

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 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	Combining Learning with Classical Pipelines	6
1.3	Original Contributions	8
2	Mathematical Foundations	10
2.1	Coordinate Frames	10
2.2	Rotations	11
2.3	Spatial Transforms	14
2.4	Perturbations and Tangent Spaces	15
2.5	Uncertainty on Lie Groups	16
2.6	Deep Learning	17
3	Classical Visual Odometry	21
3.1	Canonical VO Pipeline	22
3.2	Robust Estimation	27
3.3	Outstanding Issues	29
4	Predictive Robust Estimation	30
4.1	Introduction	30
4.2	Related Work	31
4.3	Predictive Robust Estimation for VO	33
4.4	Prediction Space	39
4.5	Experiments	44
4.6	Summary	49
5	Learned Probabilistic Sun Sensor	51
5.1	Introduction	51
5.2	Motivation	52
5.3	Related Work	53
5.4	Sun-Aided Stereo Visual Odometry	55

5.5	Orientation Correction	56
5.6	Bayesian Convolutional Neural Networks	57
5.7	Indirect Sun Detection using a Bayesian Convolutional Neural Network	62
5.8	Urban Driving Experiments: The KITTI Odometry Benchmark	64
5.9	Planetary Analogue Experiments: The Devon Island Rover Navigation Dataset	73
5.10	Sensitivity Analysis	77
5.11	Summary	83
6	Learned Pose Corrections	85
6.1	Introduction	85
6.2	Motivation	85
6.3	Related Work	87
6.4	System Overview: Deep Pose Correction	89
6.5	Experiments	93
6.6	Results & Discussion	100
6.7	Summary	102
7	Learned Probabilistic Rotations	103
7.1	Introduction	103
7.2	Motivation	104
7.3	Related work	104
7.4	Approach	105
7.5	Experiments	114
7.6	Summary	121
8	Conclusion	122
8.1	Summary of Contributions	122
8.2	Future Work	124
8.3	Coda: In Search of the Right Ends	126
Appendices		129
A	PROBE: Isotropic Covariance Models through k-Nearest Neighbours	130
A.1	Introduction	130
A.2	Theory	130
A.3	Training	131
A.4	Testing	131
A.5	Experiments	133

B Visual Odometry Implementation Details	135
B.1 Overview	135
B.2 Solution with Robust Loss	136
B.3 Deriving the Necessary Jacobians	136
Bibliography	139

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 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 5

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*

5.1 Introduction

Building on the results of PROBE, wherein we showed that it is possible to extract useful uncertainty information from images, we turn to the problem of constructing a pseudo-sensor that can infer a concrete geometric quantity that could be used to improve egomotion estimates. Specifically, in this chapter, we address the problem of replacing a costly dedicated hardware-based sun-sensor with a *sun pseudo-sensor*. We show, through extensive experiments, that it is possible to train a deep parametric model (a Bayesian Convolutional Neural Network, or BCNN) to act as a virtual sun sensor that can predict the direction of the sun from a single RGB image where the sun is not directly observed. Further, we leverage recent advances in the theory of approximate variational inference of deep networks to extract uncertainty estimates for each prediction.

Remark (Associated Publications). This project was a collaboration with Lee Clement. Lee lead the integration of sun information into a visual odometry pipeline, while I lead the design and implementation of the learning architecture. It is associated with three publications. The first publication was exploratory work on (non-probabilistic) virtual sun sensors lead by Lee,

1. 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.

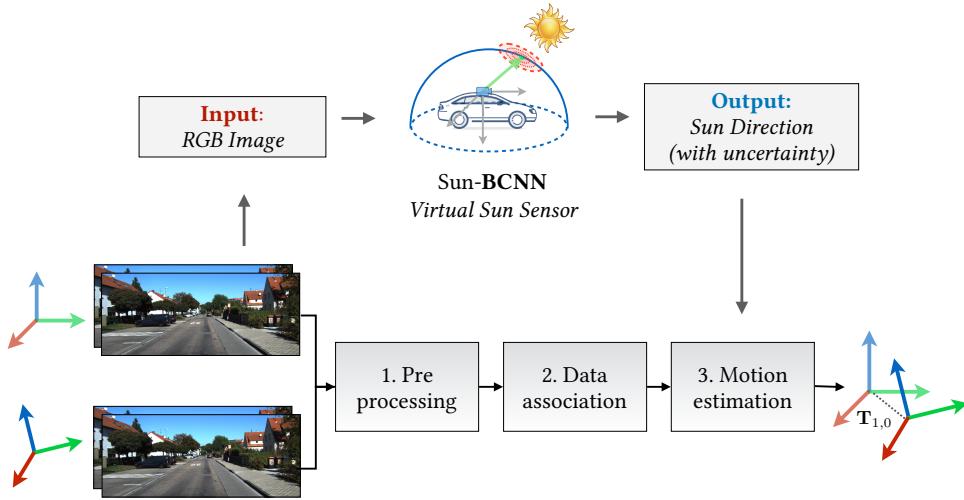


Figure 5.1: Sun-BCNN is a learned virtual sun sensor that outputs sun direction with an associated uncertainty based on a single RGB image. We use this as a source of orientation information within a privileged reference frame.

Invited to Journal Special Issue,

while the latter two publications were a joint collaboration where I developed and investigated the BCNN formulation and we collaborated on the experimental validation,

2. 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,
3. 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.

This chapter is largely a reproduction of the latter journal publication which summarizes the approach, with additional elaboration on the pseudo-sensor architecture.

5.2 Motivation

The accumulated error, or drift, of visual egomotion estimation can be limited by incorporating external global information such as observations of known landmarks, loop closures, or position updates from the global navigation satellite system (GNSS). In many situations, however, a globally consistent map may be unavailable or prohibitively expensive to compute, loop closures may not occur, or GNSS information may be unavailable or inaccurate (e.g., on the surface of Mars). In such cases, outdoor navigation can still extract external information from measurements of celestial objects (c.f. early maritime navigation described in Chapter 1). During the day, for example, observations of the sun can

provide global orientation information since the sun’s apparent motion in the sky is well-described by ephemeris models.

For visual odometry, the addition of global orientation information can limit the growth of drift error to be linear rather than superlinear with distance traveled (Olson et al., 2003). Accordingly, sun-based orientation corrections have been successfully used to improve VO estimates in planetary analogue environments (Furgale et al., 2011; Lambert et al., 2012) as well as on board the Mars Exploration Rovers (MERs) (Eisenman et al., 2002; Maimone et al., 2007). In particular, Lambert et al. (2012) showed that incorporating sun sensor and inclinometer measurements directly into the motion estimation pipeline can significantly reduce VO drift over long trajectories.

In this work, we build a virtual pseudo-sensor that can replicate the performance of a dedicated sun-sensor while requiring only the image stream that is already being used to compute VO. In particular, we leverage recent advances in approximate variational inference of Bayesian Convolutional Neural Networks to demonstrate how we can build and train a deep model capable of inferring the direction of the sun from a single RGB image, and moreover, produce a covariance estimate for each observation.

The remainder of this chapter begins with a discussion of related work, followed by an overview of the theory underlying BCNNs and a discussion of our model architecture, implementation, and training procedure. We then outline our chosen visual odometry pipeline, which is based on an adapted version of the pipeline described in Chapter 3, and describe how observations of the sun can be incorporated directly into the motion estimation problem following the technique of Lambert et al. (2012). Finally, we present several sets of experiments designed to test and validate both Sun-BCNN and our sun-aided VO pipeline in variety of environments. These include experiments on 21.6 km of urban driving data from the KITTI odometry benchmark training set (Geiger et al., 2013), as well as a 10 km traverse through a planetary analogue site taken from the Devon Island Rover Navigation Dataset collected in a planetary analogue site in the Canadian High Arctic (Furgale et al., 2012). Then, we investigate the possibility of model generalization between different cameras and environments, and further explore the sensitivity of Sun-BCNN to cloud cover during training and testing, using data from the Oxford Robotcar Dataset (Maddern et al., 2016). Finally, we examine the impact of different methods for computing the mean and covariance of a norm-constrained vector on the accuracy and consistency of the estimated sun directions.

5.3 Related Work

Hardware sun sensors have been used to improve the accuracy of VO in planetary analogue environments (e.g., the Sinclair Interplanetary SS-411 sun sensor used by Furgale et al. (2011) and Lambert et al. (2012)), while the MERs articulated their Pancam apparatus to directly image the sun (Maimone et al., 2007; Eisenman et al., 2002). More recently, software-based alternatives have been developed that can estimate the direction of the sun from a single image, making sun-aided navigation possible without additional sensors or a specially-oriented camera (Clement et al., 2017). Some of these methods

have been based on hand-crafted illumination cues such as shadows and variation in sky brightness (Lalonde et al., 2011; Clement et al., 2017), while others have attempted to learn such cues from data using deep Convolutional Neural Networks (CNNs) (Ma et al., 2016).

Convolutional Neural Networks (CNNs) have been applied to a wide range of classification, segmentation, and learning tasks in computer vision (LeCun et al., 2015). Recent work has shown that CNNs can learn orientation information directly from images by modifying the loss functions of existing discrete classification-based CNN architectures into continuous regression losses (Ma et al., 2016; Kendall et al., 2015; Kendall and Cipolla, 2016). Despite their success in improving prediction accuracy, most existing CNN-based models do not report uncertainty estimates, which are important in the context of data fusion.

For classification, it is possible to restrict CNN model outputs to a certain range (e.g., using a softmax function) and interpret these values as the model’s confidence in its output. As Gal (2016) noted, however, this can be misleading because these values can be unjustifiably large for test points far away from training data. To address this, Gal and Ghahramani (2016b) showed that it is possible to compute a better estimate of model uncertainty for classification and regression tasks, with only minor modifications to existing CNN architectures. An early application of this uncertainty quantification was presented by Kendall and Cipolla (2016) who used it to improve their prior work (Kendall et al., 2015) on camera pose regression.

We build on previous work by Clement et al. (2017), who demonstrated empirically that techniques for single-image sun estimation based on hand-crafted models (Lalonde et al., 2011) and Convolutional Neural Networks (CNNs) (Ma et al., 2016) could be incorporated into a stereo visual odometry pipeline to reduce estimation error in the manner of Lambert et al. (2012). We also build on the work of Peretroukhin et al. (2017), who presented preliminary experimental results comparing Sun-BCNN against the method of Lalonde et al. (2011) and its VO-informed variant (Clement et al., 2017) as well as the Sun-CNN of Ma et al. (2016) on the KITTI odometry benchmark (Geiger et al., 2013), both in terms of raw measurement accuracy and in terms of their impact on VO accuracy.

While our method is similar in spirit to the work of Ma et al. (2016), who built a CNN-based sun sensor as part of a relocalization pipeline, our model makes three important improvements: 1) in addition to a point estimate of the sun direction, we output a principled covariance estimate that is incorporated into our estimator; 2) we produce a full 3D sun direction estimate with azimuth and zenith angles that is better suited to 6-DOF robot pose estimation problems (as opposed to only the azimuth angle and 3-DOF estimator used by Ma et al. (2016)); and 3) we incorporate the sun direction covariance into a VO estimator that accounts for growth in pose uncertainty over time (unlike Clement et al. (2017)). Furthermore, our Bayesian CNN includes a dropout layer after every convolutional and fully connected layer (as outlined by Gal and Ghahramani (2016b) but not done by Kendall and Cipolla (2016)).

5.4 Sun-Aided Stereo Visual Odometry

For this work, we adopt a sliding window sparse stereo VO technique (derived from the frame-to-frame pipeline described in Chapter 3) that has been used in a number of successful mobile robotics applications (Cheng et al., 2006; Furgale and Barfoot, 2010; Geiger et al., 2011; Kelly et al., 2008). Our task is to estimate a window of SE(3) poses $\{\mathbf{T}_{k_1,0}, \mathbf{T}_{k_1+1,0}, \dots, \mathbf{T}_{k_2-1,0}, \mathbf{T}_{k_2,0}\}$ expressed in a base coordinate frame \mathcal{F}_0 , given a prior estimate of the transformation $\mathbf{T}_{k_1,0}$. We accomplish this by tracking keypoints across pairs of stereo images and computing an initial guess for each pose in the window using frame-to-frame point cloud alignment, which we then refine by solving a local bundle adjustment problem over the window. In our experiments we choose a window size of two, which we observed to provide good VO accuracy at low computational cost. We select the initial pose $\mathbf{T}_{1,0}$ to be the first ground truth pose such that \mathcal{F}_0 is a local East-North-Up (ENU) coordinate system with its origin at the first GPS position. We note that this requires some source of initial orientation information—e.g., a measurement of the local gravity vector and a magnetic heading.

5.4.1 Observation Model

We assume that incoming stereo images have been undistorted and rectified in a pre-processing step, and model the stereo camera as a pair of perfect pinhole cameras with focal lengths f_u, f_v and principal points (c_u, c_v) , separated by a fixed and known baseline b (see Section 3.1.2).

If we take \mathbf{p}_0^j to be the homogeneous 3D coordinates of keypoint j , expressed in our chosen base frame \mathcal{F}_0 , we can transform the keypoint into the camera frame at pose k to obtain $\mathbf{p}_k^j = \mathbf{T}_{k,0}\mathbf{p}_0^j = [p_{k,x}^j \ p_{k,y}^j \ p_{k,z}^j \ 1]^\top$. Our observation model $\mathbf{g}(\cdot)$ (defined with disparity, unlike the function $\mathbf{f}(\cdot)$ in Section 3.1.2) can then be formulated as

$$\mathbf{y}_{k,j} = \mathbf{g}(\mathbf{p}_k^j) = \begin{bmatrix} u \\ v \\ d \end{bmatrix} = \begin{bmatrix} f_u p_{k,x}^j / p_{k,z}^j + c_u \\ f_v p_{k,y}^j / p_{k,z}^j + c_v \\ f_u b / p_{k,z}^j \end{bmatrix}, \quad (5.1)$$

where (u, v) are the keypoint coordinates in the left image and d is the disparity in pixels.

5.4.2 Sliding Window Bundle Adjustment

Similar to PROBE, we use the open-source viso2 package (Geiger et al., 2011) to detect and track keypoints between stereo image pairs. Based on these keypoint tracks, a three-point Random Sample Consensus (RANSAC) algorithm (Fischler and Bolles, 1981) generates an initial guess of the interframe motion and rejects outlier keypoint tracks by thresholding their reprojection error. We compound these pose-to-pose transformation estimates through our chosen window and refine them using a local bundle adjustment, which we solve using the nonlinear least-squares solver Ceres (Agarwal et al.,

2016). The objective function to be minimized can be written as

$$\mathcal{J} = \mathcal{J}_{\text{reprojection}} + \mathcal{J}_{\text{prior}}, \quad (5.2)$$

where

$$\mathcal{J}_{\text{reprojection}} = \sum_{k=k_1}^{k_2} \sum_{j=1}^J \mathbf{e}_{\mathbf{y}_{k,j}}^\top \mathbf{R}_{\mathbf{y}_{k,j}}^{-1} \mathbf{e}_{\mathbf{y}_{k,j}} \quad (5.3)$$

and

$$\mathcal{J}_{\text{prior}} = \mathbf{e}_{\hat{\mathbf{T}}_{k_1,0}}^\top \mathbf{R}_{\hat{\mathbf{T}}_{k_1,0}}^{-1} \mathbf{e}_{\hat{\mathbf{T}}_{k_1,0}}. \quad (5.4)$$

The quantity $\mathbf{e}_{\mathbf{y}_{k,j}} = \hat{\mathbf{y}}_{k,j} - \mathbf{y}_{k,j}$ represents the reprojection error of keypoint j for camera pose k , with $\mathbf{R}_{\mathbf{y}_{k,j}}$ being the covariance of these errors. The predicted measurements are given by $\hat{\mathbf{y}}_{k,j} = \mathbf{g}(\hat{\mathbf{T}}_{k,0}\hat{\mathbf{p}}_0^j)$, where $\hat{\mathbf{T}}_{k,0}$ and $\hat{\mathbf{p}}_0^j$ are the estimated poses and keypoint positions in base frame \mathcal{F}_0 .

The cost term $\mathcal{J}_{\text{prior}}$ imposes a normally distributed prior $\check{\mathbf{T}}_{k_1,0}$ on the first pose in the current window, based on the estimate of this pose in the previous window. The error in the current estimate $\hat{\mathbf{T}}_{k_1,0}$ of this pose compared to the prior can be computed via the SE(3) matrix logarithm as $\mathbf{e}_{\check{\mathbf{T}}_{k_1,0}} = \text{Log}(\check{\mathbf{T}}_{k_1,0}^{-1}\hat{\mathbf{T}}_{k_1,0}) \in \mathbb{R}^6$. The 6×6 matrix $\mathbf{R}_{\check{\mathbf{T}}_{k_1,0}}$ is the covariance associated with $\check{\mathbf{T}}_{k_1,0}$ in its local tangent space, and is obtained as part of the previous window's bundle adjustment solution. This prior term allows consecutive windows of pose estimates to be combined in a principled way that appropriately propagates global pose uncertainty from window to window, which is essential in the context of optimal data fusion.

5.5 Orientation Correction

In order to combat drift in the VO estimate produced by accumulated orientation error, we adopt the technique of Lambert et al. (2012) to incorporate absolute orientation information from the sun directly into the estimation problem. We assume the initial camera pose and its timestamp are available from GPS-INS and use them to determine the global direction of the sun \mathbf{s}_0 , expressed as a 3D unit vector, from ephemeris data. We define the world frame \mathcal{F}_0 to be a local ENU coordinate system with the initial GPS position as its origin. At each timestep we update \mathbf{s}_0 by querying the ephemeris model using the current timestamp and the initial camera pose, allowing our model to account for the apparent motion of the sun over long trajectories.

By transforming the global sun direction into each camera frame \mathcal{F}_k in the window, we obtain predicted sun directions $\hat{\mathbf{s}}_k = \hat{\mathbf{T}}_{k,0}\mathbf{s}_0$, where $\hat{\mathbf{T}}_{k,0}$ is the current estimate of camera pose k in the base frame. We compare the predicted and estimated sun directions to introduce an additional error term into the bundle adjustment cost function (cf. Equation (5.2)):

$$\mathcal{J} = \mathcal{J}_{\text{reprojection}} + \mathcal{J}_{\text{prior}} + \mathcal{J}_{\text{sun}}, \quad (5.5)$$

where

$$\mathcal{J}_{\text{sun}} = \sum_{k=k_1}^{k_2} \mathbf{e}_{\mathbf{s}_k}^T \mathbf{R}_{\mathbf{s}_k}^{-1} \mathbf{e}_{\mathbf{s}_k}, \quad (5.6)$$

and $\mathcal{J}_{\text{reprojection}}$ and $\mathcal{J}_{\text{prior}}$ are defined in Equations (5.3) and (5.4), respectively. This additional term constrains the orientation of the camera, which helps limit drift in the VO result due to orientation error (Lambert et al., 2012).

Since \mathbf{s}_k is constrained to be unit length, there are only two underlying degrees of freedom. We therefore define $\mathbf{f}(\cdot)$ to be a function that transforms a 3D unit vector in camera frame \mathcal{F}_k to a zenith-azimuth parametrization:

$$\begin{bmatrix} \theta \\ \phi \end{bmatrix} = \mathbf{f}(\mathbf{s}_k) = \begin{bmatrix} \cos(-s_{k,y}) \\ \text{atan2}(s_{k,x}, s_{k,z}) \end{bmatrix} \quad (5.7)$$

where $\mathbf{s}_k = [s_{k,x} \ s_{k,y} \ s_{k,z}]^T$. We can then define the term $\mathbf{e}_{\mathbf{s}_k} = \mathbf{f}(\hat{\mathbf{s}}_k) - \mathbf{f}(\mathbf{s}_k)$ to be the error in the predicted sun direction, expressed in azimuth-zenith coordinates, and $\mathbf{R}_{\mathbf{s}_k}$ to be the covariance of these errors. While $\mathbf{R}_{\mathbf{s}_k}$ would generally be treated as an empirically determined static covariance, in our approach we use the per-observation covariance computed using Equation (5.29), which allows us to weight each observation individually according to a measure of its intrinsic quality. In practice, we also attempt to mitigate the effect of outlier sun predictions by applying a robust Huber loss to the sun measurements in our optimizer.

5.6 Bayesian Convolutional Neural Networks

Recall from Section 2.6.4 that dropout (Figure 5.2) is a stochastic regularization technique that was originally designed to improve the generalization performance of deep networks. Dropout works by randomly ‘turning off’ certain inputs of layers to simulate an ensemble of smaller networks during training. Gal and Ghahramani (2016a) identified a link between applying dropout at test-time given a network trained with dropout (through a technique called *Monte Carlo dropout*, or *MC dropout*) and approximate variational inference for a particular type of Bayesian Neural Network. In this work, we leverage this insight to build a Bayesian Convolutional Neural Network through MC dropout. To elucidate the intuition behind this technique, we first review variational inference and then outline its connection to the technique of dropout.

5.6.1 Bayesian Modelling and Variational Inference

Consider the goal of generating a function $f(\mathbf{x}; \boldsymbol{\pi})$ that maps inputs \mathbf{x} to outputs \mathbf{y} as a function of parameters $\boldsymbol{\pi}$. Recall from Chapter 4 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, in regression problems, we often define the Gaussian likelihood

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\pi}) = \mathcal{N}(f(\mathbf{x}; \boldsymbol{\pi}), \tau^{-1} \mathbf{1}) \quad (5.8)$$

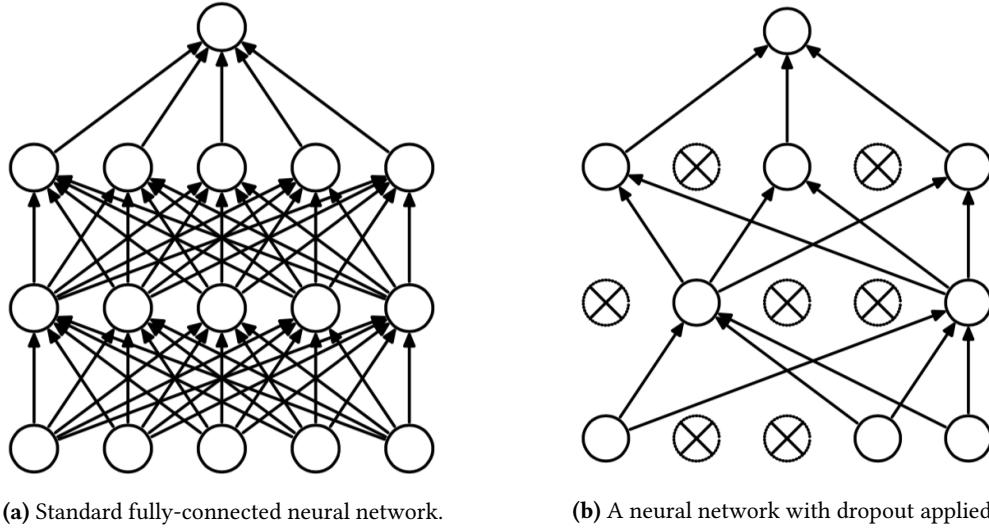


Figure 5.2: The technique of *dropout* stochastically removes the contribution of certain neurons to regularize learning. Figures from [Srivastava et al. \(2014\)](#).

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 $\boldsymbol{\pi}$:

$$p(\boldsymbol{\pi}|\mathcal{D}) = \frac{p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\pi})p(\boldsymbol{\pi})}{p(\mathbf{Y}|\mathbf{X})} \quad (5.9)$$

where we may think of this posterior as evaluating the sentence “the likelihood of the observed outputs given a particular set of parameters (weighted 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,

$$p(\mathbf{Y}|\mathbf{X}) = \int_{\boldsymbol{\pi}} p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\pi})p(\boldsymbol{\pi})d\boldsymbol{\pi}. \quad (5.10)$$

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:

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \int_{\boldsymbol{\pi}} p(\mathbf{y}^*|\mathbf{x}^*, \boldsymbol{\pi})p(\boldsymbol{\pi}|\mathcal{D})d\boldsymbol{\pi}. \quad (5.11)$$

For most problems, evaluating the posterior (Equation (5.9)) analytically is not feasible. In such cases, we can turn to 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 ([Kullback and Leibler, 1951](#))

between $q(\boldsymbol{\pi}; \boldsymbol{\theta})$ and the true posterior $p(\boldsymbol{\pi}|\mathcal{D})$:

$$\boldsymbol{\theta}^* = \operatorname{argmin}_{\boldsymbol{\theta}} \text{KL}(q(\boldsymbol{\pi}; \boldsymbol{\theta}) || p(\boldsymbol{\pi}|\mathcal{D})) = \operatorname{argmin}_{\boldsymbol{\theta}} \int_{\boldsymbol{\pi}} q(\boldsymbol{\pi}; \boldsymbol{\theta}) \log \frac{q(\boldsymbol{\pi}; \boldsymbol{\theta})}{p(\boldsymbol{\pi}|\mathcal{D})} d\boldsymbol{\pi}, \quad (5.12)$$

which then allows us to compute the approximate predictive distribution as

$$p(\mathbf{y}^* | \mathbf{x}^*, \mathcal{D}) \approx \int_{\boldsymbol{\pi}} p(\mathbf{y}^* | \mathbf{x}^*, \boldsymbol{\pi}) q(\boldsymbol{\pi}; \boldsymbol{\theta}^*) d\boldsymbol{\pi}. \quad (5.13)$$

One can show that minimizing KL divergence (Equation (5.12)) is the same as maximizing the evidence lower bound (ELBO)¹

$$\boldsymbol{\theta}^* = \operatorname{argmax}_{\boldsymbol{\theta}} -\mathcal{L}_{\text{ELBO}} = \operatorname{argmax}_{\boldsymbol{\theta}} \int_{\boldsymbol{\pi}} \log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\pi}) q(\boldsymbol{\pi}; \boldsymbol{\theta}) d\boldsymbol{\pi} - \text{KL}(q(\boldsymbol{\pi}; \boldsymbol{\theta}) || p(\boldsymbol{\pi})). \quad (5.14)$$

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

5.6.2 Monte Carlo Dropout as Approximate Variational Inference

A key insight of Gal (2016); Gal and Ghahramani (2016b) was that the training objective of a neural network with dropout resembles the ELBO optimization of Equation (5.14) with a careful choice of $q(\boldsymbol{\pi}; \boldsymbol{\theta})$. Namely, consider training a neural network with dropout and with another form of regularization called *weight decay* (which penalizes large weight magnitudes). The loss function for supervised training in this scenario can be written as

$$\mathcal{L}_{\text{dropout}}(\boldsymbol{\pi}) = \underbrace{\frac{1}{N} \sum_{i=1}^N \mathcal{E}(\mathbf{y}_i, \tilde{\mathbf{y}}_i)}_{\text{error function}} + \underbrace{\lambda \sum_{\ell=1}^L (\|\mathbf{W}_\ell\|_2^2 + \|\mathbf{b}_\ell\|_2^2)}_{\text{weight decay}}, \quad (5.15)$$

where \mathbf{y}_i is the target output, λ is a scalar weight decay hyper-parameter, and $\tilde{\mathbf{y}}_i = NN(\mathbf{x}_i)$ is a stochastic sample of the network output with dropout applied. Now if we define $q(\boldsymbol{\pi})$ to be a *Bernoulli variational distribution* over the network parameters as the re-parametrized density,

$$q(\boldsymbol{\pi}; \boldsymbol{\theta}) = \{\operatorname{diag}(\{\epsilon_{i\ell}\}) \mathbf{W}_\ell, \mathbf{b}_\ell\}_{\ell=1}^L \quad (5.16)$$

$$\epsilon_{i\ell} \sim \text{Bernoulli}(p) \quad (5.17)$$

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

where the auxiliary variables become the network parameters themselves, $\boldsymbol{\theta} = \{\mathbf{W}_\ell, \mathbf{b}_\ell\}_{\ell=1}^L$, then we can show an equivalence between optimizing the variational inference ELBO loss function,

$$\mathcal{L}_{\text{ELBO}}(\boldsymbol{\theta}) = - \int_{\boldsymbol{\pi}} \log p(\mathbf{Y}|\mathbf{X}, \boldsymbol{\pi}) q(\boldsymbol{\pi}; \boldsymbol{\theta}) d\boldsymbol{\pi} + \text{KL}(q(\boldsymbol{\pi}; \boldsymbol{\theta}) \parallel p(\boldsymbol{\pi})) \quad (5.18)$$

$$= - \sum_{i=1}^N \int_{\boldsymbol{\pi}} \log p(\mathbf{y}_i | \mathbf{x}_i, \boldsymbol{\pi}) q(\boldsymbol{\pi}; \boldsymbol{\theta}) d\boldsymbol{\pi} + \text{KL}(q(\boldsymbol{\pi}; \boldsymbol{\theta}) \parallel p(\boldsymbol{\pi})) \quad (5.19)$$

and optimizing Equation (5.15) above as follows. First, the first term above can be approximated through Monte Carlo integration with a single sample² $\tilde{\boldsymbol{\pi}}$ of $q(\boldsymbol{\pi}; \boldsymbol{\theta})$ such that,

$$= - \sum_{i=1}^N \log p(\mathbf{y}_i | \mathbf{x}_i, \tilde{\boldsymbol{\pi}}) + \text{KL}(q(\boldsymbol{\pi}; \boldsymbol{\theta}) \parallel p(\boldsymbol{\pi})). \quad (5.20)$$

Second, it is possible to show that there exists a Gaussian prior $p(\boldsymbol{\pi})$ such that the condition (Eq. 3.12, Gal (2016))

$$\frac{\partial \text{KL}(q(\boldsymbol{\pi}; \boldsymbol{\theta}) \parallel p(\boldsymbol{\pi}))}{\partial \boldsymbol{\theta}} \approx \frac{\partial \lambda \sum_{\ell=1}^L (\|\mathbf{W}_\ell\|_2^2 + \|\mathbf{b}_\ell\|_2^2)}{\partial \boldsymbol{\theta}} \quad (5.21)$$

holds. Finally, if we interpret $-\log p(\mathbf{y}_i | \mathbf{x}_i, \tilde{\boldsymbol{\pi}})$ as $\mathcal{E}(\mathbf{y}_i, \tilde{\mathbf{y}}_i)$ then it follows that

$$\frac{\partial \mathcal{L}_{\text{ELBO}}}{\partial \boldsymbol{\theta}} \propto \frac{\partial \mathcal{L}_{\text{Dropout}}}{\partial \boldsymbol{\theta}} \quad (5.22)$$

where the proportional constant can be written down analytically (Gal, 2016). Thus the two objectives result in the same optimal parameters.

Computing Moments

The results above indicate that, once trained, a neural network with dropout gives us access to a predictive distribution over outputs \mathbf{y} . An unbiased estimator for its mean and covariance are given by applying dropout at test time and computing statistics over T samples:

$$\mathbb{E}[\mathbf{y}^*] = \frac{1}{T} \sum_{t=1}^T \tilde{\mathbf{y}}_t^*, \quad (5.23)$$

where $\tilde{\mathbf{y}}_t^* = NN(\mathbf{x}^*; \tilde{\boldsymbol{\pi}}_t)$ and

$$\text{Var}(\mathbf{y}^*) = \tau^{-1} \mathbf{1} + \frac{1}{T} \sum_{t=1}^T \tilde{\mathbf{y}}_t^* (\tilde{\mathbf{y}}_t^*)^\top - \mathbb{E}[\mathbf{y}^*] \mathbb{E}[\mathbf{y}^*]^\top, \quad (5.24)$$

²In practice, stochastic gradient descent will select a mini-batch of these samples to give an unbiased estimate.

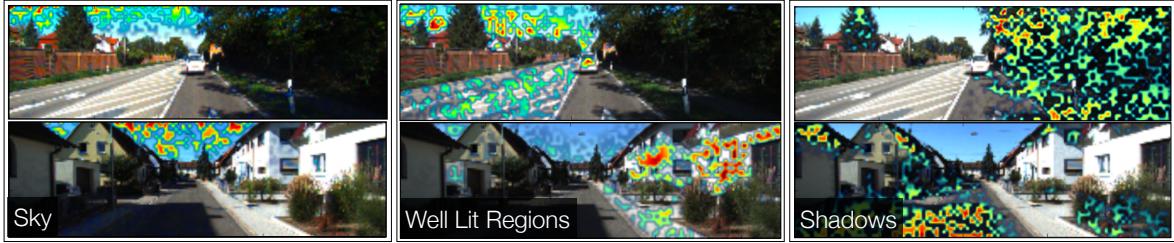


Figure 5.3: Three conv1 layer activation maps superimposed on two images from the KITTI odometry benchmark (Geiger et al., 2013) 00 and 04 for three selected filters. Each filter picks out salient parts of the image that aid in sun direction inference.

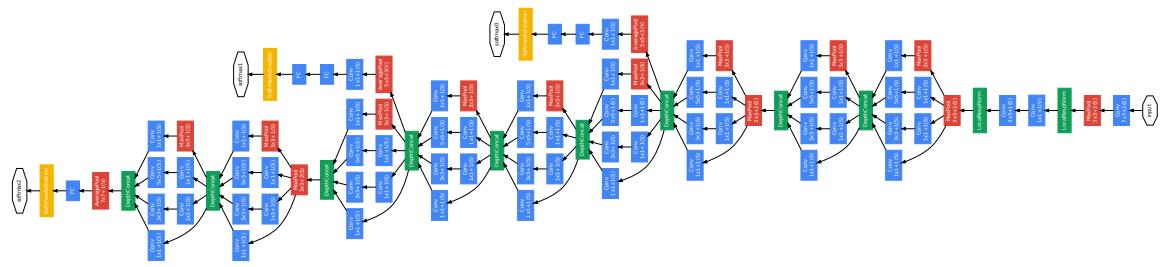


Figure 5.4: The GoogLeNet architecture used in this work. This deep convolutional network relies on *inception* modules that concatenate the output of several convolutions with a max-pooling operation. Figure from Szegedy et al. (2015).

where the precision τ defines the likelihood

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\pi}) = \mathcal{N}(NN(\mathbf{x}; \boldsymbol{\pi}), \tau^{-1}\mathbf{1}) \quad (5.25)$$

and can be alternatively modelled as

$$\tau = \frac{l^2(1-p)}{2N\lambda}, \quad (5.26)$$

where l defines a characteristic length scale (Gal, 2016). This test-time stochastic sampling through dropout is referred to as Monte Carlo dropout.

5.6.3 Extension to Convolutional Neural Networks

Although the derivation is more tedious, Gal (2016); Gal and Ghahramani (2016a) show that the same approximate variational inference can be carried out for convolutional neural networks by applying dropout after every convolution (and before pooling) during training and using MC dropout during testing.

5.7 Indirect Sun Detection using a Bayesian Convolutional Neural Network

Leveraging the work of [Gal and Ghahramani \(2016a\)](#), we apply MC dropout to a Convolutional Neural Network (CNN) (approximating a BCNN, or Bayesian CNNN) to infer the direction of the sun and an associated uncertainty, and refer to our model as Sun-BCNN. We motivate the choice of a deep model through the empirical findings of [Clement et al. \(2017\)](#) and [Ma et al. \(2016\)](#), who demonstrated that a CNN-based sun detector can substantially outperform hand-crafted models such as that of [Lalonde et al. \(2011\)](#) both in terms of measurement accuracy and in its application to a VO task.

We choose a deep neural network structure based on GoogLeNet ([Szegedy et al., 2015](#)) due to its use in past work that adapted it for orientation regression ([Kendall and Cipolla, 2016; Kendall et al., 2015](#)). Unlike [Ma et al. \(2016\)](#), we choose to transfer weights trained on the MIT Places dataset ([Zhou et al., 2014](#)) rather than ImageNet ([Deng et al., 2009](#)). We believe the MIT Places dataset is a more appropriate starting point for localization tasks than ImageNet since it includes outdoor scenes and is concerned with classifying physical locations rather than objects.

5.7.1 Cost Function

We train Sun-BCNN by minimizing the cosine distance between the unit-norm target sun direction vector \mathbf{s}_k and the predicted unit-norm sun direction vector $\hat{\mathbf{s}}_k$, where k indexes the images in the training set:

$$\mathcal{L}(\hat{\mathbf{s}}_k) = 1 - (\hat{\mathbf{s}}_k \cdot \mathbf{s}_k). \quad (5.27)$$

Note that in our implementation, we do not formulate the cosine distance loss explicitly, but instead minimize half the square of the tip-to-tip Euclidian distance between \mathbf{s}_k and $\hat{\mathbf{s}}_k$, which is equivalent to Equation (5.27) since both vectors have unit length:

$$\begin{aligned} \frac{1}{2} \|\hat{\mathbf{s}}_k - \mathbf{s}_k\|^2 &= \frac{1}{2} \left(\|\hat{\mathbf{s}}_k\|^2 + \|\mathbf{s}_k\|^2 - 2(\hat{\mathbf{s}}_k \cdot \mathbf{s}_k) \right) \\ &= 1 - (\hat{\mathbf{s}}_k \cdot \mathbf{s}_k) \\ &= \mathcal{L}(\hat{\mathbf{s}}_k). \end{aligned}$$

We ensure that our network output, $\hat{\mathbf{s}}_k$, has a unit norm by appending a normalization layer to the network.

5.7.2 Uncertainty Estimation

Using our notation for sun direction, we obtain the first two moments of the predictive distribution from our BCNN as (c.f. Equations (5.23) and (5.24)):

$$\mathbb{E} [\hat{\mathbf{s}}^*]_k = \hat{\mathbf{s}}_k^* \approx \frac{1}{N} \sum_{n=1}^N \hat{\mathbf{s}}_k^*(\mathbf{x}_k^*, \mathbf{w}^n) \quad (5.28)$$

$$\begin{aligned} \text{Var}(\hat{\mathbf{s}}_k^*) &\approx \tau^{-1} \mathbf{1} + \frac{1}{N} \sum_{n=1}^N \hat{\mathbf{s}}_k^*(\mathbf{x}_k^*, \mathbf{w}^n) \hat{\mathbf{s}}_k^*(\mathbf{x}_k^*, \mathbf{w}^n)^T \\ &\quad - \hat{\mathbf{s}}_k^* \hat{\mathbf{s}}_k^{*\top}, \end{aligned} \quad (5.29)$$

where $\mathbf{1}$ is the identity matrix, and \mathbf{w}^n is a weight sample obtained indirectly by performing a forward pass with dropout. Following Gal and Ghahramani (2016a), we build our BCNN by adding dropout layers after every convolutional and fully connected layer in the network. We then retain these layers at test time to sample the network stochastically, following the technique of Monte Carlo Dropout, and obtain the relevant statistical quantities using Equations (5.28) and (5.29).

5.7.3 Implementation and Training

We implement our network in Caffe (Jia et al., 2014), using the L2Norm layer from the Caffe-SL fork³ to enforce a unit-norm constraint on the final output. We train the network using stochastic gradient descent, setting all dropout probabilities to 0.5, performing 30,000 iterations with a batch size of 64, and setting the initial learning rate to be between 10^{-3} and 10^{-4} . Training requires approximately 2.5 hours on an NVIDIA Titan X GPU. Interestingly, Figure 5.4 shows that some convolutional filters learned by Sun-BCNN on the KITTI dataset appear to correspond to illumination variations reminiscent of the visual cues designed by Lalonde et al. (2011).

Data Preparation & Transfer Learning

We resize images from their original size to $[224 \times 224]$ pixels to achieve the size expected by Google-LeNet. We experimented with preserving the aspect ratio of the original image and padding zeros to the top and bottom of the resized image, but found that preserving the vertical resolution (as done by Ma et al. (2016)) results in better test-time accuracy. We do not crop or rotate the images, nor do we augment the dataset in any other way.

Model Precision

We find an empirically optimal model precision τ by optimizing the Average Normalized Estimation Error Squared (ANees) across the entire test set for each dataset. While this hyperparameter should

³<https://github.com/wanji/caffe-sl>

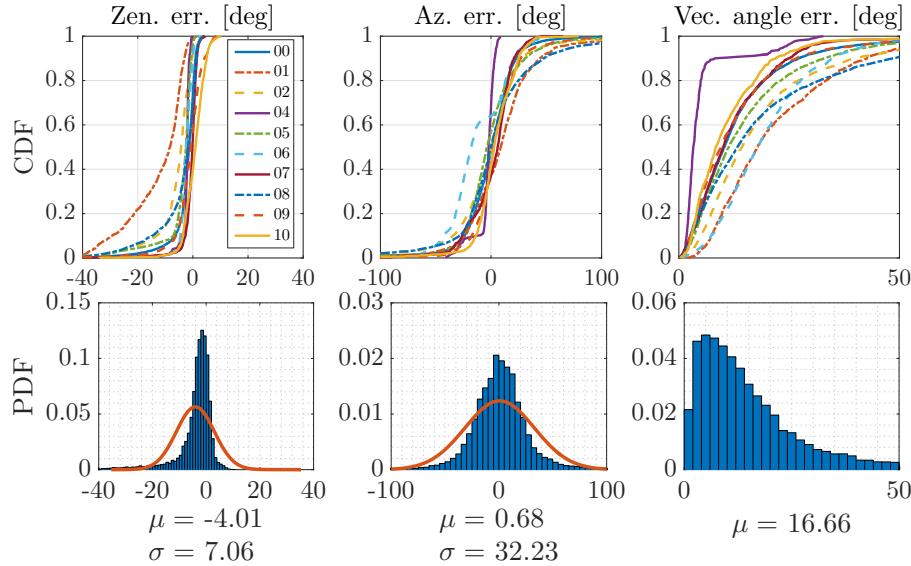


Figure 5.5: Distributions of azimuth error, zenith error, and angular distance for Sun-BCNN compared to ground truth over each test sequence in the KITTI dataset. *Top row*: Cumulative distributions of errors for each test sequence individually. *Bottom row*: Histograms and Gaussian fits of aggregated errors.

in principle be tuned using a validation set, we omit this step to keep our training procedure consistent with that of [Ma et al. \(2016\)](#). We note that the BCNN uncertainty estimates are affected by two significant factors: 1) variational inference is known to underestimate predictive variance ([Gal, 2016](#)); and 2) we assume the observation noise is homoscedastic. As noted by [Gal \(2016\)](#), the BCNN can be made heteroscedastic by learning the model precision during training, but this extension is outside the scope of this work.

Data Partitioning

We partition our data into training and testing sets using a leave-one-out approach based on temporally disjoint sequences of images. That is, given N sequences, the model tested on sequence i is trained with sequences $\{1, 2, \dots, N\} \setminus i$. This process varies based on the dataset, and we discuss the specifics in the experimental discussion corresponding to each. In contrast to randomly holding out a subset of the data, this method minimizes the similarity of training and testing data for temporally correlated image streams.

5.8 Urban Driving Experiments: The KITTI Odometry Benchmark

We investigated the performance of Sun-BCNN on the KITTI odometry benchmark training set ([Geiger et al., 2013](#)), which consists of 21.6 km of urban driving data⁴. Importantly, the dataset includes 6-

⁴Because we rely on the first pose reported by the GPS/INS system, we used the raw (rectified and synchronized) sequences corresponding to each odometry sequence. However, the raw sequence 2011_09_26_drive_0067 corresponding to odometry sequence 03 was not available on the KITTI website at the time of writing, so we omit sequence 03 from our analysis.

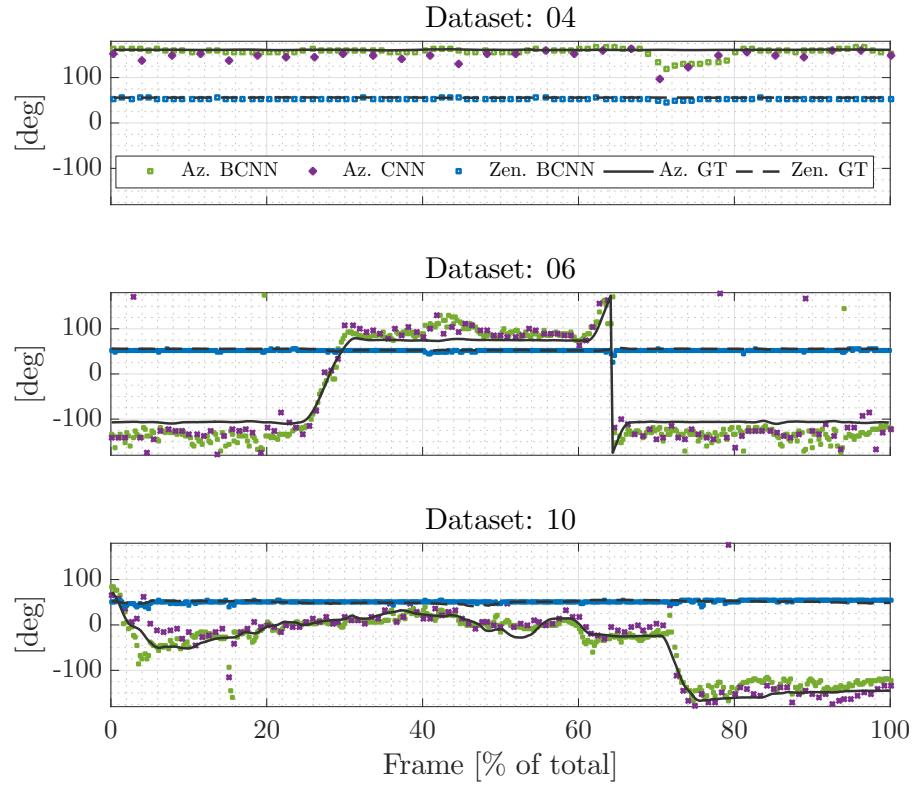


Figure 5.6: Azimuth (Sun-CNN and Sun-BCNN) and zenith (Sun-BCNN only) predictions over time for KITTI test sequences 04, 06 and 10. Sun-CNN is trained and tested on every tenth image, whereas Sun-BCNN is trained and tested on every image. In our VO experiments, we use the Sun-BCNN predictions of every tenth image to make a fair comparison.

Table 5.1: Test Errors for Sun-BCNN on KITTI odometry sequences with estimates computed at every image.

Sequence	Zenith Error [deg]			Azimuth Error [deg]			Vector Angle Error [deg]			ANEE¹
	Mean	Median	Stdev	Mean	Median	Stdev	Mean	Median	Stdev	
00	-2.59	-1.37	5.15	-0.33	0.81	25.61	13.56	10.31	13.14	1.00
01	-12.53	-8.31	10.33	8.95	8.83	33.67	22.16	17.85	15.00	1.38
02	-6.13	-4.26	7.38	-1.03	0.74	37.61	19.69	14.32	18.25	1.40
04	-2.42	-2.11	1.64	-3.89	-2.18	9.14	5.33	3.29	6.44	0.30
05	-4.31	-2.51	6.18	-0.74	-3.80	29.81	15.66	11.33	14.80	1.05
06	-2.48	-2.52	2.27	-12.22	-17.86	25.78	19.78	17.72	11.35	1.93
07	-0.69	-0.16	3.26	1.25	5.98	20.27	12.44	10.05	9.97	0.97
08	-4.46	-1.61	8.14	3.66	-0.14	41.73	19.90	13.30	19.59	1.04
09	-1.35	-0.75	5.60	4.78	2.36	23.84	13.09	9.48	12.66	0.73
10	0.59	0.95	3.90	3.64	2.61	19.15	11.23	8.34	9.83	1.08
All	-4.01	-2.26	7.06	0.68	0.53	32.23	16.66	12.08	15.91	-

¹ We compute Average Normalized Estimation Error Squared (ANEE¹) values with all sun directions that fall below a cosine distance threshold of 0.3 (relative to ground truth) and set $\tau^{-1} = 0.015$.

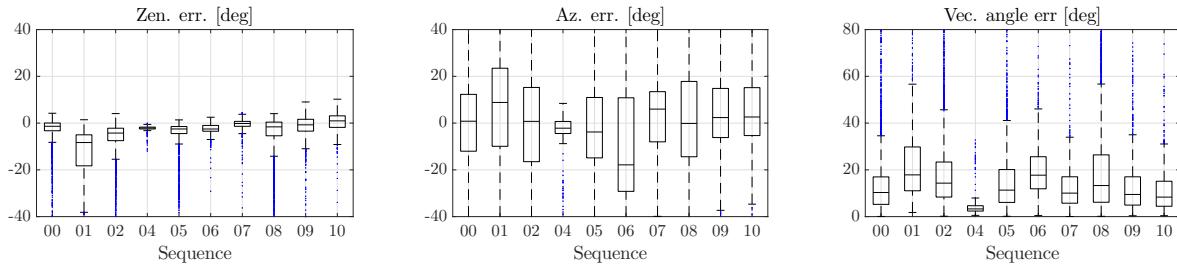


Figure 5.7: Box-and-whiskers plot of final test errors on all ten KITTI odometry sequences (c.f. Table 5.1).

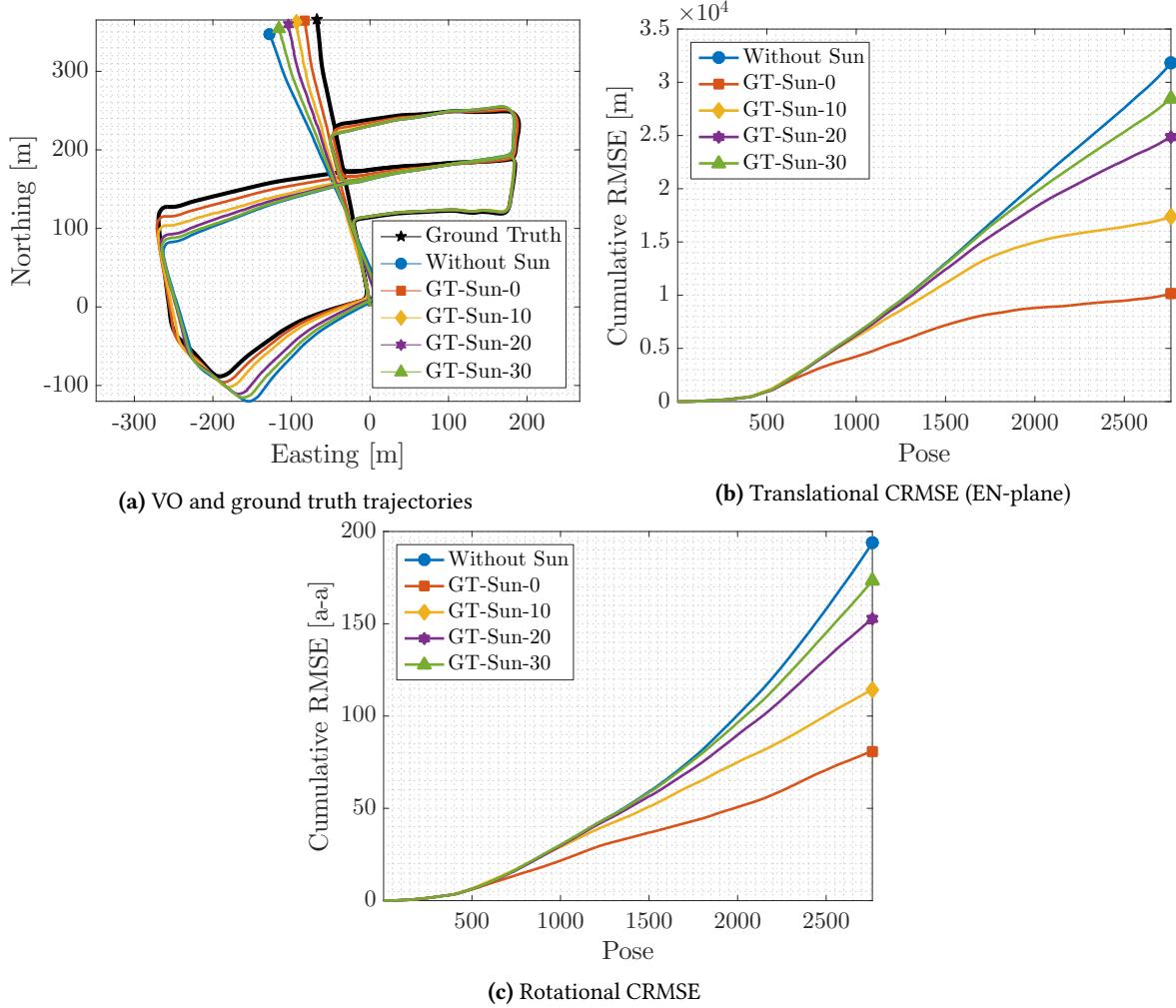


Figure 5.8: VO results for KITTI odometry sequence 05 using simulated sun measurements at every tenth pose. We observe a clear progression in cumulative root mean squared error (CRMSE) in translation and rotation as noise in the simulated sun measurements increases.

DOF ground truth poses obtained from an accurate GPS/INS tracking system, as well as calibrated transformations between this sensor and the colour stereo pair we use for sun estimation and VO in our experiments. This allows us to create a training set of ground truth sun vectors for each image by querying the solar ephemeris model at each ground truth pose and rotating the resulting vector from the GPS/INS frame \mathcal{F}_0 (which is an ENU coordinate system) into the camera coordinate frame \mathcal{F}_k . For each of our experiments, we trained Sun-BCNN on nine benchmark sequences and tested on the remaining sequence. This procedure is consistent with that of [Ma et al. \(2016\)](#), against whose Sun-CNN we directly compare, and allows us to evaluate each sequence using the maximum amount of training data.



Figure 5.9: Sun BCNN predictions and associated ground truth sun directions on the KITTI sequence 05. *Top two rows:* Sun BCNN produces accurate predictions in a variety of azimuth values. *Bottom row:* Poor results occur rarely due to shadow ambiguities.

5.8.1 Sun-BCNN Test Results

Once trained, we analyzed the accuracy and consistency of the Sun-BCNN mean and covariance estimates. We obtained the mean estimated sun vector by evaluating Equation (5.28) with $N = 25$ and then re-normalized the resulting vector to preserve unit length. To obtain the required covariance on azimuth and zenith angles, we sampled the vector outputs, converted them to azimuth and zenith angles using Equation (5.7), and then applied Equation (5.29).

Remark (Number of stochastic passes). We base our choice of $N = 25$ on the results presented in [Gal \(2016\)](#), Figure 4.12. Namely, Gal shows that $N \approx 20$ provides a good balance of efficiency and accuracy (in terms of test-time error) for MC dropout with convolutional networks.

We investigate the impact of this parametrization (as opposed to working in azimuth and zenith coordinates directly) later in this chapter. As shown in Table 5.1, we chose a value for the model

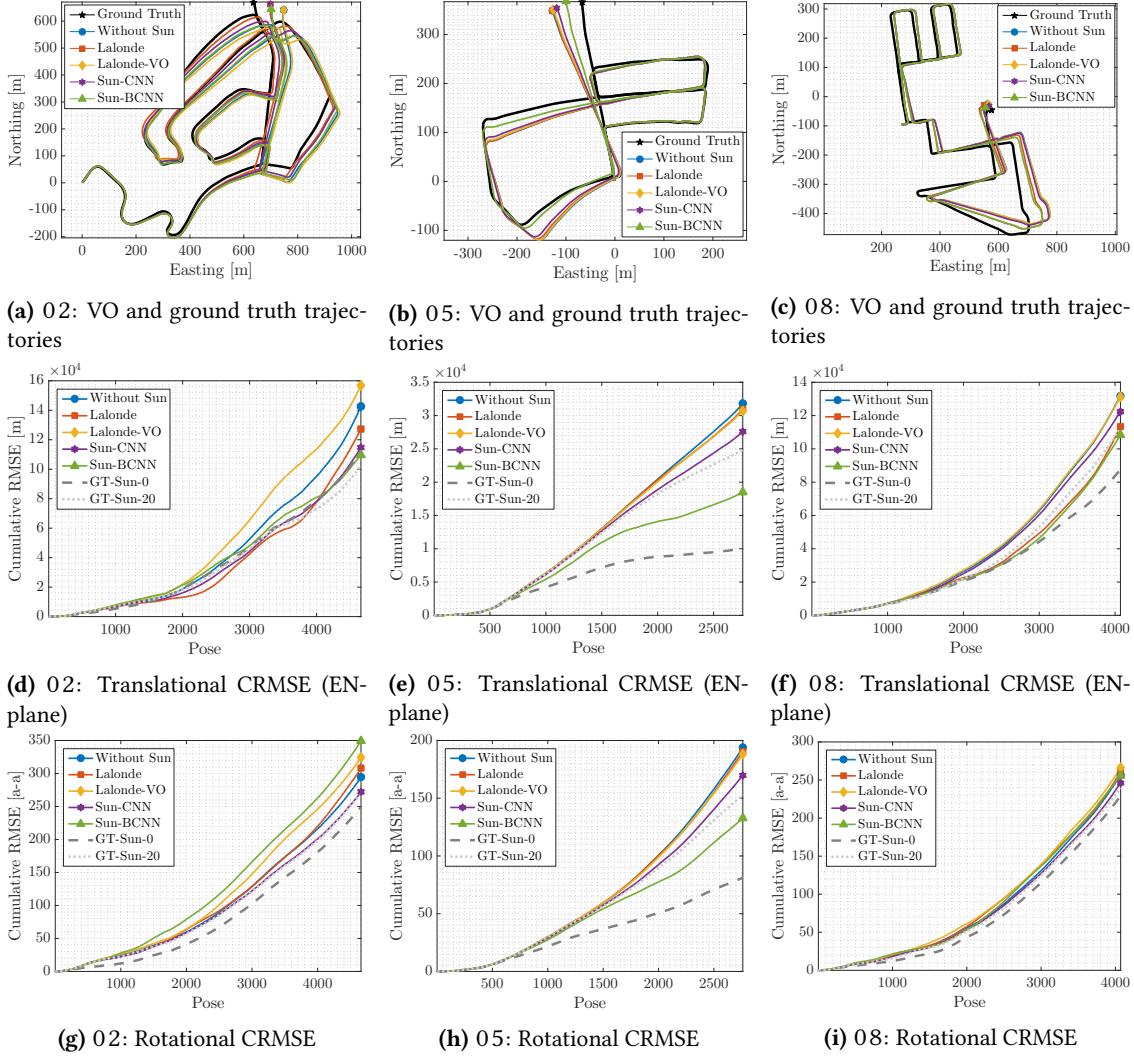


Figure 5.10: VO results for KITTI odometry sequences 02, 05, and 08 using estimate sun directions at every tenth pose. *Top row:* Estimated and ground truth trajectories in the Easting-Northing (EN) plane. *Middle row:* Translational cumulative root mean squared error (CRMSE) in the EN-plane. *Bottom row:* Rotational CRMSE. Sun-BCNN significantly reduces the estimation error on sequence 05, while the Lalonde (Lalonde et al., 2011), Lalonde-VO (Clement et al., 2017), and Sun-CNN (Ma et al., 2016) methods provide modest reductions in estimation error. The remaining sequences are less clear, but Sun-BCNN generally provides some benefit.

Table 5.2: Comparison of translational and rotational average root mean squared error (ARMSE) on KITTI odometry sequences with and without sun direction estimates at every tenth image. The best result (excluding simulated sun sensing) is highlighted in bold.

Sequence ¹	00	01 ²	02	04	05	06	07	08	09	10
Length [km]	3.7	2.5	5.1	0.4	2.2	1.2	0.7	3.2	1.7	0.9
Trans. ARMSE [m]										
Without Sun	4.33	198.52	28.59	2.48	9.90	3.35	4.55	28.05	10.44	5.54
GT-Sun-0	5.40	114.69	23.83	2.23	4.84	3.50	1.58	31.55	8.21	3.67
GT-Sun-10	4.85	123.84	25.34	2.45	5.84	2.80	2.94	28.47	8.65	4.81
GT-Sun-20	4.78	136.60	22.33	2.46	8.16	3.03	3.90	27.54	8.68	5.45
GT-Sun-30	4.83	157.14	27.30	2.48	8.93	3.44	4.62	26.73	10.10	5.28
Lalonde	3.81	200.34	28.13	2.47	9.88	3.36	4.61	29.70	10.49	5.48
Lalonde-VO	4.87	199.03	29.41	2.48	9.74	3.30	4.52	27.82	11.06	5.59
Sun-CNN	4.36	192.50	26.58	2.48	8.92	3.38	4.30	26.99	10.15	5.58
Sun-BCNN	4.44	188.46	26.89	2.48	8.50	4.10	4.21	27.71	10.13	5.61
Trans. ARMSE (EN-plane) [m]										
Without Sun	4.53	230.73	30.66	1.81	11.50	3.68	5.44	32.37	11.65	5.95
GT-Sun-0	3.41	136.76	24.12	1.46	3.67	3.96	1.80	21.51	7.77	3.71
GT-Sun-10	5.05	149.36	24.79	1.79	6.29	2.73	3.51	22.41	8.90	5.09
GT-Sun-20	5.14	164.37	22.04	1.80	9.01	3.13	4.66	27.58	8.86	5.81
GT-Sun-30	5.12	188.61	22.65	1.83	10.31	3.83	5.50	27.65	11.16	5.58
Lalonde	3.95	232.66	27.30	1.81	11.20	3.70	5.52	27.84	11.41	5.87
Lalonde-VO	5.38	231.33	33.68	1.82	11.13	3.61	5.42	32.24	12.41	6.00
Sun-CNN	4.56	224.91	24.65	1.82	9.99	3.74	5.16	30.09	11.21	5.99
Sun-BCNN	4.68	220.54	23.58	1.82	6.70	4.78	5.05	26.59	10.97	6.03
Rot. ARMSE ($\times 10^{-3}$) [axis-angle]										
Without Sun	23.88	185.30	63.18	12.97	70.18	23.24	49.96	63.13	26.77	21.54
GT-Sun-0	11.20	38.82	53.48	11.75	29.38	17.66	20.37	56.39	17.00	12.60
GT-Sun-10	17.05	64.51	58.78	12.86	41.47	18.90	34.05	54.89	19.71	14.26
GT-Sun-20	18.84	94.65	58.03	12.91	55.39	19.67	43.34	58.82	20.99	25.87
GT-Sun-30	23.40	121.21	57.79	13.01	62.73	23.96	49.92	56.74	25.63	20.15
Lalonde	21.10	188.06	66.02	12.96	69.00	23.27	50.49	64.22	26.27	20.49
Lalonde-VO	27.91	185.52	69.52	12.98	68.09	22.79	49.74	65.35	28.82	22.10
Sun-CNN	24.05	177.45	58.32	13.00	61.48	23.34	47.77	60.55	26.19	21.99
Sun-BCNN	26.96	175.21	75.02	13.00	47.96	23.80	47.57	62.85	26.29	20.85

¹ Because we rely on the timestamps and first pose reported by the GPS/INS system, we use the raw (rectified and synchronized) sequences corresponding to each odometry sequence. However, the raw sequence 2011_09_26_drive_0067 corresponding to odometry sequence 03 was not available on the KITTI website at the time of writing, so we omit sequence 03 from our analysis.

² Sequence 01 consists largely of self-similar, corridor-like highway driving which causes difficulties when detecting and matching features using viso2. The base VO result is of low quality, although we note that including global orientation from the sun nevertheless improves the VO result.

precision τ such that the Average Normalized Estimation Error Squared (ANES) of each test sequence is close to one (i.e., the estimator is consistent).

Figures 5.5 and 5.7 plot the error distributions for azimuth, zenith, and angular distance for all ten KITTI odometry sequences, while Figure 5.6 shows three characteristic plots of the azimuth and zenith predictions over time. We see that the errors in azimuth and zenith are strongly peaked around zero and are reasonably well described by a Gaussian distribution, which are important properties assumed by our VO pipeline to produce maximum likelihood motion estimates based on the fusion of multiple data sources. Note that the error distribution in zenith is slightly biased towards negative values due to the presence of a long tail on the negative side of the mean. This is an artifact of the azimuth-zenith parameterization when the sun zenith is small (i.e., when the sun is high in the sky), since zenith angles are defined on $[0, \pi]$. In practice, we attempt to reduce the influence of the long negative tail by imposing a robust Huber loss on the sun measurement errors in our optimization problem.

Table 5.1 summarizes the Sun-BCNN test errors numerically. Sun-BCNN achieved median vector angle errors of less than 15 degrees on every sequence except sequence 01 and 06, which were particularly difficult in places due to challenging lighting conditions. It is interesting to note that sequences 00 and 06 also have higher than average ANES values, which indicates that the estimator is overconfident in its estimates despite their low quality. We suspect this behaviour stems from the assumption of homoscedastic noise in the BCNN, which treats all input images as being equally amenable to sun estimation across the entire sequence.

5.8.2 Visual Odometry Experiments

We evaluated the influence of the estimated sun directions and covariances obtained from Sun-BCNN on the KITTI odometry benchmark using the sun-aided VO pipeline previously described. To place these results in context, we compare them against the results obtained using simulated sun measurements with varying levels of noise, the method of [Lalonde et al. \(2011\)](#) and its VO-informed variant ([Clement et al., 2017](#)), and the Sun-CNN of [Ma et al. \(2016\)](#).

Simulated Sun Sensing

In order to gauge the effectiveness of incorporating sun information in each sequence, and to determine the impact of measurement error, we constructed several sets of simulated sun measurements by computing ground truth sun vectors and artificially corrupting them with varying levels of zero-mean Gaussian noise. We obtained these ground truth sun vectors by transforming the ephemeris vector into each camera frame using ground truth vehicle poses. Using the same convention as our experiments with simulated trajectories, we created four such measurement sets with 0°, 10°, 20°, and 30° mean angular distance from ground truth.

Figure 5.8 shows the results we obtained using simulated sun measurements on sequence 05, in which the basic VO suffers from substantial orientation drift.⁵ Incorporating absolute orientation

⁵In order to make a fair comparison to the Sun-CNN of [Ma et al. \(2016\)](#), who compute sun directions for every tenth image of the KITTI odometry benchmark, we subsample the sun directions obtained through each other method to match.

information from the simulated sun sensor allows the VO to correct these errors, but the magnitude of the correction decreases as sensor noise increases, consistent with the results of our simulation experiments. As shown in Table 5.2, which summarizes our VO results for all ten sequences, this is typical of sequences where orientation drift is the dominant source of error.

While the VO solutions for sequences such as 00 do not improve in terms of translational ARMSE, Table 5.2 shows that rotational ARMSE nevertheless improves on all ten sequences when low-noise simulated sun measurements are included. This implies that the estimation errors of the basic VO solutions for certain sequences are dominated by non-rotational effects, and that the apparent benefit of the Lalonde method on translational ARMSE in sequence 00 is likely coincidental.

Vision-Based Sun Sensing

Figure 5.9 illustrates the behaviour of Sun-BCNN on four characteristic images from test sequence 05 by overlaying the Sun-BCNN predictions and associated ground truth sun directions for each image. The two frames in the top row both contain strong shadows which typically result in very accurate sun predictions. Conversely, the bottom row highlights two examples of rare situations where ambiguous shadows lead to very inaccurate predictions. As previously mentioned, we mitigate the influence of these outlier measurements by imposing a robust Huber loss on the sun measurement errors in our optimizer.

Figure 5.10 shows the results we obtained for sequences 02, 05, and 08 using the Sun-CNN of Ma et al. (2016), which estimates only the azimuth angle of the sun, our Bayesian Sun-BCNN which provides full 3D estimates of the sun direction as well as a measure of the uncertainty associated with each estimate, and the method of Lalonde et al. (2011) in its original and VO-informed (Clement et al., 2017) forms, which provide 3D estimates of the sun direction without reasoning about uncertainty. A selection of results using simulated sun measurements are also displayed for reference. All four sun detection methods succeed in reducing the growth of total estimation error on this sequence, with Sun-BCNN reducing both translational and rotational error growth significantly more than the other three methods. Both Sun-CNN and Sun-BCNN outperform the two Lalonde variants, consistent with the results of Ma et al. (2016) and Clement et al. (2017).

Table 5.2 shows results for all ten sequences using each method. With few exceptions, the VO results using Sun-BCNN achieve improvements in rotational and translational ARMSE comparable to those achieved using the simulated sun measurements with between 10 and 30 degrees average error. As previously noted, sequences such as 00 do not benefit significantly from sun sensing since rotational drift is not the dominant source of estimation error in these cases. Nevertheless, these results indicate that CNN-based sun sensing is a valuable tool for improving localization accuracy in VO and an improvement that comes without the need for additional sensors or a specially oriented camera.

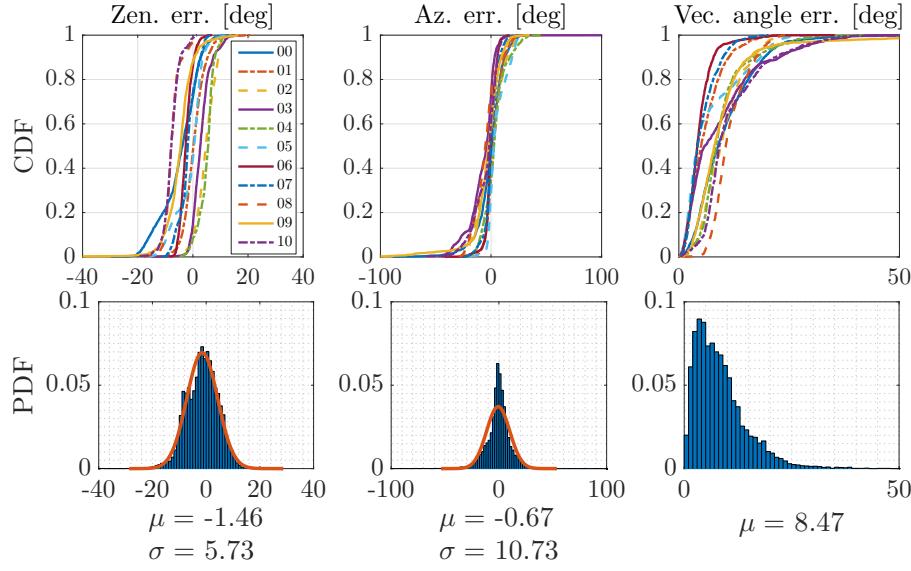


Figure 5.11: (Devon Island) Distributions of azimuth error, zenith error, and angular distance for Sun-BCNN compared to ground truth over each test sequence. *Top row*: Cumulative distributions of errors for each test sequence individually. *Bottom row*: Histograms and Gaussian fits of aggregated errors.

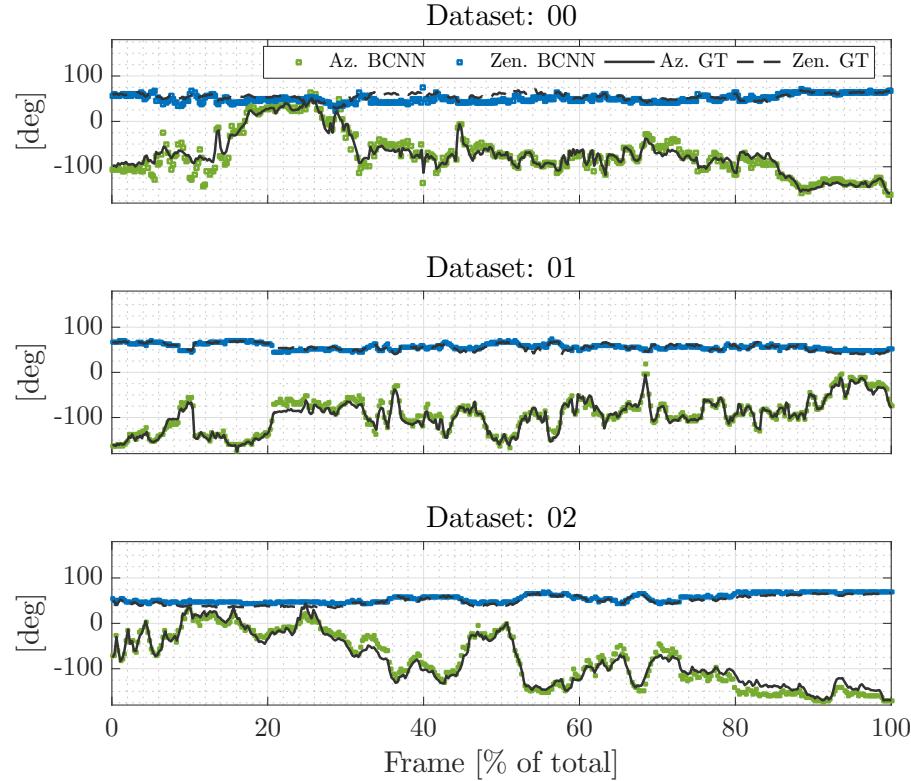


Figure 5.12: Azimuth (Sun-BCNN azimuth and zenith predictions over time for Devon Island test sequences 00, 01 and 12. Sun-BCNN is trained and tested on all frames (in our VO experiments, we use the Sun-BCNN predictions of every tenth image to make a fair comparison).

5.9 Planetary Analogue Experiments: The Devon Island Rover Navigation Dataset

In addition to urban driving, we further investigate the usefulness of Sun-BCNN in the context of planetary exploration using the Devon Island Rover Navigation Dataset ([Furgale et al., 2012](#)), which consists of various sensor data collected using a mobile sensor platform traversing a 10 km loop on Devon Island in the Canadian High Arctic (Figure 5.13).



Figure 5.13: GPS track and sample images from the Devon Island traverse, with the start of each sequence highlighted. The Devon Island dataset is conducive to visual sun sensing due to the presence of strong environmental shadows, reflective surfaces such as mud and water, occasionally visible sun, and self-shadowing by the sensor platform. (Map data: Google, DigitalGlobe)

The rugged landscape of Devon Island (Figure 5.13) is a significant departure from the structured urban environment of Karlsruhe. Unlike the KITTI odometry benchmark, the Devon Island dataset provides ground truth vehicle orientations for only a small number of images, which means that our previous method of generating ground truth sun vectors using ground truth poses is not applicable. However, the sensor platform used to collect the dataset was equipped with a hardware sun sensor and inclinometer, both of which were used by [Lambert et al. \(2012\)](#) to correct VO drift. For our purposes, we ignore the inclinometer and use the sun sensor measurements as training targets for Sun-BCNN.

The Devon Island environment contains many features one might expect to be amenable to visual sun detection. As shown in Figure 5.13, the dataset contains strong environmental shadows, stretches of wet terrain featuring reflective mud and water, and some self-shadowing from the sensor platform itself. At times the sun is partially visible to the camera, although these images tend to be saturated and do not immediately allow for accurate localization of the sun in the image.

For the purposes of our experiments, we partition the dataset into 11 sequences of approximately 1 km each, chosen such that the full pose of the vehicle at the beginning of each sequence is available from the ground truth data (see Figure 5.13). In aggregate, the sequences contain 13257 poses with associated sun sensor measurements. We apply a similar training and testing procedure as for the KITTI dataset, with the exception that we now withhold one sequence for validation and hyper-parameter tuning in addition to the sequence withheld for testing. This leaves nine sequences remaining to form the training sets for each test and validation pair.

Table 5.3: Test Errors for Sun-BCNN on Devon Island odometry sequences with estimates computed at every image.

Sequence	Zenith Error [deg]			Azimuth Error [deg]			Vector Angle Error [deg]			ANEE²
	Mean	Median	Stdev	Mean	Median	Stdev	Mean	Median	Stdev	
00	-4.77	-3.77	6.82	-0.65	0.69	12.41	10.48	8.86	6.96	1.27
01	0.47	0.21	3.91	2.96	2.31	7.01	5.97	5.06	4.01	0.59
02	4.66	4.68	3.52	-0.72	-1.32	11.78	10.02	9.51	4.76	1.37
03	3.09	2.70	3.41	-7.47	-4.03	12.88	9.39	5.83	8.75	1.11
04	4.93	5.53	2.90	3.27	2.72	10.09	9.78	8.41	5.60	0.89
05	-1.01	0.46	4.97	5.26	2.46	8.23	7.19	4.15	6.60	0.92
06	-2.45	-2.58	2.23	-0.23	-0.30	5.07	4.72	4.17	3.16	0.31
07	-1.80	-1.87	3.28	0.47	0.20	6.45	5.23	4.25	3.38	0.41
08	-7.46	-7.88	2.85	-4.93	-5.14	10.30	11.61	10.63	3.96	1.33
09	-4.72	-4.46	5.27	-3.91	-2.13	14.61	9.90	8.02	8.56	0.86
10	-7.69	-7.82	2.92	-4.81	-1.54	10.80	11.79	9.19	7.52	0.91
All	-1.46	-1.23	5.73	-0.67	-0.14	10.73	8.47	7.15	6.31	-

¹ We compute Average Normalized Estimation Error Squared (ANEE²) values with all sun directions that fall below a cosine distance threshold of 0.3 (relative to ground truth) and set $\tau^{-1} = 0.01$.

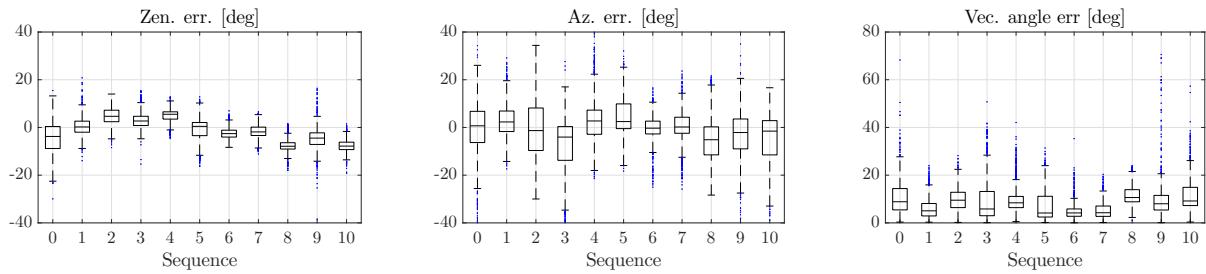


Figure 5.14: Box-and-whiskers plot of final test errors on Devon Island odometry sequences (c.f. Table 5.3).

5.9.1 Sun-BCNN Test Results

As in our experiments with the KITTI odometry benchmark, we obtained the mean estimated sun vector by evaluating Equation (5.28) with $N = 25$ and re-normalizing the resulting vector to preserve unit length. To obtain the required covariance on azimuth and zenith angles, we again sampled the vector outputs, converted them to azimuth and zenith angles using Equation (5.7), and then applied Equation (5.29). As shown in Table 5.3, we chose a value for the model precision τ such that the Average Normalized Estimation Error Squared (ANEES) of each test sequence is close to one (i.e., the estimator is consistent).

Figures 5.11 and 5.14 plot the error distributions for azimuth, zenith, and angular distance for all 11 Devon Island odometry sequences, while Figure 5.12 shows three characteristic plots of the azimuth and zenith predictions over time. We see that the errors in azimuth and zenith are strongly peaked around zero and are better described by a Gaussian distribution than in the case of KITTI (c.f. Figure 5.5), which as we previously mentioned are important properties assumed by our VO pipeline to appropriately fuse data. The distribution of zenith errors in the Devon Island dataset does not exhibit the same bias and long tail we observed in the KITTI dataset. This is likely because the sun is much lower in the sky (i.e., the zenith angle is further from zero) in the Devon Island dataset than in the KITTI dataset, so there is no clipping of the distribution near zero zenith.

Table 5.3 summarizes the test errors and ANEES of each sequence numerically, while Figures 5.11 and 5.14 plot the error distributions for azimuth, zenith, and angular distance for each sequence. Figure 5.12 shows three characteristic plots of the azimuth and zenith predictions over time. Sun-BCNN achieved median vector angle errors of less than 10 degrees on every sequence except sequence 08. Consistent with the results we observed in the KITTI experiments, the sequences with the highest median vector angle error (sequences 02 and 08) also have the highest ANEES values, again indicating that the homoscedastic noise assumption is perhaps ill suited to this environment.

5.9.2 Visual Odometry Experiments

As in our KITTI benchmark experiments, we compare visual odometry results on each of our 11 test sequences both with sun-based orientation corrections and without. Notably, we do not report results using simulated sun measurements since we are unable to generate these measurements without ground truth vehicle orientations for every image. We also do not report results using the Sun-CNN of Ma et al. (2016) since we do not have access to their model. However, we do compare the results obtained using Sun-BCNN to those obtained using the hardware sun sensor as well as the Lalonde (Lalonde et al., 2011) and Lalonde-VO (Clement et al., 2017) methods.

Figure 5.15 shows sample VO results on three sequences from the Devon Island dataset using no sun measurements, the hardware sun sensor, Sun-BCNN, and the Lalonde variants. While the Lalonde methods struggle in this environment, Sun-BCNN yields significant improvements in VO accuracy, nearly on par with those obtained using the hardware sun sensor.

Table 5.4 summarizes these results numerically for all 11 sequences in the dataset. While the addition of sun sensing using either the hardware sensor or Sun-BCNN generally results in significant

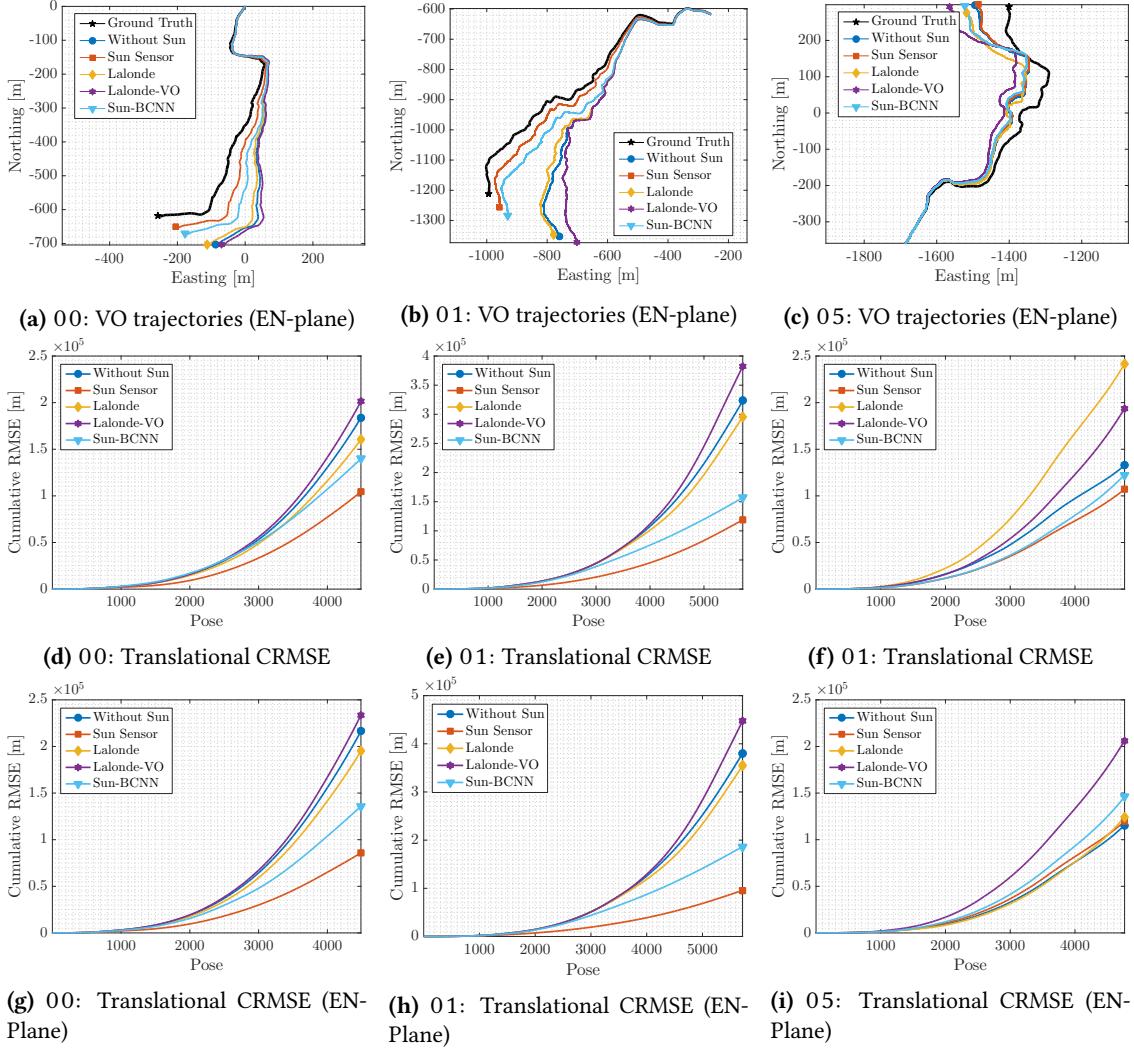


Figure 5.15: VO results for Devon Island sequences 00, 01, and 05 using estimated sun directions. *Top row:* Estimated and ground truth trajectories in the EN-plane. *Bottom rows:* Translational cumulative root mean squared error (CRMSE). Sun-BCNN significantly reduces the estimation error on sequences where the sun sensing has an impact (c.f. ??).

Table 5.4: Comparison of average root mean squared error (ARMSE) on Devon Island sequences with and without sun direction estimates using both a hardware sun sensor and vision-based methods. The best result using a vision-based method is bolded.

Sequence	00	01	02	03	04	05	06	07	08	09	10
Length [km]	0.9	1.1	1.0	1.0	0.9	1.0	1.1	1.0	0.9	0.7	0.6
Trans. ARMSE [m]											
Without Sun	40.93	56.51	41.58	42.04	30.52	27.82	58.91	40.04	47.22	11.39	12.94
Hardware Sun Sensor	23.26	20.79	9.79	22.03	30.79	22.47	24.14	29.59	47.97	6.26	8.50
Lalonde	35.77	51.74	53.32	47.00	39.55	50.70	94.77	59.37	45.78	10.03	16.23
Lalonde-VO	44.83	66.91	44.17	59.84	42.87	40.62	52.16	36.04	50.52	11.34	16.74
Sun-BCNN	31.17	27.45	16.00	26.02	29.34	25.70	33.43	32.25	50.80	4.27	14.92
Trans. ARMSE (EN-plane) [m]											
Without Sun	48.20	66.49	43.58	45.92	31.08	24.23	43.01	22.33	40.85	9.30	15.59
Hardware Sun Sensor	19.13	16.74	8.99	21.18	28.27	25.08	29.27	21.76	28.89	5.14	9.70
Lalonde	43.45	62.03	36.21	49.44	20.13	26.13	53.22	18.10	35.62	6.01	18.45
Lalonde-VO	52.05	78.26	40.20	59.09	50.12	43.28	53.62	42.71	49.99	11.74	20.17
Sun-BCNN	30.28	32.65	9.62	14.32	33.26	30.62	36.44	23.18	13.53	4.45	14.75

reductions in error, we note that in certain cases (e.g., sequence 05), sun sensing has little or no impact on the VO result. We suspect that the translation errors in these cases are dominated by non-rotational effects, similarly to those observed in our experiments with the KITTI dataset, although it is difficult to be certain in the absence of rotational ground truth. As previously mentioned, the incorporation of a motion prior in the VO estimator would likely reduce the impact of these errors.

5.10 Sensitivity Analysis

In this section we analyze the sensitivity of our model to cloud cover, investigate the possibility of model transfer between urban and planetary analogue environments, and examine the impact of different methods for computing the mean and covariance of a norm-constrained vector on the accuracy and consistency of the estimated sun directions.

5.10.1 Cloud Cover

Given that both the KITTI and Devon Island datasets were collected in sunny conditions, it is natural to wonder whether and to what extent Sun-BCNN is affected by cloud cover. As shown in Figure 5.4, Sun-BCNN relies in part on shadows and other local illumination variations to estimate the direction of the sun. Since the diffuse nature of daylight in cloudy conditions tends to soften shadows and other shading variations, one might expect Sun-BCNN to perform worse in cloudy conditions. Accordingly, we investigated the effect of cloud cover on Sun-BCNN using selected sequences from the Oxford Robotcar Dataset (Maddern et al., 2016), which consists of 1000 km of urban driving along a consistent route but in varying weather conditions and at varying times over the course of a year.

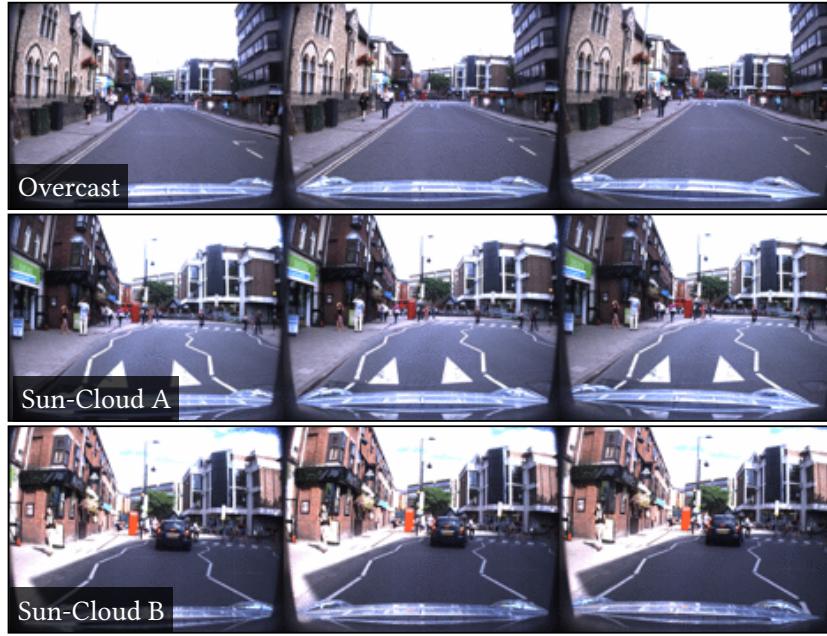


Figure 5.16: Sample images of approximately the same location taken from three different Oxford Robotcar sequences we used to investigate the effect of cloud cover on Sun-BCNN.

Procedure

We selected three sequences collected within a two hour period on the same day (namely 2014-07-14-14-49-50, 2014-07-14-15-16-36, and 2014-07-14-15-42-55), which consist of the same route observed under different lighting conditions. Figure 5.16 presents sample images from each of these sequences, which we label *Overcast*, *Sun-Cloud A*, and *Sun-Cloud B*, respectively. To evaluate the performance of Sun-BCNN in each of these conditions, we partition each sequence into a randomly selected set of training (80%), validation (10%) and test (10%) images, and then train and test Sun-BCNN on each of the nine train-test permutations.

Results

Figure 5.17 shows the results of these experiments with box and whisker plots for azimuth, zenith and vector angle errors while Table 5.5 summarizes the results numerically. We obtained the most accurate test predictions using the model trained on *Sun-Cloud B*, the sequence with the least amount of cloud cover. Notably, this model produced vector angle errors on the *Overcast* test set that were lower than those trained with its own *Overcast* training set. Moreover, we note that the *Sun-Cloud A* model achieved similar test errors when applied to the *Sun-Cloud B* test set as when applied to the *Overcast* test set. Similarly, the *Sun-Cloud B* model achieved similar test errors when applied to the *Sun-Cloud A* test set as when applied to the *Overcast* test set. From this we can conclude the following: 1) that Sun-BCNN can still perform well in the presence of cloud cover; and 2) that training in environments illuminated by strong directional light (i.e., sunny conditions) can significantly improve sun estimation accuracy in different test conditions.

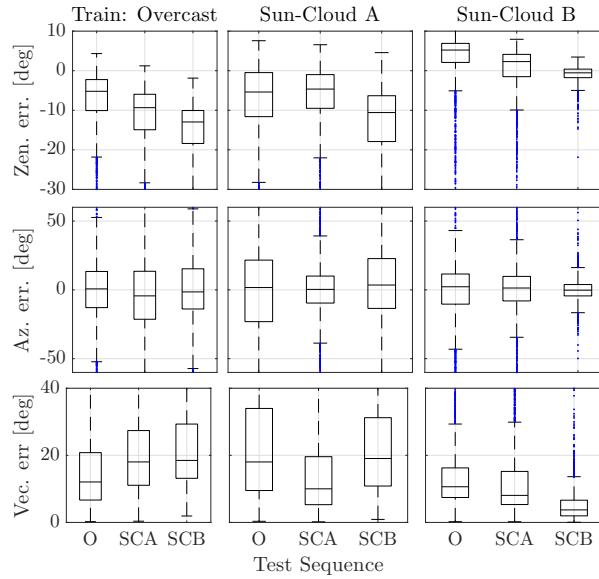


Figure 5.17: Box-and-whiskers plot for zenith, azimuth and vector angle errors for nine different combinations of train-test sequences taken from the Oxford Robotcar dataset. Each column corresponds to a different training sequence, and each plot contains three different test sequences. In the bottom legend, we use the labels O: Overcast, SCA: Sun-Cloud A, SCB: Sun-Cloud B.

5.10.2 Model Generalization

It may also be natural to ask how well a Sun-BCNN model trained in an urban environment performs in a planetary analogue environment and vice versa. This would provide some indication of whether the model generalizes to new environments or if a philosophy of place-specific excellence (e.g., the place-specific visual features of [McManus et al. \(2014\)](#)) is more appropriate for the task of illumination estimation.

Procedure

We attempted to answer this question by creating three larger datasets from combinations of the sequences used in our previous experiments:

1. KITTI odometry sequences 00 - 10;
2. Devon Island sequences 00 - 10; and
3. the previously discussed *Overcast*, *Sun-Cloud A*, and *Sun-Cloud B* sequences from the Oxford Robotcar dataset.

We randomly partitioned each dataset into training (90%) and test (10%) sets. We then trained three separate Sun-BCNN models on each training set, and evaluated each trained model on each of the three test sets.

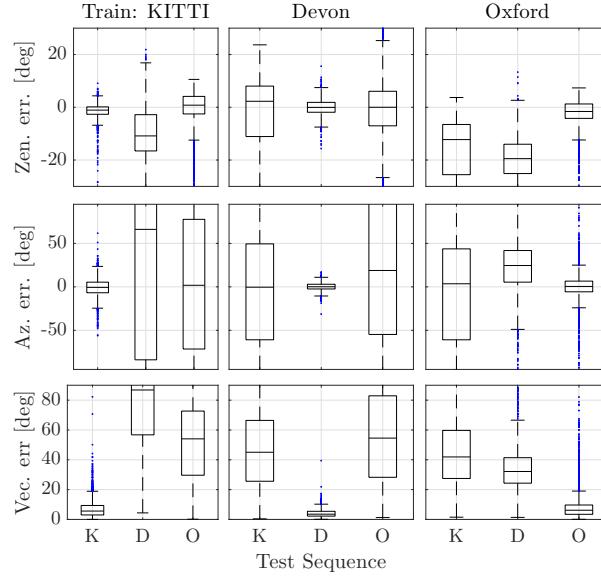


Figure 5.18: Box-and-whiskers plot for zenith, azimuth and vector angle errors for nine different combinations of train-test datasets. Each column corresponds to a different training sequence, and each plot contains three different test sequences. In the bottom legend, we use the labels K: KITTI, D: Devon Island, O: Oxford. All three models produce large biased errors when applied to other datasets, likely due to variations in optical properties and parameter settings across cameras.

Results

Figure 5.18 shows the results of these experiments with box and whisker plots for azimuth, zenith and vector angle errors while Table 5.5 summarizes the results numerically. We see that none of the three models generalize well to environments other than the one in which they were trained, yielding large and significantly biased test errors. We note, however, that the Oxford model was the least egregious offender, and speculate that this may be because the Oxford sequences contain significantly more training images than the other two datasets (approximately 3 times as many as the KITTI odometry benchmark and 5 times as many as the Devon Island dataset).

A possible explanation for the poor generalization of these models is the fact that each dataset was collected using different cameras with different optical properties and parameter settings. We believe these differences affect Sun-BCNN’s ability to recover an accurate estimate of a three dimensional direction vector, since metrically important quantities such as the principal point and focal length of the sensor can vary significantly from camera to camera. Furthermore, differences in dynamic range may also significantly affect the ability of Sun-BCNN to treat shading variations consistently.

5.10.3 Mean and Covariance Computation

In our formulation, Sun-BCNN outputs a sampling of unit-norm 3D vectors. Due to the unit-norm constraint, it is not immediately clear how to apply Equations (5.28) and (5.29) to calculate the mean and covariance of these samples. In this section we present and empirically evaluate two possible procedures for each computation using the previously discussed combined datasets for KITTI, Devon

Table 5.5: Test Errors for Sun-BCNN on three different Oxford Robotcar sequences collected on the same day with different lighting conditions.

Train	Test	Zenith Error [deg]			Azimuth Error [deg]			Vector Error [deg]		
		Mean	Median	Std.	Mean	Median	Std.	Mean	Median	Std.
Overcast ¹	Overcast	-7.12	-5.20	7.04	-0.66	0.72	29.36	15.22	12.06	11.73
	Sun-Cloud A	-11.58	-9.34	7.94	-5.71	-4.37	37.21	21.19	18.03	14.07
	Sun-Cloud B	-15.23	-12.96	8.00	0.05	-1.49	38.83	23.36	18.49	15.05
Sun-Cloud A ²	Overcast	-7.17	-5.39	9.05	-0.67	1.68	51.27	23.66	18.03	18.11
	Sun-Cloud A	-6.49	-4.64	7.88	0.29	0.35	27.42	14.31	10.02	12.75
	Sun-Cloud B	-12.89	-10.58	8.94	1.87	3.51	40.41	23.45	19.06	16.75
Sun-Cloud B ³	Overcast	3.34	5.22	6.46	-0.32	2.24	26.07	13.95	10.63	11.32
	Sun-Cloud A	-0.14	2.30	7.36	-1.08	1.34	28.54	13.76	8.06	14.60
	Sun-Cloud B	-0.84	-0.54	2.07	-0.36	-0.22	9.00	5.11	3.73	5.13

¹ 2014-07-14-14-49-50 ² 2014-07-14-15-16-36 ³ 2014-07-14-15-42-55

Table 5.6: Test Errors for Sun-BCNN on different training and test datasets.

Train	Test	Zenith Error [deg]			Azimuth Error [deg]			Vector Error [deg]		
		Mean	Median	Std.	Mean	Median	Std.	Mean	Median	Std.
KITTI	KITTI	-1.49	-1.08	2.99	-0.64	-0.60	11.46	7.16	5.61	6.23
	Devon Island	-9.27	-10.86	9.97	26.78	66.15	113.23	81.32	86.82	33.48
	Oxford	-0.02	0.80	6.59	-0.44	1.81	91.30	52.39	54.05	29.46
Devon Island	KITTI	-2.37	2.27	14.30	-5.58	-0.38	78.01	48.16	45.06	27.85
	Devon Island	-0.08	-0.05	3.20	0.20	0.12	5.52	4.24	3.52	2.96
	Oxford	-1.35	0.00	11.57	17.12	18.85	96.86	55.52	54.55	29.88
Oxford	KITTI	-17.05	-12.25	13.19	-6.94	3.55	77.70	44.66	41.91	23.00
	Devon Island	-20.07	-19.47	9.81	20.92	24.56	45.52	35.16	32.15	16.07
	Oxford	-1.96	-1.59	4.60	0.19	0.48	15.08	8.08	6.16	7.68

Table 5.7: A comparison of prediction errors from different mean estimation methods.

Sequence	Mean Type	Zenith Error [deg]			Azimuth Error [deg]			Vector Error [deg]		
		Mean	Median	Std.	Mean	Median	Std.	Mean	Median	Std.
KITTI	Method I	-1.50	-1.06	2.96	-0.56	-0.47	11.52	7.16	5.52	6.27
	Method II	-1.06	-0.76	2.44	-0.30	-0.37	30.18	11.49	5.95	18.60
Devon	Method I	-0.07	0.02	3.18	0.19	0.27	5.76	4.22	3.55	3.04
	Method II	0.04	0.09	3.17	1.11	0.26	24.62	9.19	4.05	20.22
Oxford	Method I	-1.97	-1.66	4.59	0.20	0.51	15.31	8.12	6.10	7.74
	Method II	-1.45	-1.27	3.95	-1.58	0.11	34.46	13.18	6.76	19.24

Island, and Oxford.

Mean computation

Procedure We investigated two different methods for computing the mean of the sampled sun vectors, which we refer to as *Method I* and *Method II*.

1. In *Method I* (used in this work), we first evaluate Equation (5.28) directly on the constrained unit vectors produced by N stochastic passes through the BCNN. We then re-normalize the resulting mean vector to enforce unit length, and convert it to azimuth and zenith angles using Equation (5.7).
2. In *Method II*, we first convert each of the N unit vectors produced through stochastic passes through the BCNN to azimuth and zenith angles using Equation (5.7). We then evaluate Equation (5.28) on the angles themselves to obtain the mean in azimuth-zenith coordinates.

We evaluated both methods using the same combined datasets and partitioning scheme as in the transfer learning experiment previously presented.

Results Table 5.7 presents the azimuth, zenith and vector errors for the two mean computation methods. *Method I* produces lower vector errors and smaller standard deviations in azimuth and zenith on all three datasets.

Covariance Computation

Procedure We further investigated two different covariance computation methods, which we also refer to as *Method I* and *Method II*.

1. In *Method I*, we first evaluate Equation (5.29) directly on the constrained unit vectors produced by N stochastic passes through the BCNN, yielding a 3×3 covariance. We then compute a 2×2 covariance on azimuth and zenith by propagating the 3×3 covariance through a linearized Equation (5.7).

Table 5.8: A comparison of ANEES values for different mean and covariance propagation methods.

Sequence	Covariance Type	Mean Type	ANEES
KITTI	Method I	Method I	0.95
		Method II	5.10
	Method II	Method I	1.40
		Method II	0.87
Devon	Method I	Method I	1.29
		Method II	10.05
	Method II	Method I	0.50
		Method II	0.85
Oxford	Method I	Method I	1.50
		Method II	2.14
	Method II	Method I	1.30
		Method II	0.89

2. In *Method II* (used in this work), we first convert each of the N unit vectors produced by stochastic passes through the BCNN to azimuth and zenith angles, and then evaluate Equation (5.29) on the angles themselves.

We once again re-used the transfer learning datasets with the same partitioning scheme, and evaluated covariances on the test sets corresponding to each of the three models. To control for the effect of tuning the model precision τ , we replace the diagonal elements of each covariance matrix with the diagonal elements of the empirical covariance corresponding to the entire test set (computed based ground truth azimuth and zenith errors). We then compared the consistency of the cross-correlations of each method (i.e., the off-diagonal components of the covariance matrix) by computing ANEES values over the each model’s corresponding test set using both mean computation methods.

Results Table 5.8 lists the ANEES values produced by each method of covariance computation when paired with each mean computation method. *Method I* covariances produced better ANEES values when paired with *Method I* mean estimation, but *Method II* covariances paired well with either mean estimation scheme.

5.11 Summary

In this chapter, we detailed an approach to improving the visual egomotion pipeline by learning to extract a latent quantity within an existing image stream. Namely, we presented a deep-learning-based model that could infer the direction of the sun from a single RGB image where the sun may not be directly observable. We leveraged the technique of Monte Carlo dropout to produce a consistent uncertainty estimate for each prediction.

The prediction paired with the uncertainty estimate allowed us to treat our model, named Sun-BCNN, as a ‘pseudo-sensor’ of sun direction whose output could be fused with existing motion estimates much like those from a hardware-based sun sensor. In sum, our contributions with Sun-BCNN were,

1. the application of a Bayesian CNN to the problem of sun direction estimation, incorporating the resulting covariance estimates into a visual odometry pipeline;
2. an empirical demonstration that a Bayesian CNN with dropout layers after each convolutional and fully-connected layer can achieve state-of-the-art accuracy at test time;
3. a loss function that incorporated a 3D unit-length sun direction vector, appropriate for full 6-DOF pose estimation;
4. experimental results on over 30 km of visual navigation data in urban ([Geiger et al., 2013](#)) and planetary analogue ([Furgale et al., 2012](#)) environments;
5. an investigation into the sensitivity of the Bayesian CNN-based sun estimate to cloud cover, camera and environment changes, and measurement parameterization; and
6. an open-source implementation of Sun-BCNN in the Caffe framework⁶.

⁶<https://github.com/utiasSTARS/sun-bcnn-vo>.

Appendices

Bibliography

- Agarwal, S., Mierle, K., et al. (2016). Ceres solver.
- Alberth, 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., Calafio, 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.
- 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 *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.
- 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.