Learned Improvements to the Visual Egomotion Pipeline

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

Learned Improvements to the Visual Egomotion Pipeline

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 four ways in which conventional pipelines can be improved through the addition of a learned hyper-parametric model. By combining traditional pipelines with learning, 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 by extracting latent information in existing visual data. We demonstrate the improvements derived from 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 supervisor[1], the perennial love of my family and friends[2], and the limitless patience of my lab mates[3]. Thank you all.

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

Contents

Notation

a	:	Symbols in this font are real scalars.
$\mathbf{a}$	:	Symbols in this font are real column vectors.
$\boldsymbol{a}$	:	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.
$\underrightarrow{\boldsymbol{\mathcal{F}}}_a$	:	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.
$\mathbf{p}_a^{cb}$	:	A vector from point b to point c (denoted by the superscript) and expressed in $\underrightarrow{\boldsymbol{\mathcal{F}}}_a$ (denoted by the subscript).
$\mathbf{C}_{ab}$	:	The 3×3 rotation matrix that transforms vectors from $\underrightarrow{\boldsymbol{\mathcal{F}}}_b$ to $\underrightarrow{\boldsymbol{\mathcal{F}}}_a$: $\mathbf{p}_a^{cb} = \mathbf{C}_{ab}\mathbf{p}_b^{cb}$.
$\mathbf{T}_{ba}$	:	The 4×4 transformation matrix that transforms homogeneous points from $\underrightarrow{\boldsymbol{\mathcal{F}}}_a$ to $\underrightarrow{\boldsymbol{\mathcal{F}}}_b$: $\boldsymbol{p}_b^{cb} = \mathbf{T}_{ba}\boldsymbol{p}_a^{ca}$.

Chapter 7

Learning Rotation with Uncertainty

> Anyone who has ever used any other parametrization of the rotation group will, within hours of taking up the quaternion parametrization, lament his or her misspent youth.
>
> Simon Altmann

Finally, building on the lessons of Sun-BCNN (Chapter 5) and DPC-Net (Chapter 6), we focus on extracting estimates of camera rotation from visual data. To facilitate fusion with motion estimates from an existing egomotion pipeline, we develop a network structure and loss to extract consistent estimates of three degree-of-freedom uncertainty alongside rotation predictions. To do this, we develop a network structure we call *HydraNet* that can account for both **epistemic and aleatoric** sources of uncertainty and adapt it to the problem of estimating elements of $\mathrm{SO}(3)$.

Remark (*Associated Publications*). This work is associated with the 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.

In this chapter we elaborate on the formulation and experimental validation presented within that publication.

7.1 Motivation

Accounting for position and orientation, or pose, is at the heart of computer vision. Many algorithms in image classification and feature tracking, for example, are explicitly concerned with output that

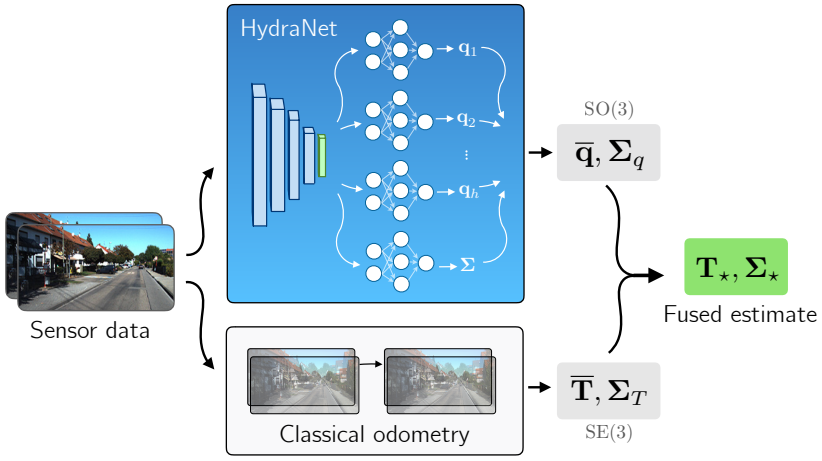


Figure 7.1: HydraNet produces an estimate of relative rotation (with a principled covariance matrix) that can be fused with existing egomotion pipelines through pose graph optimization.

is robust to camera orientation. Conversely, visual odometry, structure from motion, and SLAM use visual sensors to estimate and track the pose of a camera as it moves through some environment.

As discussed in the previous chapter, several recent authors (Clark et al., 2017; Melekhov et al., 2017; Kendall et al., 2015) have attempted to transfer the success of deep neural networks in many areas of computer vision to the task of camera pose estimation. These approaches, however, can produce arbitrarily poor pose estimates if sensor data differs from what is observed during training (i.e., it is 'out of training distribution') and their monolithic nature makes them difficult to debug. Further, as we have pointed out in previous chapters, despite much research effort, classical motion estimation algorithms, like indirect stereo visual odometry, still achieve state-of-the-art performance in nominal conditions. Nevertheless, the representational power of deep regression algorithms makes them an attractive option to complement classical motion estimation when these latter methods perform poorly (e.g., under diverse lighting conditions or low scene texture). By endowing deep regression models with a useful notion of uncertainty, we can account for out-of-training-distribution errors and fuse these models with classical methods using probabilistic factor graphs. In this work, we choose to focus on rotation regression, since many motion algorithms are sensitive to rotation errors (Peretroukhin et al., 2018; Olson et al., 2003), and good rotation initializations can be critical to robust optimization.

7.2 Related work

Much recent work in the literature has been devoted to replacing classical localization algorithms with deep network equivalents. Some approaches (Clark et al., 2017; Kendall et al., 2015; Kendall and Cipolla, 2017; Melekhov et al., 2017) learn poses directly, while others learn them indirectly as the spatial transforms that result in minimal loss defined over some other domain (e.g., pixel or depth space) (Byravan and Fox, 2017; Handa et al., 2016).

Despite this surge of research in neural-network-based replacements, some authors have nevertheless used deep networks to augment classical state estimation algorithms. Deep networks have been

trained as pose correctors whose corrections can be fused with existing estimates through pose graph relaxation (Chapter 6), and as depth prediction networks that can be incorporated into a classical monocular pipelines to provide an initial estimate for metric scale (Yang et al., 2018). The pseudo-sensor we present in this chapter is perhaps closest in spirit to (Haarnoja et al., 2016) which fuses deep probabilistic observation functions with classical models using a Kalman Filter, but focuses on unconstrained targets and does not investigate uncertainty quantification on manifolds.

In the robotics community, there has been significant effort to leverage the tools of matrix Lie groups to handle poses and associated uncertainty (Solà et al., 2018; Barfoot and Furgale, 2014). In parallel, the computer vision community has developed a rich literature of rotation averaging (Hartley et al., 2013) which focuses on principled ways to combine elements of $\mathrm{SO}(3)$ based on different metrics defined over the group.

Finally, uncertainty in the context of deep learning has been investigated through the technique of MC Dropout (Chapter 5, Gal (2016), Kendall and Gal (2017)). Concurrently, *ensembles of networks* have been shown to be a scalable way to extract uncertainty for deep regression and classification (Lakshminarayanan et al., 2017), while multi-headed networks have been proposed in the context of ensemble learning (Lee et al., 2015) and for bootstrapped uncertainty in reinforcement learning (Osband et al., 2016). Finally, an an alternate binary approach (Richter and Roy, 2017) to dealing with uncertainty is to classify test samples as either *in training distribution* (i.e., cases where our model should have high accuracy) and *out of training distribution* (i.e., indeterminate cases where we may revert to an alternate prediction schema). This latter binary classification can be thought of as a thresholded epistemic uncertainty, and we believe, can be obviated through good uncertainty quantification.

7.3 Approach

We develop our method for probabilistic $\mathrm{SO}(3)$ regression in three steps. First, we motivate why learning elements of $\mathrm{SO}(3)$ is particularly germane to the task of egomotion estimation. Second, we present a multi-headed network that can regress unconstrained targets and produce consistent uncertainty estimates. Toward this end, we present a one-dimensional regression experiment, validating prior works (Lakshminarayanan et al., 2017; Osband et al., 2016) that suggest a bootstrap-inspired approach provides better calibrated uncertainties than one based on stochastic sampling through MC dropout and can be straightforwardly extended to incorporate both aleatoric and epistemic uncertainty. Finally, we generalize these results to targets that belong to $\mathrm{SO}(3)$ by defining a rotation average using the quaternionic metric, and show how we can compute anisotropic uncertainty on four-dimensional unit quaternions.

7.3.1 Why Rotations?

We focus our attention on learning rotations for three primary reasons. First, rotations can be learned without reference to scale, using monocular images without the need for metric depth estimation. These images can come from cheap, light-weight imaging sensors that can be found on many ground

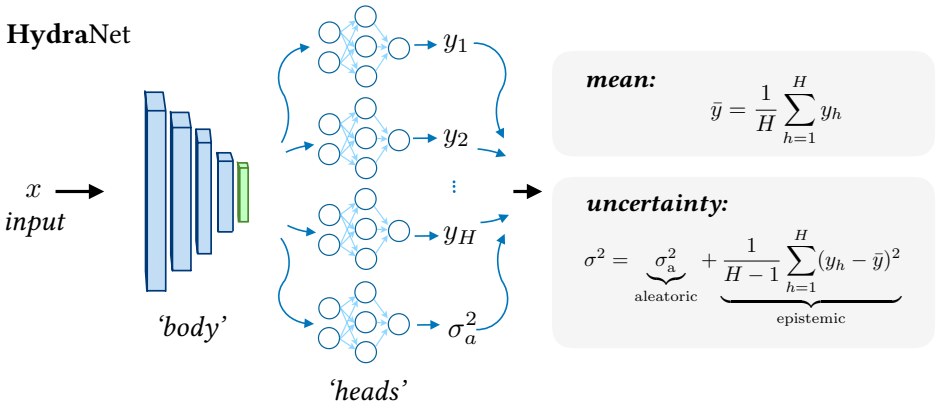


Figure 7.2: The HydraNet structure. Input data is passed through a main body and then through a number of heads. Outputs are combined to produce an average and an uncertainty.

and aerial vehicles. Furthermore, many depth-equipped sensors like stereo cameras and RGB-D cameras have limited depth range and produce poor depth estimates in large-scale outdoor environments. Second, many egomotion estimation techniques, like visual odometry or visual SLAM, are particularly sensitive to rotation estimates as small early errors have a large influence on final pose estimates (Olson et al., 2003). Finally, the constrained nature of rotations presents several difficulties for optimization algorithms. Indeed, if rotations are known, the general problem of pose graph relaxation becomes a linear least squares problem that can be solved with no initial guess for translations (Carlone et al., 2015b).

7.3.2 Probabilistic Regression and HydraNet

To begin, we will consider a (non-Bayesian) approach to uncertainty quantification. Consider the one dimensional regression task where, given an input $x \in \mathbb{R}$, with a target output $y_t \in \mathbb{R}$, we desire a probabilistic estimate $\{\overline{y}, \sigma^2\}$ such that σ^2 maximizes a likelihood model of our prediction $\overline{y}$ given that we know y_t.

HydraNet

One possible way to obtain $\overline{y}$ is to train a deep neural network, $g(x)$. To endow this network with uncertainty, we present a network structure we call HydraNet (see Figure 7.2). HydraNet is composed of a large, main 'body', $b(x; \boldsymbol{\pi}_b) = NN(x; \boldsymbol{\pi}_b)$ with $H+1$ heads, $h_i(x; \boldsymbol{\pi}_{h_i}) = NN(x; \boldsymbol{\pi}_{h_i})$, attached to the output of the body. Given an input x, we get $H+1$ outputs as:

$$\{y_1, ..., y_H, \sigma_a^2\} = \{h_1 \circ b(x), h_2 \circ b(x), ..., h_{H+1} \circ b(x)\} \tag{7.1}$$

where $\circ$ denotes function composition. To compute $\overline{y}$, we compute the arithmetic mean of the outputs,

$$\overline{y} = \frac{1}{H}\sum_{h=1}^{H} y_h(x). \tag{7.2}$$

The head structure, however, provides several key advantages toward the goal of estimating consistent uncertainty. Namely, it allows us to define the overall uncertainty in terms of two sources, *epistemic* (σ_e) and *aleatoric* (σ_a) (Kendall and Gal, 2017):

$$\sigma^2 = \underbrace{\sigma_{\text{e}}^2}_{\text{epistemic}} + \underbrace{\sigma_{\text{a}}^2}_{\text{aleatoric}} . \tag{7.3}$$

Remark (*Epistemic and Aleatoric Uncertainty*). The former, σ_e, is also sometimes referred to as model uncertainty; it is a measure of how close a particular test sample is to known training samples. The latter, σ_a, is inherent to the observation of the target itself. Even if the model can localize a test sample exactly in some salient input space, the aleatoric uncertainty will prevent exact regression due to physical processes like sensor noise.

To account for aleatoric uncertainty, we follow prior work (Haarnoja et al., 2016; Lakshminarayanan et al., 2017) and dedicate one head of the network to regressing a variance directly through a negative log likelihood loss under the assumption of Gaussian likelihood. That is, we define a supervised loss,

$$\boldsymbol{\pi}_{h_i}^*, \boldsymbol{\pi}_b^* = \underset{\boldsymbol{\pi}_i, \boldsymbol{\pi}_b}{\operatorname{argmin}} \mathcal{L}(y_h, \sigma_a^2, y_t) = \underset{\boldsymbol{\pi}}{\operatorname{argmin}} \frac{1}{2\sigma_a^2}(y - y_t)^2 + \log(\sigma_a^2), \tag{7.4}$$

where y_t is a target output.

To capture epistemic uncertainty, we train each head with random weight initializations and apply losses independently during training. During test time, we compute a sample covariance over the different outputs. That is, at test time, we compute:

$$\sigma_e^2 = \frac{1}{H-1}\sum_{h=1}^{H}(y_h - \bar{y})^2 \tag{7.5}$$

This approach is inspired by the method of the statistical bootstrap (Osband et al., 2016), which predicts population statistics by computing statistics over subsets of a sample chosen with replacement. Unlike Osband et al. (2016), we do not train each head of the network with a bootstrapped sample, but instead rely on the random initializations of their parameters and the method of dropout to introduce sufficient stochasticity into their outputs. Further, unlike Lakshminarayanan et al. (2017), we do not require numerous trained models that can incur high computational cost for complex regression tasks.

One-dimensional experiment

To build intuition for the advantages of HydraNet over other methods of extracting uncertainty (e.g., uncertainty through MC dropout (Gal and Ghahramani, 2016b)), we constructed an experiment similar to that presented in (Osband et al., 2016). We compared HydraNet to four other approaches: (1) direct aleatoric variance regression where the network outputs a second variance parameter that is constrained to be positive, (2) uncertainty through dropout at test time (Gal and Ghahramani, 2016b),

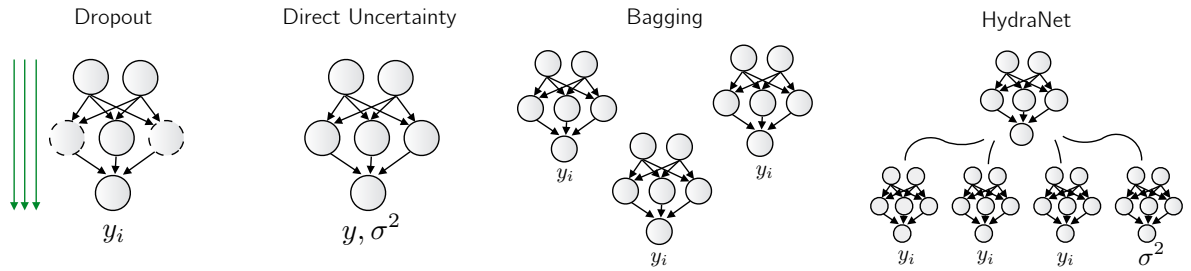


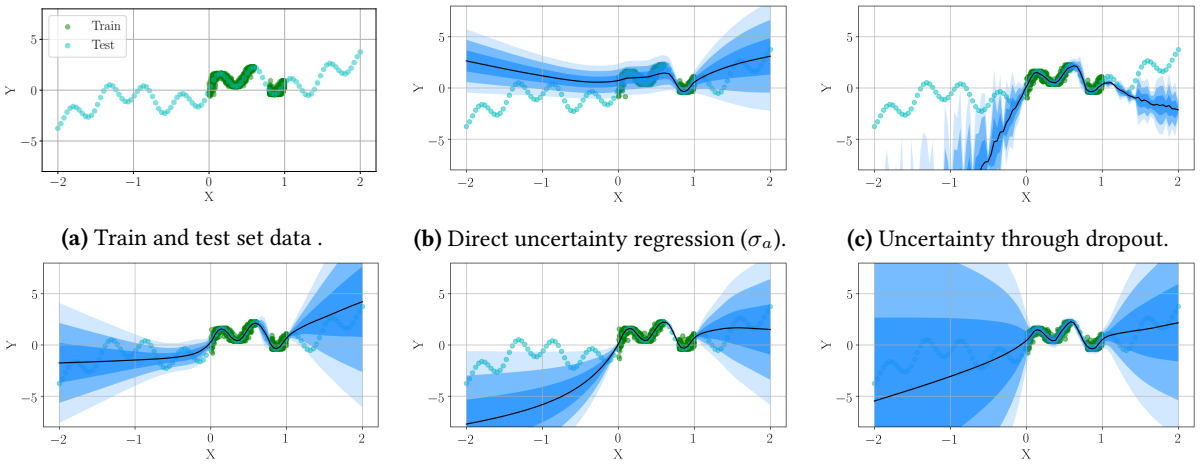
Figure 7.3: Different scalable approaches to neural network uncertainty.

(3) bootstrap aggregation (or bagging) of multiple independent models, and (4) HydraNet with no aleatoric uncertainty output.

For each method, we trained a four-layer fully-connected network to regress the output of a one-dimensional function:

$$y_i = x_i + \sin\left(4(x_i + \omega)\right) + \sin\left(13(x_i + \omega)\right) + \omega, \tag{7.6}$$

where $w \sim \mathcal{N}(\mu = 0, \sigma^2 = 3^2)$. Our training set consisted of 1000 samples randomly drawn from $x \in [0.0, 0.6] \bigcup [0.8, 1.0]$, while the test set consisted of 100 samples uniformly drawn from $x \in [-2, 2]$. The function and the train/test samples are shown in Figure 7.4a.



(a) Train and test set data .

(b) Direct uncertainty regression (σ_a).

(c) Uncertainty through dropout.

(d) Bootstrap aggregation.

(e) HydraNet (no aleatoric uncertainty).

(f) HydraNet.

Figure 7.4: A comparison of different ways to extract uncertainty from deep networks. Each shade of blue represents one standard deviation σ produced by the model.

The direct aleatoric uncertainty regression and HydraNet methods were trained using a negative log likelihood loss under the assumption of Gaussian likelihood, while the other methods were trained to minimize mean squared error. We repeated training 100 times, and recorded the test-time negative log likelihood for each method at each repetition. We summarize the results in Figure 7.5. Figure 7.4 presents representative samples from the 100 repetitions for each method. Typically, direct uncertainty regression and dropout are overconfident in the out-of-distribution regions. We replicated the findings

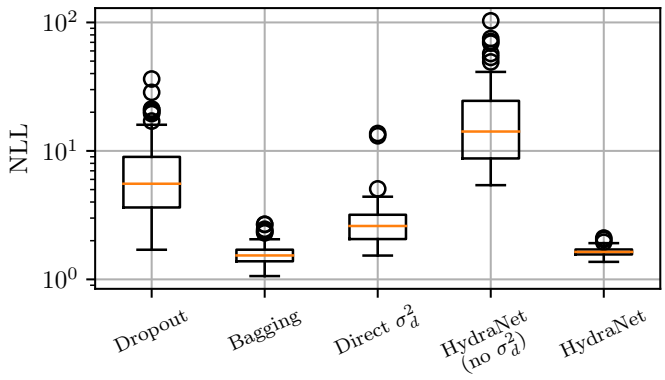


Figure 7.5: Negative log likelihood statistics of 100 repetitions of five neural-network-based uncertainty estimators. HydraNet performs similarly to bagging.

of (Osband et al., 2016) who find that uncertainty with dropout does not vary smoothly and can collapse outside of the training distribution. HydraNet combined with direct aleatoric uncertainty learning, however, produced similar excellent likelihoods to bootstrap aggregation without requiring multiple models.

7.3.3 Extending HydraNet to $\mathrm{SO}(3)$

In order to extend the ideas of HydraNet to the matrix Lie group SO(3), we consider different ways to regress and combine several estimates of rotation. Given a network, $g(\cdot)$, and an input $\mathcal{I}$, we consider how to extend the ideas of HydraNet to process several outputs, $g_i(\mathcal{I})$, and combine them into an estimate of a 'mean' rotation, $\overline{\mathbf{R}}$, and an associated 3×3 covariance matrix, $\mathbf{\Sigma}$. To produce estimates of rotation for a given HydraNet head, we consider two options. First if $g(\mathcal{I}) \in \mathbb{R}^3$, then we can use the matrix exponential to produce a rotation matrix,

$$\mathbf{R} = \mathrm{Exp}\left(g(\mathcal{I})\right). \tag{7.7}$$

Since the capitalized exponential map $\mathrm{Exp}\left(\cdot\right)$ is surjective (Barfoot, 2017; Solà et al., 2018), this approach can parametrize any valid rotation matrix. Alternatively, if $g(\mathcal{I}) \in \mathbb{R}^4$, we can normalize it to produce a unit quaternion that resides on S^3,

$$\mathbf{q} = \frac{g(\mathcal{I})}{\|g(\mathcal{I})\|}. \tag{7.8}$$

As we noted in Chapter 2, unit quaternions are a double cover of $\mathrm{SO}(3)$, and can represent any rotation. We choose to use this latter parametrization because of its simple analytic mean expression that we describe below.

Rotation Averaging

To produce a mean of several $\mathrm{SO}(3)$ elements (i.e., to evaluate Equation (7.2) for rotations), we turn to the field of rotation averaging (Hartley et al., 2013). Given several estimates of a rotation, we define the mean as the rotation which minimizes some squared metric defined over the group[1],

$$\overline{\mathbf{R}} = \underset{\mathbf{R}\in\mathrm{SO}(3)}{\operatorname{argmin}} \sum_{i=1}^{n} d(\mathbf{R}_i, \mathbf{R})^2. \tag{7.9}$$

There are three common choices for a bijective metric (Hartley et al., 2013; Carlone et al., 2015b) on $\mathrm{SO}(3)$. The angular, chordal and quaternionic:

$$d_{\mathrm{ang}}(\mathbf{R}_a, \mathbf{R}_b) = \left\| \mathrm{Log}\left(\mathbf{R}_a \mathbf{R}_b^{\mathsf{T}}\right) \right\|_2, \tag{7.10}$$

$$d_{\mathrm{chord}}(\mathbf{R}_a, \mathbf{R}_b) = \left\| \mathbf{R}_a - \mathbf{R}_b \right\|_{\mathrm{F}}, \tag{7.11}$$

$$d_{\mathrm{quat}}(\mathbf{q}_a, \mathbf{q}_b) = \min\left(\left\| \mathbf{q}_a - \mathbf{q}_b \right\|_2, \left\| \mathbf{q}_a + \mathbf{q}_b \right\|_2 \right), \tag{7.12}$$

where $\mathrm{Log}\,(\cdot)$, represents the capitalized matrix logarithm (Solà et al., 2018), and $\|\cdot\|_F$ the Frobenius norm. In the context of Equation (7.9), using the angular metric leads to the *Karcher mean*, which requires an iterative solver and has no known analytic expression. Applying the chordal metric leads to an analytic expression for the average but requires the use of Singular Value Decomposition. Using the quaternionic metric, however, leads to a simple, analytic expression for the rotation average as the normalized arithmetic mean of a set of unit quaternions (Hartley et al., 2013),

$$\overline{\mathbf{q}} = \underset{\mathbf{R}(\mathbf{q})\in\mathrm{SO}(3)}{\operatorname{argmin}} \sum_{i=1}^{H} d_{\mathrm{quat}}(\mathbf{q}_i, \mathbf{q})^2 = \frac{\sum_{i=1}^{H} \mathbf{q}_i}{\left\| \sum_{i=1}^{H} \mathbf{q}_i \right\|}. \tag{7.13}$$

This expression is simple to evaluate numerically, and if necessary, can be easily differentiated with respect to its constituent parts. For these reasons, we opt to construct our $\mathrm{SO}(3)$ HydraNet using unit quaternion outputs, and evaluate the rotation average using the quaternionic metric.

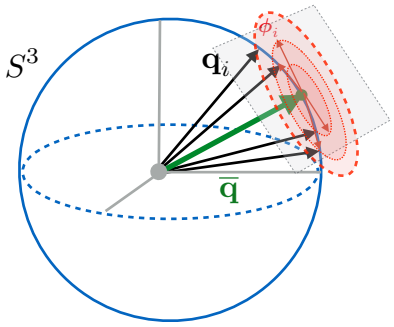


Figure 7.6: We can define uncertainty in the left tangent space of a mean element.

SO(3) Uncertainty

There are several ways to approach uncertainty on $\mathrm{SO}(3)$. One method (Carlone et al., 2015a) is to define a probability density directly on the group via the isotropic von Mises-Fisher density. This approach has two downsides: (1) it is isotropic and cannot account for dominant degrees of freedom (e.g., vehicle yaw

[1] Although this is a natural formulation for the rotation mean, it is possible to define other means in terms of absolute errors - see (Hartley et al., 2013).

during driving), and (2) estimating the concentration parameter
requires approximations or iterative solvers.

Instead, we opt to parametrize uncertainty over $\mathrm{SO}(3)$ by injecting uncertainty onto the manifold (Forster et al., 2015; Barfoot and Furgale, 2014; Barfoot, 2017) from a local tangent space about some mean element, $\overline{\mathbf{q}}$,

$$\mathbf{q} = \mathrm{Exp}\left(\boldsymbol{\epsilon}\right) \otimes \overline{\mathbf{q}}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}), \tag{7.14}$$

where $\otimes$ represents quaternion multiplication. In this formulation, $\boldsymbol{\Sigma}$ provides a 3×3 covariance matrix that can express uncertainty in different directions. Further, given a mean rotation, $\overline{\mathbf{q}}$, and samples, $\mathbf{q}_i$, we use the logarithmic map to compute a sample covariance matrix,

$$\boldsymbol{\Sigma}_e = \frac{1}{H-1} \sum_{i=1}^{H} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^{\mathsf{T}}, \quad \boldsymbol{\phi}_i = \mathrm{Log}\left(\mathbf{q}_i \otimes \overline{\mathbf{q}}^{-1}\right). \tag{7.15}$$

7.3.4 Loss Function

As with one-dimensional HydraNet, we train a direct regression of covariance through a parametrization of positive semi-definite matrices using a Cholesky decomposition[2] (Hu and Kantor, 2015; Haarnoja et al., 2016)). Given the network outputs of a unit quaternion $\mathbf{q}$, and a positive semi-definite matrix Σ, we define a loss function as the negative log likelihood of a given rotation under Equation (7.14) (see (Forster et al., 2015)) for a given target rotation, $\mathbf{q}_t$, as

$$\mathcal{L}_{\mathrm{NLL}}(\mathbf{q}, \mathbf{q}_t, \boldsymbol{\Sigma}_a) = \frac{1}{2}\boldsymbol{\phi}^{\mathsf{T}}\boldsymbol{\Sigma}_a^{-1}\boldsymbol{\phi} + \frac{1}{2}\log\det\left(\boldsymbol{\Sigma}_a\right), \tag{7.16}$$

where $\boldsymbol{\phi} = \mathrm{Log}\left(\mathbf{q} \otimes \mathbf{q}_t{}^{-1}\right)$. Combining the sample covariance, with the learned covariance, we extend Equation (7.3) to

$$\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_e + \boldsymbol{\Sigma}_a. \tag{7.17}$$

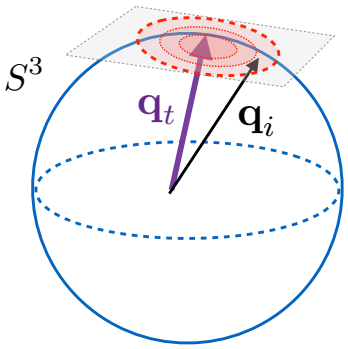
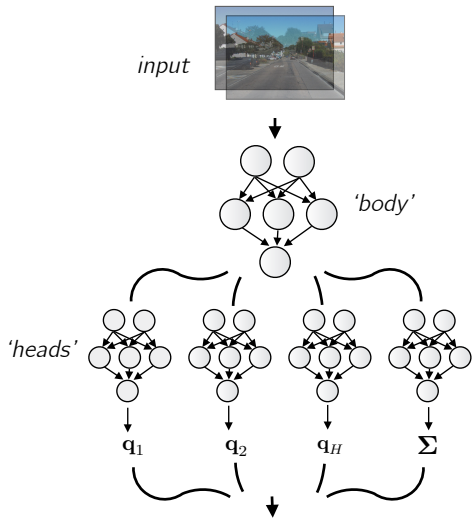


Figure 7.7: We define our negative log likelihood loss in the left tangent space of the target unit quaternion.

This covariance estimate is designed to grow for out-of-training-distribution errors (and account for *domain shift* (Lakshminarayanan et al., 2017)) while still accounting for uncertainty within the training set. We note that unlike Bayesian methods, we do not interpret each head as a *sample* from a posterior distribution[3]. Indeed, we note that in our 1D experiments, the heads have very small variance within the training distribution. The multi-headed structure and rotating averaging serves simply as a way to model epistemic uncertainty when the model encounters inputs that differ from those seen during training. We summarize our training and test procedures in Figure 7.8 as well as Algorithm 4 and

[2] Note that in all the experiments presented in this paper, we omit the off-diagonal components of this covariance and only learn a diagonal matrix with non-negative components.

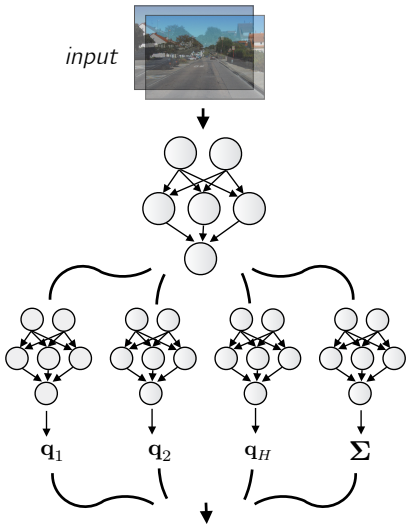
[3] Notably, this means we do not scale our direct uncertainty when averaging as $\frac{1}{H}\boldsymbol{\Sigma}_a$.



1. Evaluate loss for H heads: $\mathcal{L} = \sum_{h=1}^{H} \mathcal{L}_{\text{NLL}}(\mathbf{q}_h, \mathbf{q}_t, \boldsymbol{\Sigma})$

$$\mathcal{L}_{\text{NLL}}(\mathbf{q}, \mathbf{q}_t, \boldsymbol{\Sigma}_a) = \frac{1}{2}\boldsymbol{\phi}^{\mathsf{T}}\boldsymbol{\Sigma}_a^{-1}\boldsymbol{\phi} + \frac{1}{2}\log\det(\boldsymbol{\Sigma}_a),$$
$$\boldsymbol{\phi} = \text{Log}\left(\mathbf{q} \otimes \mathbf{q}_t{}^{-1}\right)$$

(a) Training procedure (Algorithm 4).



1. Compute mean rotation $\bar{\mathbf{q}} = \dfrac{\sum_{h=1}^{H} \mathbf{q}_h}{\left\|\sum_{h=1}^{H} \mathbf{q}_h\right\|}$

2. Compute residuals $\boldsymbol{\phi}_h = \text{Log}\left(\mathbf{q}_h \otimes \bar{\mathbf{q}}^{-1}\right)$

3. Compute covariance as $\boldsymbol{\Sigma} = \dfrac{1}{H-1}\sum_{h=1}^{H} \boldsymbol{\phi}_h\boldsymbol{\phi}_h^{\mathsf{T}} + \boldsymbol{\Sigma}_a$

(b) Testing procedure (Algorithm 5).

Figure 7.8: The training and testing procedures for HydraNet for unit quaternion targets.

Algorithm 5 respectively.

Remark (*Combining epistemic and aleatoric uncertainty*). The simple addition of the two covariance matrices produces a valid covariance matrix (and follows prior work Kendall and Gal (2017) on combining aleatoric and epistemic sources). We leave an investigation of other possible way to combine these sources to future work.

7.4 Experiments

7.4.1 Uncertainty Evaluation: Synthetic Data

Before we embarked on training with real data, we analyzed our proposed HydraNet structure on a synthetic world. Our goal was to produce probabilistic estimates of camera orientation based on noisy pixel coordinates of a set of fixed point landmarks. To accomplish this, we simulated a monocular camera observing a planar grid of evenly spaced (see Figure 7.9a) landmarks from a hemisphere surrounding the grid. We aligned the monocular camera's optical axis with the centre of the hemisphere so that all landmarks were visible in every camera pose. At each pose, we computed noisy pixel locations of the projection of every landmark, and stacked these 2D locations as an input vector. We

Algorithm 4 Supervised training for $\mathrm{SO}(3)$ regression

Require: Training data $\mathcal{T}$, training targets $\mathbf{q}_t$, untrained model $g_\theta(\cdot)$ with parameters θ and $H+1$ heads

Ensure: Probabilistic regression model $g_\theta(\cdot)$

function TRAINHYDRANET($\mathcal{T}$)
  for each mini-batch $\mathcal{T}_i$ **do**
    Output $\mathbf{\Sigma}_a$ ▷ *1st head, Chol. decom.*
    for heads $2...(H+1)$ in g **do**
      Output $\mathbf{q}_h$ ▷ *Equation* (7.8)
      Evaluate NLL loss ▷ *Equation* (7.16)
    end forend
    Backprop, update θ
  end forend
  return $g(\cdot)$
end function

Algorithm 5 Testing of $\mathrm{SO}(3)$ regression

Require: Test sample $\mathcal{I}_j$, trained model $g_\theta(\cdot)$

Ensure: Test prediction $\mathbf{q}$, covariance $\mathbf{\Sigma}_t \succcurlyeq 0$

function TESTHYDRANET($\mathcal{I}_j$, $g_\theta(\cdot)$)
  Output $\mathbf{\Sigma}_a$ ▷ *1st head, Chol. decom.*
  for heads $2...(H+1)$ in g **do**
    Output $\mathbf{q}_h$ ▷ *Equation* (7.8)
  end forend
  Compute $\overline{\mathbf{q}}$ ▷ *Equation* (7.13)
  Compute $\mathbf{\Sigma}_e$ ▷ *Equation* (7.15)
  return $\overline{\mathbf{q}}$, $\mathbf{\Sigma}_e + \mathbf{\Sigma}_a$
end function

Table 7.1: HydraNet regression results for the 7scenes dataset compared to results reported in (Kendall and Cipolla, 2017). We report mean angular errors and the negative log likelihood (lower is better).

	Error (deg)		NLL	
Scene	HydraNet	PoseNet	HydraNet	PoseNet
Chess	6.3	4.5	-6.0	—
Fire	14.9	11.3	-3.6	—
Heads	14.3	13.0	-3.9	—
Office	8.6	5.6	-5.4	—
Pumpkin	9.0	4.8	-5.0	—
Kitchen	8.8	5.4	-5.0	—
Stairs	11.8	12.4	-4.7	—

generated 15000 training samples with poses that were randomly sampled from the hemisphere in the polar angle range of $[-60, 60]$ degrees. For testing, we sampled 500 poses in the range of $[-80, 80]$ degrees, purposely widening the range to include orientations that were not part of training.

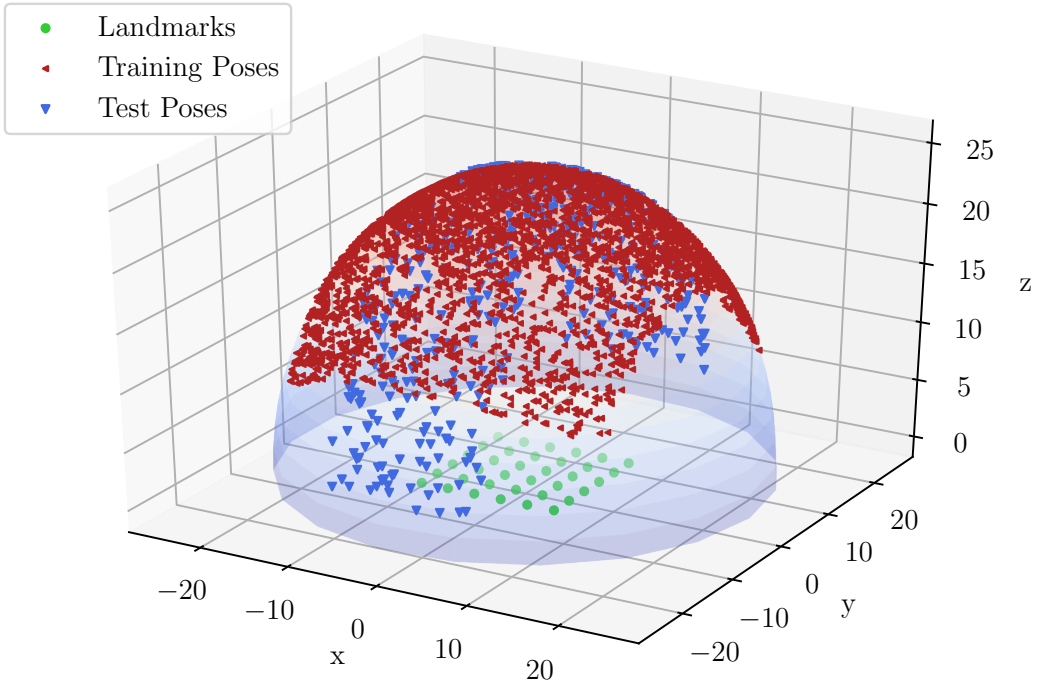
To regress the camera orientation, we constructed a five layer residual network and attached 26 heads (25 + 1 for direct uncertainty learning) to regress a probabilistic estimate of $\mathbf{q}_{c,w}$, the orientation of the camera with respect to the world frame.

Figure 7.9b plots rotational errors $\boldsymbol{\phi} = \text{Log}\left(\mathbf{q} \otimes \mathbf{q}_t^{-1}\right)$ along with 3 sigma bounds based on both the total covariance, $\boldsymbol{\Sigma}_t$, and the direct covariance $\boldsymbol{\Sigma}_a$. The final regression estimates have empirically consistent uncertainty, composed of a static aleatoric uncertainty and an epistemic uncertainty (Equation (7.15)) that grows when the test samples come from unfamiliar input data.

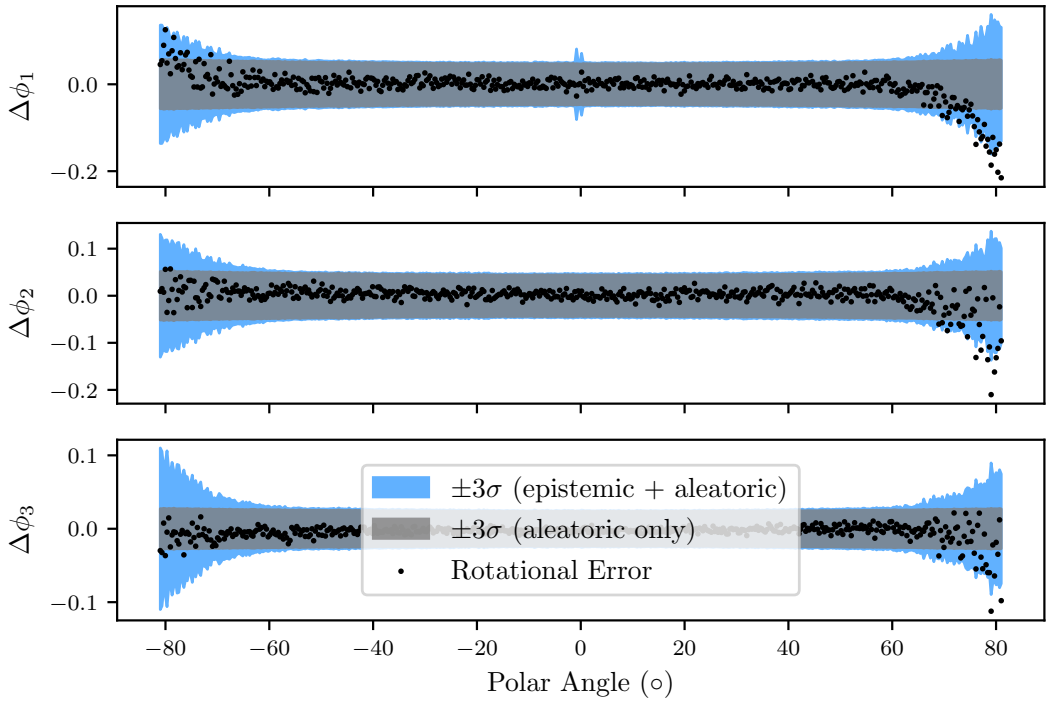
7.4.2 Absolute Orientation: 7-Scenes

Next, we used HydraNet to regress absolute orientations from RGB images from the 7-Scenes dataset (Glocker et al., 2013). Our goal was to achieve similar errors to other regression techniques (Kendall and Cipolla, 2017) but augment them with consistent covariance estimates. For this experiment, we used resnet34 (He et al., 2016) (pre-trained on the ImageNet dataset) for the body of HydraNet and attached 25 HydraNet heads, each consisting of two fully connected layers. We cropped and resized all RGB images to match the expected ImageNet size and omitted the depth channel.

Table 7.1 presents the mean angular errors and negative log likelihoods achieved by our method. The HydraNet-based network produces similar angular errors to other regression methods (Kendall and Cipolla, 2017) but with additional benefit of consistent three-degree-of-freedom uncertainty. Note that we spent little time optimizing the network itself, and note that state-of-the art errors can be achieved using more sophisticated pixel-based losses (Brachmann and Rother, 2018). However, the general HydraNet structure and loss can be used whenever a probabilistic rotation output is required. Further, our results show that our covariance formulation can be used for 'large' rotation elements, where techniques (e.g., (Peretroukhin and Kelly, 2018)) that assume 'small' corrections may fail.

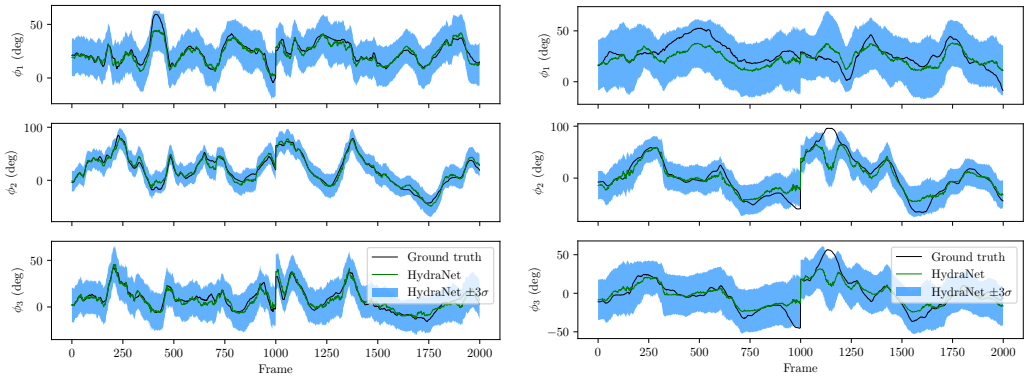


(a) Synthetic world used to illustrate our method. A monocular camera observes a 6×6 grid of point landmarks from poses sampled on a semi-sphere. The test set includes poses that are outside the training distribution.



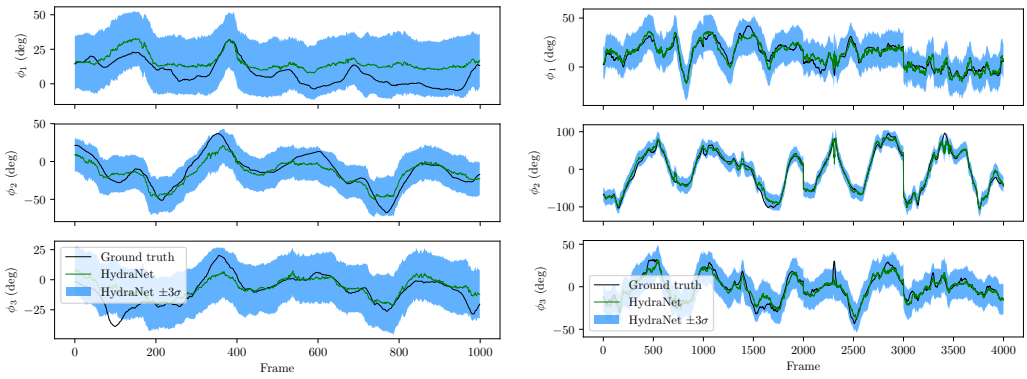
(b) Rotation estimation errors for a deep network trained using our HydraNet approach on synthetic data (noisy pixel locations of 36 landmarks). We note that outside of the training distribution, our epistemic uncertainty ($\mathbf{\Sigma}_e$) grows, as expected.

Figure 7.9: Synthetic experiments of probabilistic rotation regression with HydraNet.



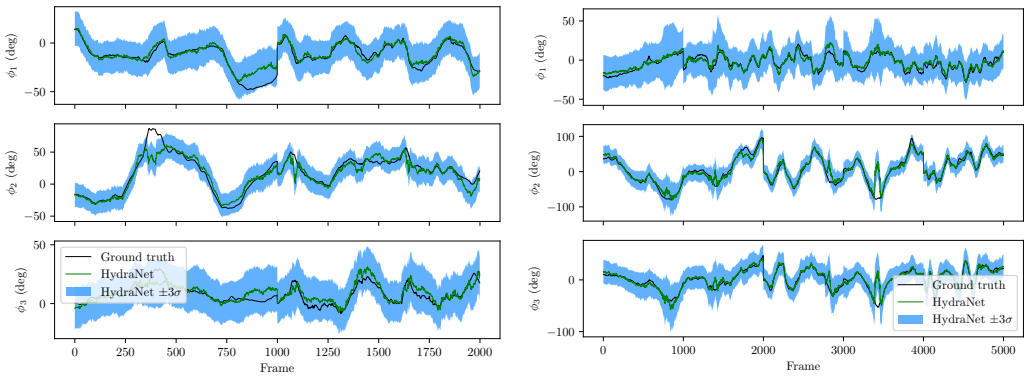
(a) Chess

(b) Fire



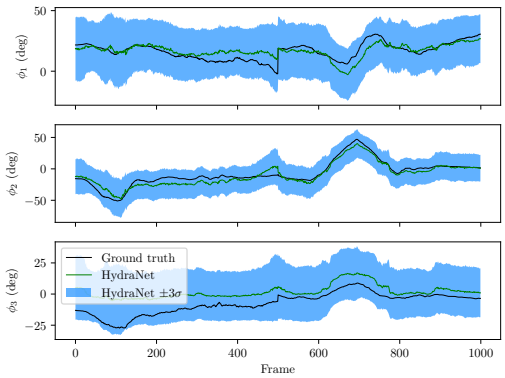
(c) Heads

(d) Office



(e) Pumpkin

(f) Kitchen



(g) Stairs

Figure 7.10: Probabilistic regression plots for all seven datasets from the 7-Scenes dataset.

Table 7.2: Results of fusing HydraNet relative rotation regression with classical stereo visual odometry.

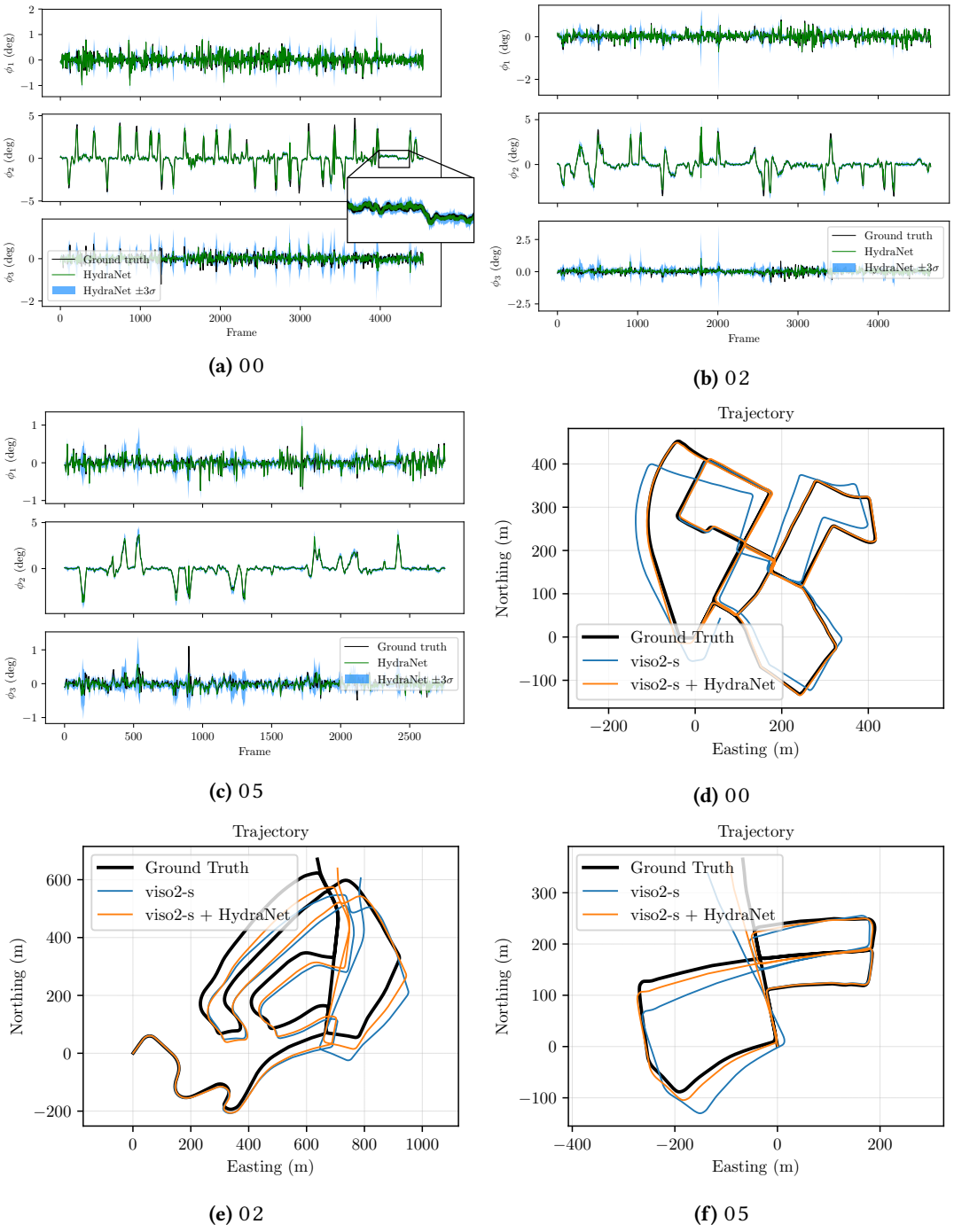
		m-ATE		Mean Segment Errors	
Sequence (Length)	Estimator	Translation (m)	Rotation (°)	Translation (%)	Rotation (°/100m)
00 (3.7 km)	DeepVO (Wang et al., 2017b)	–	–	–	–
	SfMLearner (Zhou et al., 2017)	–	–	65.27	6.23
	UnDeepVO (Li et al., 2017b)	–	–	4.14	1.92
	`viso2-s`	27.91	6.25	1.96	0.81
	`viso2-s` + HydraNet	9.86	2.83	1.34	0.63
	Keyframe Direct VO	12.41	2.45	1.28	0.54
02 (5.1 km)	DeepVO	–	–	–	–
	SfMLearner	–	–	57.59	4.09
	UnDeepVO	–	–	5.58	2.44
	`viso2-s`	64.67	8.45	1.47	0.56
	`viso2-s` + HydraNet	50.19	6.51	1.47	0.63
	Keyframe Direct VO	16.33	3.19	1.21	0.47
05 (2.2 km)	DeepVO	–	–	2.62	3.61
	SfMLearner	–	–	16.76	4.06
	UnDeepVO	–	–	3.40	1.50
	`viso2-s`	23.72	8.10	1.79	0.79
	`viso2-s` + HydraNet	9.85	3.23	1.38	0.60
	Keyframe Direct VO	5.83	2.05	0.69	0.32

Table 7.3: HydraNet regression results for the KITTI odometry dataset. We report mean angular errors and the negative log likelihood (lower is better).

Sequence	Mean Angular Error (°)	NLL
00	0.199	-16.84
02	0.138	-18.44
05	0.109	-19.31

7.4.3 Relative Rotation: KITTI Visual Odometry

Finally, to show the benefit of fusing deep probabilistic estimates with classical estimators, we trained a HydraNet network to estimate relative frame-to-frame rotations on the KITTI visual odometry (VO) benchmark. For each pair of poses, we process two RGB images (taken from the left RGB camera) into a two channel dense optical flow image using a fast classical algorithm (Farnebäck, 2003). Compared to using raw images, we found that using the optical flow pre-processing greatly improved training robustness and rotation accuracy. Since we use two-channel flow images, the body of the network is not pre-trained and instead contains an eight layer convolutional network. We maintained the same head structure as the 7-Scenes experiment. Table 7.3 and Figure 7.12 detail the mean test error and negative log likelihood for KITTI odometry sequences 00, 02 and 05 (chosen for their complexity and length). For each sequence, we trained the model on the remaining sequences in the benchmark. We found our model produced mean errors of approximately 0.1 degrees on all three test sequences. The covariance produced by HydraNet was consistent, spiking during yawing motions when the largest errors occurred (see Figure 7.11). Despite its consistency, the network covariance was dominated by $\boldsymbol{\Sigma}_a$. We suspect that unlike the synthetic data, $\boldsymbol{\Sigma}_e$ remained small throughout the tests sets due to a



(a) 00 **(b)** 02 **(c)** 05 **(d)** 00 **(e)** 02 **(f)** 05

Figure 7.11: Results for KITTI sequences 00, 02 and 05. Top-down trajectory plots show localization improvements after fusion with a classical stereo visual odometry pipeline.

more constrained input space (RGB or flow images, compared to pixel locations), but leave a thorough investigation to future work.

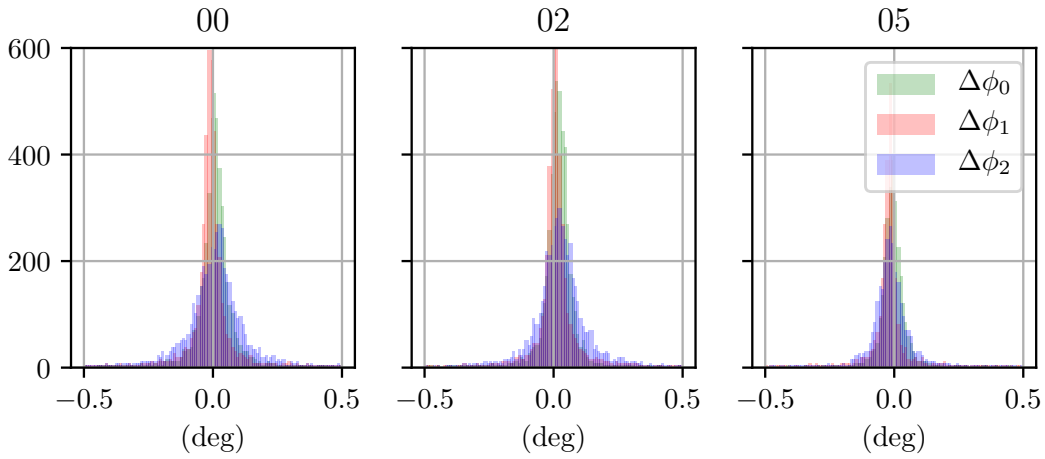


Figure 7.12: Error histograms for test KITTI sequences 00, 02, and 05 on three rotational axes.

Canonical Indirect Stereo Visual Odometry

For the classical visual odometry estimator, we use a similar pipeline to that used in the prior two chapters based on the open-source viso2 package (Geiger et al., 2011) to detect and track sparse stereo image key-points. In brief, our pipeline modelled stereo re-projection errors, $\mathbf{e}_{l,t_i}$, as zero-mean Gaussians with a known static covariance, $\mathbf{R}$. To generate an initial guess and to reject outliers, we used three point Random Sample Consensus (RANSAC) based on stereo re-projection error. Finally, we solved for the maximum likelihood transform, $\mathbf{T}_t^*$, through a Gauss-Newton minimization of

$$\mathbf{T}_t^* = \underset{\mathbf{T}_t \in \text{SE}(3)}{\text{argmin}} \sum_{i=1}^{N_t} \mathbf{e}_i^\mathsf{T} \mathbf{R}^{-1} \mathbf{e}_i. \tag{7.18}$$

After convergence, we approximate the frame-to-frame transformation uncertainty as (Barfoot, 2017):

$$\boldsymbol{\Sigma}_{\text{vo}} \approx \left(\sum_{i=1}^{N_t} \mathbf{J}_{\mathbf{e}_i}^\mathsf{T} \mathbf{R}^{-1} \mathbf{J}_{\mathbf{e}_i} \right)^{-1}, \tag{7.19}$$

where $\mathbf{J}_{\mathbf{e}_i}$ refers to the Jacobian of each reprojection error.

Fusion via Graph Relaxation

To fuse the output of our HydraNet pseudo-sensor with our canonical VO pipeline, we used pose graph relaxation. We describe our method briefly and refer the reader to (Barfoot, 2017) for a more detailed treatment. For every two poses, we defined a loss function based on a contribution from the estimator

and from the network, weighed by their respective covariances:

$$\mathbf{T}^*_{1,w}, \mathbf{T}^*_{2,w} = \underset{\mathbf{T}_{1,w}, \mathbf{T}_{2,w} \in \mathrm{SE}(3)}{\operatorname{argmin}} \mathcal{L}(\hat{\mathbf{T}}_{2,1}, \hat{\mathbf{C}}_{2,1}) \tag{7.20}$$

$$= \delta\boldsymbol{\xi}_{1,2}^{\mathsf{T}} \boldsymbol{\Sigma}_{\mathrm{vo}}^{-1} \delta\boldsymbol{\xi}_{1,2} + \delta\boldsymbol{\phi}_{1,2}^{\mathsf{T}} \boldsymbol{\Sigma}_{\mathrm{hn}}^{-1} \delta\boldsymbol{\phi}_{1,2} \tag{7.21}$$

where $\delta\boldsymbol{\xi}_{1,2} = \operatorname{Log}\left(\left(\mathbf{T}_{2,w}\mathbf{T}_{1,w}^{-1}\right)\hat{\mathbf{T}}_{2,1}^{-1}\right)$ and $\delta\boldsymbol{\phi}_{1,2} = \operatorname{Log}\left(\left(\mathbf{C}_{2,w}\mathbf{C}_{1,w}^{T}\right)\hat{\mathbf{C}}_{2,1}^{T}\right)$. The estimates $\hat{\mathbf{T}}_{2,1}$, $\boldsymbol{\Sigma}_{\mathrm{vo}}$ and $\hat{\mathbf{C}}_{2,1}$, $\boldsymbol{\Sigma}_{\mathrm{hn}}$ are provided by our classical estimator and the HydraNet network respectively.

Table 7.2 summarizes the results when we perform this fusion - and Figure 7.11 shows the final effect on the trajectory for sequence 00. We found that fusing deep rotation regression with classical methods results in motion estimates that significantly out-perform other methods that rely on deep regression alone. However, we note that even with consistent estimates, a small bias can affect the final fused estimates (e.g., sequence 05) and removing bias is an important avenue for future work. Further, the KITTI dataset contains few deleterious effects that negatively affect classical algorithms, and therefore we expect that this fusion would produce even more pronounced improvements on more varied visual data.

Remark (*Improving Uncertainty Estimates*). There are several salient extensions to the work presented here.

1. In order to ensure that the output of HydraNet and the output of the canonical VO pipeline are fused optimally, we can apply the method of *covariance intersection* (CI) (Julier and Uhlmann, 2007). CI is a technique to fuse measurements when the correlation between them is unknown and provides provably consistent estimates.

2. To improve epistemic uncertainty, we can investigate the application of a *gradient blockade* (Brachmann and Rother, 2019) between the heads and body of HydraNet. A gradient blockade would ensure that each head learns independently. At present, the $H+1$th head (which outputs *aleatoric* covariance) indirectly connects the gradients of the remaining heads by weighting the likelihood loss for each.

3. To further improve epistemic uncertainty, it is possible (as in Osband et al. (2016)) to train each head with a subset of training examples (mimicking the method of the statistical bootstrap, instead of relying solely on random initializations).

7.5 Summary

In this chapter, we described a method to regress probabilistic estimates of rotation using a deep multi-headed network structure. We used the quaternionic metric on $\mathrm{SO}(3)$ to define a rotation average, and extracted anisotropic covariances by modelling uncertainty through noise injection on the manifold.

Our novel contributions were

1. a deep network structure we call *HydraNet* that builds on prior work (Lakshminarayanan et al., 2017; Osband et al., 2016) 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.[4]

[4] https://github.com/utiasSTARS/so3_learning

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.

Amos, B. and Kolter, J. Z. (2017). OptNet: Differentiable optimization as a layer in neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 136–145. PMLR.

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.

Brachmann, E. and Rother, C. (2019). Neural- Guided RANSAC: Learning where to sample model hypotheses. In *ICCV*.

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.

Carlone, 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.

Censi, A. (2007). An accurate closed-form estimate of icp's covariance. In *Proceedings 2007 IEEE international conference on robotics and automation*, pages 3167–3172. IEEE.

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., 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.

Durrant-Whyte, H., Rye, D., and Nebot, E. (1996). Localization of autonomous guided vehicles. In *Robotics Research*, pages 613–625. Springer.

Eisenman, A. R., Liebe, C. C., and Perez, R. (2002). Sun sensing on the mars exploration rovers. In *Aerosp. 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.

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.

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. 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.

Gallego, G. and Yezzi, A. (2015). A compact formula for the derivative of a 3-D rotation in exponential coordinates. *J. Math. Imaging Vis.*, 51(3):378–384.

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 rgb-d 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.

Gustafsson, F. and Gustafsson, F. (2000). *Adaptive filtering and change detection*, volume 1. Citeseer.

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.

Julier, S. J. and Uhlmann, J. K. (2007). Using covariance intersection for slam. *Robotics and Autonomous Systems*, 55(1):3–20.

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. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision?

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.

Ko, J. and Fox, D. (2009). Gp-bayesfilters: Bayesian filtering using gaussian process prediction and observation models. *Autonomous Robots*, 27(1):75–90.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

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.

Liu, K., Ok, K., Vega-Brown, W., and Roy, N. (2018). Deep inference for covariance estimation: Learning gaussian noise models for state estimation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1436–1443. IEEE.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2.

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.

Melkumyan, A. and Ramos, F. (2011). Multi-kernel gaussian processes. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

Moravec, H. P. (1980). Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, Stanford Univ CA Dept of Computer Science.

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

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

Punjani, A. and Abbeel, P. (2015). Deep learning helicopter dynamics models. In *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, pages 3223–3230.

Ranftl, R. and Koltun, V. (2018). Deep fundamental matrix estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 284–299.

Richter, C. and Roy, N. (2017). Safe visual navigation via deep learning and novelty detection.

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.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. R. (2011). Orb: An efficient alternative to sift or surf. In *ICCV*, volume 11, page 2. Citeseer.

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.

Sobel, D. (2005). *Longitude: The true story of a lone genius who solved the greatest scientific problem of his time*. Macmillan.

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.

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

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

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

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.

Wilson, A. G. and Ghahramani, Z. (2011). Generalised wishart processes. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 736–744. AUAI Press.

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? *Science Robotics*, 4(30).

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.