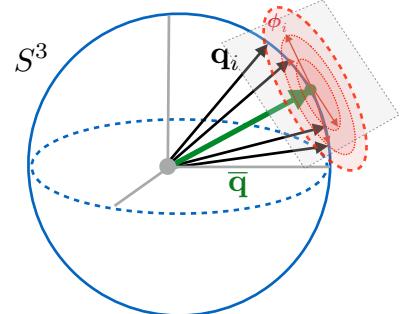


Figure 3.3: We can define uncertainty in the left tangent space of a mean element of a Lie group (here illustrated for unit quaternions).

and also, for unit quaternions,

$$\mathbf{q} = \text{Exp}(\delta\boldsymbol{\phi}) \otimes \bar{\mathbf{q}} \quad (3.29)$$

where \otimes refers to the standard quaternion product operator [Sola \(2017\)](#).

3.6 Deep Learning

Three of the four pseudo-sensors (Sun-BCNN, DPC, and HydraNet—Chapters 5 to 7) in this dissertation are built using neural networks and the tools of *deep learning* ([LeCun et al., 2015](#)). We briefly summarize these concepts here and refer the reader to [Goodfellow et al. \(2016\)](#) for a more thorough treatment.

3.6.1 Feed-forward Neural Networks

The basic premise of deep learning is that certain kinds of data are naturally decomposed into hierarchical structure. For instance, images may have local textures (e.g. edges), which compose basic primitives (e.g., leaves), which then combine to form semantic objects (e.g, a tree). Consequently, for this type of data, latent information may be efficiently extracted through an analogous hierarchical model. Typically, this is done through a neural network composed of multiple *layers* (the term *neural* comes from the loose biological basis for each layer, and *deep* refers to the amount of layers contained in state-of-the-art models). A standard neural layer computes a non-linear transformation from an input \mathbf{z}_ℓ to an output $\mathbf{z}_{\ell+1}$ as

$$\mathbf{z}_{\ell+1} = \mathbf{f}_\ell(\mathbf{z}) = \sigma(\mathbf{W}_\ell \mathbf{z}_\ell + \mathbf{b}_\ell), \quad (3.30)$$

where $\mathbf{W}_\ell \in \mathbb{R}^{D_{\ell_{out}} \times D_{\ell_{in}}}$ and $\mathbf{b}_\ell \in \mathbb{R}^{D_{\ell_{out}}}$ are the parameters of the layer (often referred to as the *weight matrix* and *bias*, respectively) and $\sigma(\cdot)$ is an element-wise *non-linearity*.

Remark (Non-linearities). The optimal choice of non-linearity is an area of active research. Some common examples include,

$$\sigma_i(x) = \begin{cases} \frac{1}{1+e^{-x}} & \text{Logistic,} \\ \frac{e^x - e^{-x}}{e^x + e^{-x}} & \text{Hyperbolic tangent, } \tanh, \\ \max(x, 0) & \text{Rectified Linear Unit, } \textit{ReLU}, \\ \begin{cases} x & \text{if } x \geq 0 \\ -\alpha x & \text{if } x < 0 \end{cases} & \text{Parametric ReLU, } \textit{PReLU}. \end{cases} \quad (3.31)$$

To produce a *feed-forward neural network*, L layers are composed in a hierarchy to create a single parametric function,

$$NN(\mathbf{x}) = \mathbf{f}_L \circ \mathbf{f}_{L-1} \circ \cdots \circ \mathbf{f}_1(\mathbf{x}), \quad (3.32)$$

where $f \circ g$ refers to function composition, $f(g(\cdot))$.

3.6.2 Convolutional Neural Networks

Convolutional Neural Networks ([LeCun et al., 1989](#)), or CNNs, are a particular form of neural network where at least one of the layers uses *convolution* in place of matrix multiplication. Due to their efficiency in processing high-dimensional image data, they have become ubiquitous in computer vision applications ([LeCun et al., 2015](#)). Briefly, a convolution is an operation that transforms a continuous input signal $x(t)$ into a new signal $y(t)$ as

$$y(t) = \int x(s)k(t-s)ds, \quad (3.33)$$

where k is often called the *kernel* of the convolution. For two-dimensional discrete input signals, convolution is defined as

$$y(i, j) = \sum_p \sum_q I(i-p, j-q)K(p, q). \quad (3.34)$$

Remark (Convolution vs. Cross-correlation). Most deep learning libraries implement convolution as the cross-correlation operation

$$y(i, j) = \sum_p \sum_q I(i+p, j+q)K(p, q) \quad (3.35)$$

which is convolution with the kernel ‘flipped.’ Unlike convolution, cross-correlation is not commutative with respect to the input and kernel (but this is typically not important in deep learning contexts). We note that kernels learned with cross-correlation will be flipped relative to those learned with true convolution ([Goodfellow et al., 2016](#)).

In this discrete case, a kernel can be represented by a kernel matrix, \mathbf{K} , which can have different size (e.g., $\mathbf{K} \in \mathbb{R}^{3 \times 3}$), and we may compute Equation (3.34) at different spatial intervals

called *strides*. To ensure that the convolution is defined at the boundaries of the signal, one can also use different *padding* strategies. Please refer to [Goodfellow et al. \(2016\)](#) for further information about these hyper-parameters.

By limiting the size of a kernel, we can use convolutional layers (parametrized by \mathbf{K}) to efficiently process high-dimensional signals. For images, a standard feed-forward layer would require a scalar weight for every pixel. In contrast, a convolutional layer can rely on a single kernel (with significantly less parameters than a weight matrix) to process an entire image with *shared weights*. This reduces the number of parameters of our network, but limits the type of spatial correlations we can model (since kernels can only capture local relations). To increase the potential modelling capacity of our network, we can use several independent kernels to process and output multiple *channels* (Figure 3.4). Finally, it is important to note that the convolution operator is particularly suited to visual data, as kernels can act as *filters* that pick out salient visual features like corners and edges.

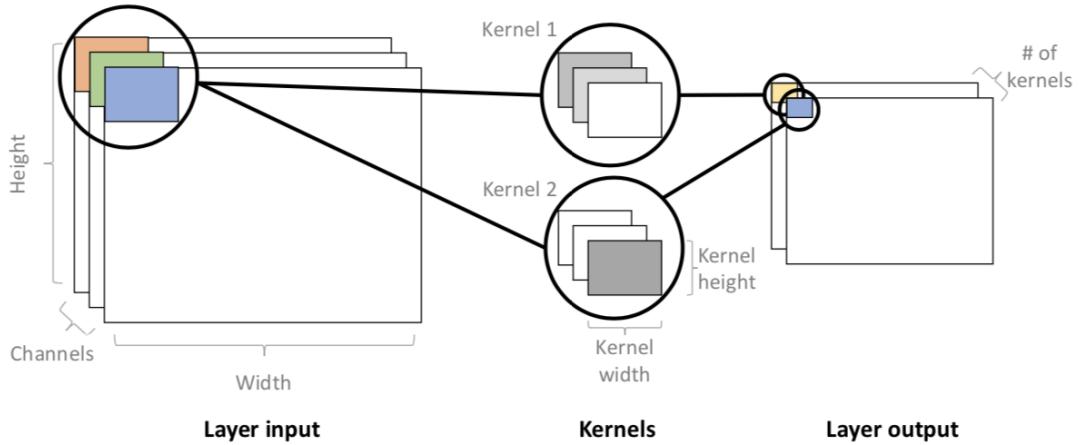


Figure 3.4: A convolutional layer with two kernels that operate over different channels of a two-dimensional input. Figure from [Gal \(2016\)](#).

3.6.3 Supervised Training

Given a (convolutional) neural network with a set of parameters π (where π may include weight and bias parameters, $\mathbf{W}_\ell, \mathbf{b}_\ell$, as well as convolutional kernel parameters, $\mathbf{K}_{k\ell}$) we can use *supervised training* to obtain an optimal parameter set π^* . Supervised training requires set of training pairs $\{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_N, \mathbf{y}_N)\}$ where $\mathbf{x} \in \mathbb{R}^{D_x}$ is an *input* and $\mathbf{y} \in \mathbb{R}^{D_y}$ is a *target* that corresponds to the desired output of our parametric model. To *train* the network,

we find the parameters which minimize a *loss function* over this dataset,

$$\boldsymbol{\pi}^* = \operatorname{argmin}_{\boldsymbol{\pi}} \mathcal{L}(\{NN(\mathbf{x}_i; \boldsymbol{\pi}), \mathbf{y}_i\}_{i=1}^N). \quad (3.36)$$

For example, a common loss function for regression problems is *mean squared error*,

$$\boldsymbol{\pi}^* = \operatorname{argmin}_{\boldsymbol{\pi}} \frac{1}{2N} \sum_{i=1}^N \|NN(\mathbf{x}_i; \boldsymbol{\pi}) - \mathbf{y}_i\|_2^2. \quad (3.37)$$

. In this dissertation, we will use supervised training in Chapters 5 to 7 and we will explore different forms of loss functions for geometric quantities that do not allow for a simple Euclidian norms.

3.6.4 Practical Considerations

Optimization

To train a deep network—that is, to solve Equation (3.36) for optimal parameters—modern approaches rely on stochastic gradient descent (SGD) through back-propagation (LeCun et al., 2015). SGD avoids the computationally prohibitive task of computing a gradient over an entire dataset by approximating the gradient using a randomly selecting subset of training data called a *mini-batch*. In the majority of this dissertation, we rely on Adam, a modern SGD-based approach that also maintains an estimate of second-order curvature (Kingma and Ba, 2017).

Regularization

In order to prevent *overfitting* to a dataset (i.e., to obtain a model that generalizes to unseen data), the literature provides a number of methods. An important approach that is used in this dissertation is *dropout* (Srivastava et al., 2014). Dropout stochastically ‘zeros-out’ inputs of a particular layer with probability p ,

$$\mathbf{z}_{\ell+1} = \mathbf{f}_{\ell}(\mathbf{z}) = \sigma(\mathbf{W}_{\ell}\tilde{\mathbf{z}}_{\ell} + \mathbf{b}_{\ell}), \quad (3.38)$$

where $\tilde{\mathbf{z}}_{\ell} = \mathbf{b} \odot \mathbf{z}_{\ell}$, $b_i \sim \text{Bernoulli}(p)$ and \odot refers to element-wise multiplication. The intuition behind dropout is that it effectively creates an ensemble of smaller-parameter networks that will generally be less prone to overfitting than a commensurately-sized monolithic network. Importantly, dropout has a fundamental connection variational inference, which we exploit in Chapter 5.

Pooling and Spatial Invariance

A common technique in convolutional neural networks is *pooling* (Goodfellow et al., 2016). Pooling is a non-parametric operation that summarizes the output of a kernel in a particular region. For example, *max pooling* selects the maximum response of a kernel in demarcated regions of an input channel, thereby downsampling the resulting output. Pooling operations are designed to make convolution operations invariant to the spatial location of a particular kernel response. This is important in many classification tasks (e.g., a tree classifier should be invariant to the location of a particular leaf) but is a detriment to regression tasks where we would like to preserve spatial information. We explore building convolutional neural networks without pooling in Chapter 6.

Appendices

Bibliography

- Agarwal, S., Mierle, K., et al. (2016). Ceres solver.
- Albertz, J. (2007). A look back. *Photogrammetric Engineering & Remote Sensing*, 73(5):504–506.
- 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.
- Brachmann, E. and Rother, C. (2018). Learning less is more-6d camera localization via 3d surface regression. In *Proc. CVPR*, volume 8.
- Bruss, A. R. and Horn, B. K. (1983). Passive Navigation. *Computer Vision, Graphics, and Image Processing*, 21(1):3–20.
- Byravan, A. and Fox, D. (2017). SE3-nets: Learning rigid body motion using deep neural networks. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 173–180.
- 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.
- Carlone, L., Rosen, D. M., Calafiore, G., Leonard, J. J., and Dellaert, F. (2015a). Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 125–132.

- Carbone, L., Tron, R., Daniilidis, K., and Dellaert, F. (2015b). Initialization techniques for 3D SLAM: A survey on rotation estimation and its use in pose graph optimization. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4597–4604.
- Cheng, Y., Maimone, M. W., and Matthies, L. (2006). Visual odometry on the mars exploration rovers - a tool to ensure accurate driving and science imaging. *IEEE Robot. Automat. Mag.*, 13(2):54–62.
- Clark, R., Wang, S., Wen, H., Markham, A., and Trigoni, N. (2017). Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem.
- Clement, L. and Kelly, J. (2018). How to train a CAT: learning canonical appearance transformations for direct visual localization under illumination change. *IEEE Robotics and Automation Letters*, 3(3):2447–2454.
- 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.
- Costante, G., Mancini, M., Valigi, P., and Ciarfuglia, T. A. (2016). Exploring representation learning with CNNs for Frame-to-Frame Ego-Motion estimation. *IEEE Robotics and Automation Letters*, 1(1):18–25.
- 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.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition, (CVPR)*, pages 248–255.
- DeTone, D., Malisiewicz, T., and Rabinovich, A. (2016). Deep image homography estimation.

- Duan, Y., Chen, X., Houthooft, R., Schulman, J., and Abbeel, P. (2016). Benchmarking deep reinforcement learning for continuous control. In *Proc. Int. Conf. on Machine Learning, ICML'16*, pages 1329–1338.
- Eisenman, A. R., Liebe, C. C., and Perez, R. (2002). Sun sensing on the mars exploration rovers. In *Aerospace Conf. Proc.*, volume 5, pages 5–2249–5–2262 vol.5. IEEE.
- Engel, J., Koltun, V., and Cremers, D. (2018). Direct Sparse Odometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):611–625.
- Engel, J., Stuckler, J., and Cremers, D. (2015). Large-scale direct SLAM with stereo cameras. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*, pages 1935–1942.
- Farnebäck, G. (2003). Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer.
- Fischler, M. and Bolles, R. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6):381–395.
- Fisher, R. (1953). Dispersion on a sphere. In *Proc. Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 217, pages 295–305. The Royal Society.
- 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., Carlone, L., Dellaert, F., and Scaramuzza, D. (2015). IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation.
- 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. IEEE.
- Furgale, P. (2011). *Extensions to the Visual Odometry Pipeline for the Exploration of Planetary Surfaces*. PhD thesis.
- Furgale, P. and Barfoot, T. D. (2010). Visual teach and repeat for long-range rover autonomy. *J. Field Robot.*, 27(5):534–560.

- 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., Enright, J., and Barfoot, T. (2011). Sun sensor navigation for planetary rovers: Theory and field testing. *IEEE Trans. Aerosp. Electron. Syst.*, 47(3):1631–1647.
- 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. (2016). *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge.
- Gal, Y. and Ghahramani, Z. (2016a). Bayesian convolutional neural networks with Bernoulli approximate variational inference. In *Proc. Int. Conf. Learning Representations (ICLR), Workshop Track*.
- Gal, Y. and Ghahramani, Z. (2016b). 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., Ziegler, J., and Stiller, C. (2011). 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.
- Glocker, B., Izadi, S., Shotton, J., and Criminisi, A. (2013). Real-time rgbd camera relocalization. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 173–179.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- 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.

- Haarnoja, T., Ajay, A., Levine, S., and Abbeel, P. (2016). Backprop KF: Learning discriminative deterministic state estimators. In *Proc. Advances in Neural Inform. Process. Syst. (NIPS)*.
- 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.
- Hartley, R., Trumpf, J., Dai, Y., and Li, H. (2013). Rotation averaging. *Int. J. Comput. Vis.*, 103(3):267–305.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Holland, P. W. and Welsch, R. E. (1977). Robust regression using iteratively reweighted least-squares. *Communications in Statistics - Theory and Methods*, 6(9):813–827.
- Hu, H. and Kantor, G. (2015). Parametric covariance prediction for heteroscedastic noise. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*, pages 3052–3057.
- Huber, P. J. (1964). Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, pages 73–101.
- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. In *Proc. ACM Int. Conf. Multimedia (MM)*, pages 675–678.
- Kelly, J., Saripalli, S., and Sukhatme, G. S. (2008). Combined visual and inertial navigation for an unmanned aerial vehicle. In *Proc. Field and Service Robot. (FSR)*, pages 255–264.
- 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.
- Kendall, A. and Cipolla, R. (2017). Geometric loss functions for camera pose regression with deep learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6555–6564.
- Kendall, A., Grimes, M., and Cipolla, R. (2015). PoseNet: A convolutional network for Real-Time 6-DOF camera relocalization. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2938–2946.
- 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.

- Kingma, D. P. and Ba, J. (2017). Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*. arXiv: 1412.6980.
- 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.
- Lalonde, J.-F., Efros, A. A., and Narasimhan, S. G. (2011). Estimating the natural illumination conditions from a single outdoor image. *Int. J. Comput. Vis.*, 98(2):123–145.
- 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.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4):541–551.
- Lee, S., Purushwalkam, S., Cogswell, M., Crandall, D., and Batra, D. (2015). Why M heads are better than one: Training a diverse ensemble of deep networks.
- 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.
- Levine, S., Finn, C., Darrell, T., and Abbeel, P. (2016). End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.*
- Li, Q., Qian, J., Zhu, Z., Bao, X., Helwa, M. K., and Schoellig, A. P. (2017a). Deep neural networks for improved, impromptu trajectory tracking of quadrotors. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 5183–5189.
- Li, R., Wang, S., Long, Z., and Gu, D. (2017b). UnDeepVO: Monocular visual odometry through unsupervised deep learning.
- 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.

- Ma, W.-C., Wang, S., Brubaker, M. A., Fidler, S., and Urtasun, R. (2016). Find your way by observing the sun and other semantic cues.
- 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.
- Maddern, W., Pascoe, G., Linegar, C., and Newman, P. (2016). 1 year, 1000 km: The oxford RobotCar dataset. *Int. J. Rob. Res.*
- 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.
- McManus, C., Upcroft, B., and Newman, P. (2014). Scene signatures: Localised and point-less features for localisation. In *Proc. Robotics: Science and Systems X*.
- 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.
- Oliveira, G. L., Radwan, N., Burgard, W., and Brox, T. (2017). Topometric localization with deep learning. *arXiv preprint arXiv:1706.08775*.
- Olson, C. F., Matthies, L. H., Schoppers, M., and Maimone, M. W. (2003). Rover navigation using stereo ego-motion. *Robot. Auton. Syst.*, 43(4):215–229.
- 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*, 37(9):996–1016.
- Peretroukhin, V. and Kelly, J. (2018). DPC-Net: Deep pose correction for visual localization. *IEEE Robotics and Automation Letters*, 3(3):2424–2431.
- Peretroukhin, V., Kelly, J., and Barfoot, T. D. (2014). Optimizing camera perspective for stereo visual odometry. In *Canadian Conference on Comp. and Robot Vision*, pages 1–7.
- 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 $\text{SO}(3)$ 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.
- Punjani, A. and Abbeel, P. (2015). Deep learning helicopter dynamics models. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 3223–3230.
- 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.
- Schonberger, J. L., Hardmeier, H., Sattler, T., and Pollefeys, M. (2017). Comparative Evaluation of Hand-Crafted and Learned Local Features. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6959–6968, Honolulu, HI. IEEE.
- Sibley, G., Matthies, L., and Sukhatme, G. (2007). Bias reduction and filter convergence for long range stereo. In *Robotics Research*, pages 285–294. Springer Berlin Heidelberg.

- 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.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958.
- 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.
- Sunderhauf, N., Shirazi, S., Jacobson, A., Dayoub, F., Pepperell, E., Upcroft, B., and Milford, M. (2015). Place recognition with ConvNet landmarks: Viewpoint-robust, condition-robust, training-free. In *Proc. Robotics: Science and Systems XII*.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proc. IEEE Conf. Comput. Vision and Pattern Recognition, (CVPR)*, pages 1–9.
- Triggs, B., McLauchlan, P. F., Hartley, R. I., and Fitzgibbon, A. W. (2000). Bundle Adjustment – A Modern Synthesis. In Goos, G., Hartmanis, J., van Leeuwen, J., Triggs, B., Zisserman, A., and Szeliski, R., editors, *Vision Algorithms: Theory and Practice*, volume 1883, pages 298–372. Springer Berlin Heidelberg, Berlin, Heidelberg.
- 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. 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.
- Wang, R., Schworer, M., and Cremers, D. (2017a). Stereo DSO: Large-Scale Direct Sparse Visual Odometry with Stereo Cameras. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3923–3931, Venice. IEEE.

- Wang, S., Clark, R., Wen, H., and Trigoni, N. (2017b). DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2043–2050.
- Warren, R. (1976). The perception of egomotion. *Journal of Experimental Psychology: Human Perception and Performance*, 2(3):448.
- Yang, F., Choi, W., and Lin, Y. (2016). Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *Proc. IEEE Int. Conf. Comp. Vision and Pattern Recognition (CVPR)*, pages 2129–2137.
- Yang, N., Wang, R., Stueckler, J., and Cremers, D. (2018). Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *European Conference on Computer Vision (ECCV)*. accepted as oral presentation, arXiv 1807.02570.
- 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?
- Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., and Oliva, A. (2014). Learning deep features for scene recognition using places database. In *Advances in Neural Inform. Process. Syst. (NIPS)*, pages 487–495.
- Zhou, T., Brown, M., Snavely, N., and Lowe, D. G. (2017). Unsupervised learning of depth and Ego-Motion from video. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6612–6619.