Journal Title
XX(X):1–19

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

Inferring Sun Direction to Improve Visual Odometry: A Deep Learning Approach

Valentin Peretroukhin[1], Lee Clement[1], and Jonathan Kelly

Abstract

We present a method to incorporate global orientation information from the sun into a visual odometry (VO) pipeline using only the existing image stream, where the sun is typicallay not visible. We leverage recent advances in Bayesian Convolutional Neural Networks (BCNNs) to train and implement a sun detection model (dubbed Sun-BCNN) that infers a three-dimensional sun direction vector from a single RGB image. Crucially, our method also computes a principled uncertainty associated with each prediction, using a Monte Carlo dropout scheme. We incorporate this uncertainty into a sliding window stereo visual odometry pipeline where accurate uncertainty estimates are critical for optimal data fusion. We evaluate our method on 21.6 km of urban driving data from the KITTI odometry benchmark training set where it achieves a median error of approximately 12 degrees and yields improvements of up to 42% in translational ARMSE and 32% in rotational ARMSE compared to standard VO. We further evaluate our method on an additional 10 km of visual navigation data from the Devon Island Rover Navigation dataset, achieving a median error of less than 8 degrees and yielding similar improvements in estimation error. In addition to reporting on the accuracy of Sun-BCNN and its impact on VO, 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. An open source implementation of Sun-BCNN using Caffe is available at https://github.com/utiasSTARS/sun-bcnn-vo.

Keywords

Illumination Estimation, Sun Sensing, Deep Learning, Visual Odometry, Robot Navigation

Introduction

A crucial competency of any autonomous mobile robot is the ability of the platform to estimate its own motion through its operating environment. While there exists a rich body of literature on the topic of motion estimation using a variety of techniques such as lidar-based point cloud matching (Zhang and Singh 2015) and visual-inertial odometry (Leutenegger et al. 2015), egomotion estimation is fundamentally a process of dead-reckoning and will accumulate unbounded error over time. This accumulated error, or drift, can be limited by incorporating global information into the motion estimation problem. This frequently takes the form of a globally consistent map, loop closure detection, or reliance on additional sensors such as GPS to make corrections to the estimated trajectory. In many situations, however, a globally consistent map may be unavailable or prohibitively expensive to compute, loop closures may not occur, or GPS may be unavailable or inaccurate. In such cases, it can be advantageous to rely on environmental cues such as the sun, which can easily provide global orientation information since it is readily detectable and its apparent motion in the sky is well described by ephemeris models.

For visual odometry (VO) in particular, 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). Sun-based orientation corrections have been successfully used in planetary

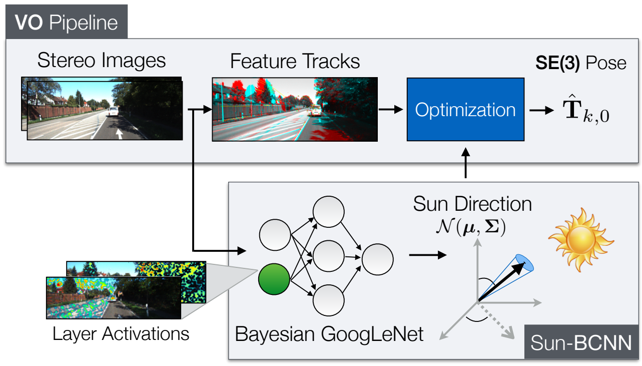


Figure 1. Our method uses a Bayesian Convolutional Neural Network (BCNN) to estimate the direction of the sun, and also produce a principled uncertainty estimate for each prediction. We incorporate this *virtual sun sensor* into a stereo visual odometry pipeline to reduce estimation error.

[1]Valentin Peretroukhin and Lee Clement contributed equally to this work and jointly assert first authorship. All authors are with the Space & Terrestrial Autonomous Robotic Systems (STARS) laboratory at the University of Toronto Institute for Aerospace Studies (UTIAS), Canada

Corresponding author:
Valentin Peretroukhin and Lee Clement, University of Toronto Institute for Aerospace Studies (UTIAS), Toronto, Canada
Email: {v.peretroukhin, lee.clement}@mail.utoronto.ca

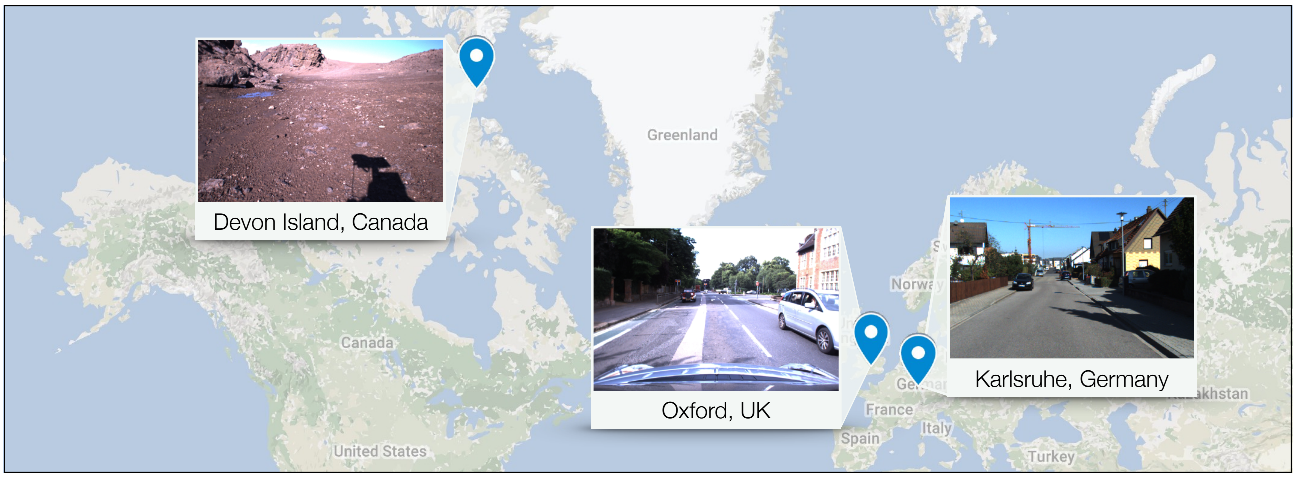


Figure 2. We train and test Sun-BCNN in a variety of environments ranging from urban driving in Europe to remote planetary analogue sites in the Canadian High Arctic. (Map data: Google, INEGI, ORION-ME.)

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 (as opposed to periodically updating the vehicle heading, as in earlier work) can significantly reduce VO drift over long trajectories.

In this work, we seek to answer the question of whether similar reductions in VO drift can be obtained solely from the image stream already being used to compute VO. The main idea here is that by reasoning over more than just the geometric information available from a standard RGB camera, we can improve existing VO techniques without needing to rely on a dedicated sun sensor or specially oriented camera. In particular, we leverage recent advances in Bayesian Convolutional Neural Networks (BCNNs) to demonstrate how we can build and train a deep model capable of inferring the direction of the sun from a single RGB image. Moreover, we show that our network, dubbed Sun-BCNN, can produce a principled covariance estimate for each observation that can readily be used to replace the hand-tuned or empirically computed static covariance typically used to fuse data in a motion estimation pipeline.

Our main contributions can be summarized as follows:

1. We apply a Bayesian CNN to the problem of sun direction estimation, incorporating the resulting covariance estimates into a visual odometry pipeline;
2. We show 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. We learn a 3D unit-length sun direction vector, appropriate for full 6-DOF pose estimation;
4. We present experimental results on over 30 km of visual navigation data in urban (Geiger et al. 2012) and planetary analogue (Furgale et al. 2012) environments;
5. We investigate the sensitivity of our Bayesian CNN-based sun estimator (Sun-BCNN) to cloud cover, camera and environment changes, and measurement parameterization; and
6. We release Sun-BCNN as open-source code.

The remainder of this paper 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 a two-frame bundle adjustment optimization, 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. 2012), as well as a further 10 km traverse through a planetary analogue site taken from the Devon Island Rover Navigation Dataset (Furgale et al. 2012). 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). We also 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.

Related Work

Visual odometry (VO), a technique to estimate the motion of a moving platform equipped with one or more cameras, has a rich history of research including a notable implementation onboard the Mars Exploration Rovers (MERs) (Scaramuzza and Fraundorfer 2011). Modern approaches to VO can achieve estimation errors below 1% of total distance traveled (Geiger et al. 2013). To achieve such accurate and robust estimates, modern techniques use careful visual feature pruning (Cvišić and Petrović 2015), adaptive robust methods (Alcantarilla and Woodford 2016; Peretroukhin et al. 2016), or operate directly on pixel intensities (Engel et al. 2015).

Independent of the estimator, VO exhibits superlinear error growth, and is particularly sensitive to errors in orientation (Olson et al. 2003; Cvišić and Petrović 2015). One way to reduce orientation error is to incorporate observations of a landmark whose position or direction in the navigation frame is known *a priori*. The sun is an example of such a known directional landmark. Accordingly, sun sensors have been used to improve the accuracy of VO in planetary analogue environments (Furgale et al. 2011; Lambert et al. 2012), and were also incorporated into the MERs (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. 2016). 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. 2016), while others have attempted to learn such cues from data using deep Convolutional Neural Networks (CNNs) (Ma et al. 2017).

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. 2017; Kendall et al. 2015; Kendall and Cipolla 2016). Despite their success in improving prediction accuracy, most existing CNN-based models do not report principled uncertainty estimates, which are important in the context of data fusion. To address this, Gal and Ghahramani (2016b) showed that it is possible to achieve principled covariance outputs 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 on camera pose regression.

We build on on previous work by Clement et al. (2016), 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. 2017) 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), in which we presented preliminary experimental results comparing our Sun-BCNN against the method of Lalonde et al. (2011) and its VO-informed variant (Clement et al. 2016) as well as the Sun-CNN of Ma et al. (2017) on the KITTI odometry benchmark (Geiger et al. 2012, 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. (2017), 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 estimation problems (as opposed to only the azimuth angle and 3-DOF estimator used by Ma et al. (2017)); 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. (2016)). 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)), which produces more principled covariance outputs.

Indirect Sun Detection using a Bayesian Convolutional Neural Network

We use a Bayesian Convolutional Neural Network (BCNN) 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. (2016) and Ma et al. (2017), 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,?). Unlike Ma et al. (2017), 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.

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:

$$\mathcal{L}\left(\hat{\mathbf{s}}_k\right) = 1 - \left(\hat{\mathbf{s}}_k \cdot \mathbf{s}_k\right), \tag{1}$$

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 (1) since both vectors have unit length:

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

Uncertainty Estimation

Following recent work on Bayesian Convolutional Neural Networks (BCNNs) (Gal and Ghahramani 2016a,b; Gal 2016), we modify our model architecture to enable the computation of principled covariance estimates associated with each predicted sun direction. To achieve this, we rely on a connection between stochastic regularization (e.g., dropout, a widely used technique in deep learning to mitigate overfitting) and approximate variational inference of a Bayesian Neural Network. We outline the technique here briefly, and refer the reader to Gal and Ghahramani (2016a) for more details.

The method begins with a prior $p(\boldsymbol{w})$ on the weights in a deep neural network and attempts to compute a posterior distribution $p(\boldsymbol{w}|\mathbf{X}, \mathbf{S})$ given training inputs $\mathbf{X} = \{\mathbf{x}_k\}$ and

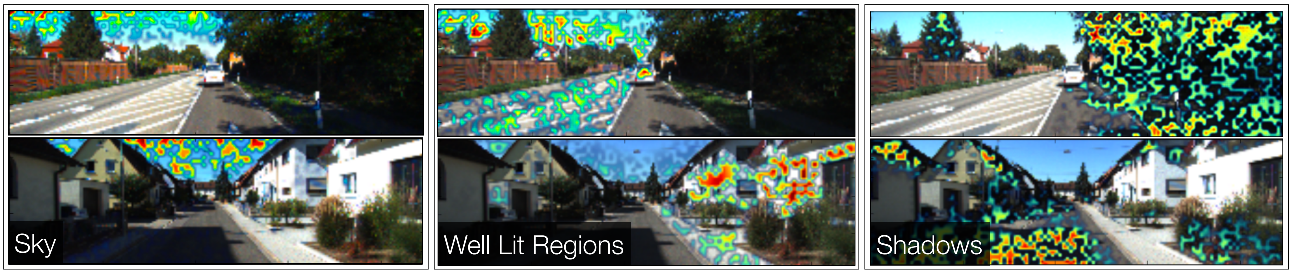


Figure 3. Three `conv1` layer activation maps superimposed on two images from the KITTI odometry benchmark (Geiger et al. 2012) `00` and `04` for three selected filters. Each filter picks out salient parts of the image that aid in sun direction inference.

targets $\mathbf{S} = \{\mathbf{s}_k\}$. This posterior can be used to compute a predictive distribution for test samples but is generally intractable. To overcome this, the BCNN approach notes that CNN training with stochastic regularization can be viewed as variational inference if we define a variational distribution $q(\boldsymbol{w})$ as:

$$q(\boldsymbol{w}_i) = \mathbf{M}_i \mathrm{diag}\left\{\{b^i_j\}_{j=1}^{K_i}\right\}, \tag{2}$$

$$b^i_j \in \mathrm{Bernoulli}(p_i). \tag{3}$$

Here, i indexes a particular layer in the neural network with K_i weights, $\mathbf{M}$ are the weights to be optimized, b^i_j are Bernoulli distributed binary variables, and p_i is the dropout probability for weights in layer i.

With this variational distribution $q(\boldsymbol{w})$, training a CNN with dropout is analogous to minimizing $\mathrm{KL}(p(\boldsymbol{w}|\mathbf{X},\mathbf{S}) \;||\; q(\boldsymbol{w}))$, the Kullback-Leibler (KL) divergence between the variational distribution and the true posterior. At test time, the first two moments of the predictive distribution are approximated using Monte Carlo integration over the weights $\boldsymbol{w}$:

$$\mathbb{E}\left[\hat{\mathbf{s}}^*\right]_k = \hat{\bar{\mathbf{s}}}^*_k \approx \frac{1}{N}\sum_{n=1}^{N} \hat{\mathbf{s}}^*_k(\mathbf{x}^*_k, \boldsymbol{w}^n) \tag{4}$$

$$\mathbb{E}\left[\hat{\mathbf{s}}^*_k \, \hat{\mathbf{s}}^{*}{}^T_k\right] \approx \tau^{-1}\mathbf{1} + \frac{1}{N}\sum_{n=1}^{N} \hat{\mathbf{s}}^*_k(\mathbf{x}^*_k, \boldsymbol{w}^n)\; \hat{\mathbf{s}}^*_k(\mathbf{x}^*_k, \boldsymbol{w}^n)^T - \hat{\bar{\mathbf{s}}}^*_k \, \hat{\bar{\mathbf{s}}}^{*}{}^T_k, \tag{5}$$

where $\mathbf{1}$ is the identity matrix, and $\boldsymbol{w}^n$ is a sample from $q(\boldsymbol{w})$ (obtained by sampling the network with dropout). The model precision, τ, is computed as

$$\tau = \frac{pl^2}{2M\lambda}, \tag{6}$$

where p is the dropout probability, l is the characteristic length scale, M is the number of samples in the training data, and λ is the weight decay.

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 (4) and (5).

Implementation and Training

We implement our network in Caffe (Jia et al. 2014), using the `L2Norm` layer from the Caffe-SL fork* to handle vector normalization. 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 3 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 image size expected by GoogLeNet. 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. (2017)) results in better test-time accuracy. We perform no additional cropping or rotating of the images.

Model Precision We find an empirically optimal model precision τ (see Equation (6)) by optimizing the Average Normalized Estimation Error Squared (ANEES) across the entire test set for each dataset. While this hyperparameter should in principle be tuned using a validation set, we omit this step to keep our training procedure consistent with that of Ma et al. (2017). 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.

Sun-Aided Stereo Visual Odometry

We adopt a sliding window sparse stereo VO technique 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 $\mathrm{SE}(3)$ poses $\{\mathbf{T}_{k_1,0}, \ldots, \mathbf{T}_{k_2,0}\}$ expressed in a base coordinate frame $\underrightarrow{\mathcal{F}}_0$, 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

*https://github.com/wanji/caffe-sl

observed to provide good VO accuracy at low computational cost. We select the initial pose $\mathbf{T}_{1,0}$ to be the first GPS ground truth pose such that $\underrightarrow{\mathcal{F}}_0$ is a local East-North-Up (ENU) coordinate system with its origin at the first GPS position.

Observation Model

We assume that incoming stereo images have been de-warped 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. If we take $\mathbf{p}_0^j$ to be the homogeneous 3D coordinates of keypoint j, expressed in our chosen base frame $\underrightarrow{\mathcal{F}}_0$, we can transform the keypoint into the camera frame at pose k to obtain $\mathbf{p}_k^j = \mathbf{T}_{k,0}\mathbf{p}_0^j = \begin{bmatrix} p_{k,x}^j & p_{k,y}^j & p_{k,z}^j & 1 \end{bmatrix}^T$. Our observation model $\mathbf{g}(\cdot)$ can then be formulated as

$$\mathbf{y}_{k,j} = \mathbf{g}\left(\mathbf{p}_k^j\right) = \begin{bmatrix} u \\ v \\ d \end{bmatrix} = \begin{bmatrix} f_u p_{k,x}^j / p_{k,z}^j + c_u \\ f_v p_{k,y}^j / p_{k,z}^j + c_v \\ f_u b / p_{k,z}^j \end{bmatrix}, \tag{7}$$

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

Sliding Window Bundle Adjustment

We use the open-source `libviso2` 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

$$\mathcal{J} = \mathcal{J}_{\text{reprojection}} + \mathcal{J}_{\text{prior}}, \tag{8}$$

where

$$\mathcal{J}_{\text{reprojection}} = \sum_{k=k_1}^{k_2} \sum_{j=1}^{J} \mathbf{e}_{\mathbf{y}_{k,j}}^T \mathbf{R}_{\mathbf{y}_{k,j}}^{-1} \mathbf{e}_{\mathbf{y}_{k,j}} \tag{9}$$

and

$$\mathcal{J}_{\text{prior}} = \mathbf{e}_{\hat{\mathbf{T}}_{k_1,0}}^T \mathbf{R}_{\hat{\mathbf{T}}_{k_1,0}}^{-1} \mathbf{e}_{\hat{\mathbf{T}}_{k_1,0}}. \tag{10}$$

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 $\hat{\mathbf{y}}_{k,j} = \mathbf{g}\left(\hat{\mathbf{T}}_{k,0}\hat{\mathbf{p}}_0^j\right)$, where $\hat{\mathbf{T}}_{k,0}$ and $\hat{\mathbf{p}}_0^j$ are the estimated poses and keypoint positions in base frame $\underrightarrow{\mathcal{F}}_0$.

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 using the $\mathrm{SE}(3)$ matrix logarithm as $\mathbf{e}_{\hat{\mathbf{T}}_{k_1,0}} = \log\left(\check{\mathbf{T}}_{k_1,0}^{-1}\hat{\mathbf{T}}_{k_1,0}\right)$. The 6×6 matrix $\mathbf{R}_{\hat{\mathbf{T}}_{k_1,0}}$ 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.

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 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 $\underrightarrow{\mathcal{F}}_0$ 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 $\underrightarrow{\mathcal{F}}_k$ 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 (8)):

$$\mathcal{J} = \mathcal{J}_{\text{reprojection}} + \mathcal{J}_{\text{prior}} + \mathcal{J}_{\text{sun}}, \tag{11}$$

where

$$\mathcal{J}_{\text{sun}} = \sum_{k=k_1}^{k_2} \mathbf{e}_{\mathbf{s}_k}^T \mathbf{R}_{\mathbf{s}_k}^{-1} \mathbf{e}_{\mathbf{s}_k}, \tag{12}$$

and $\mathcal{J}_{\text{reprojection}}$ and $\mathcal{J}_{\text{prior}}$ are defined in Equations (9) and (10), 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).

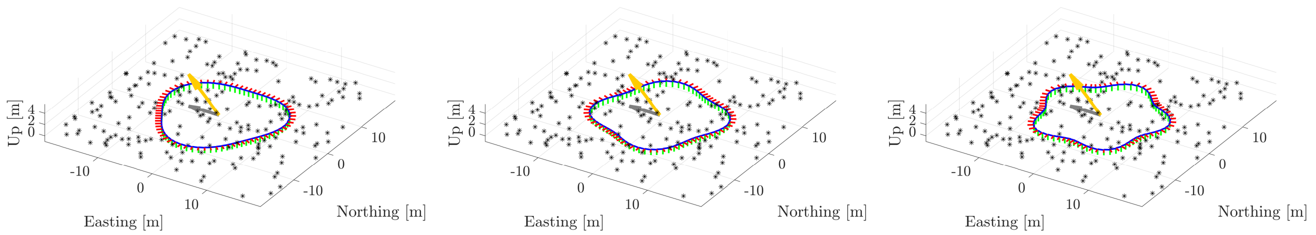
Since $\mathbf{s}_k$ is constrained to be unit length, there are only two underlying degrees of freedom. We therefore define $\mathbf{f}(\cdot)$ to be a function that transforms a 3D unit vector in camera frame $\underrightarrow{\mathcal{F}}_k$ to a zenith-azimuth parametrization:

$$\begin{bmatrix} \theta \\ \phi \end{bmatrix} = \mathbf{f}(\mathbf{s}_k) = \begin{bmatrix} \operatorname{acos}(-s_{k,y}) \\ \operatorname{atan2}(s_{k,x}, s_{k,z}) \end{bmatrix} \tag{13}$$

where $\mathbf{s}_k = \begin{bmatrix} s_{k,x} & s_{k,y} & s_{k,z} \end{bmatrix}^T$. 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), 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.

Simulation Experiments

We assess the benefit of incorporating sun observations of varying quality by conducting a series of simulation



(a) "Triangle" trajectory **(b)** "Square" trajectory **(c)** "Star" trajectory

Figure 4. One loop of the "Triangle", "Square", and "Star" trajectories, consisting primarily of translation and yaw rotation. Landmarks are shown as black asterisks, and the simulated sun direction is indicated with a yellow arrow along with its projection, in grey, on the EN-plane.

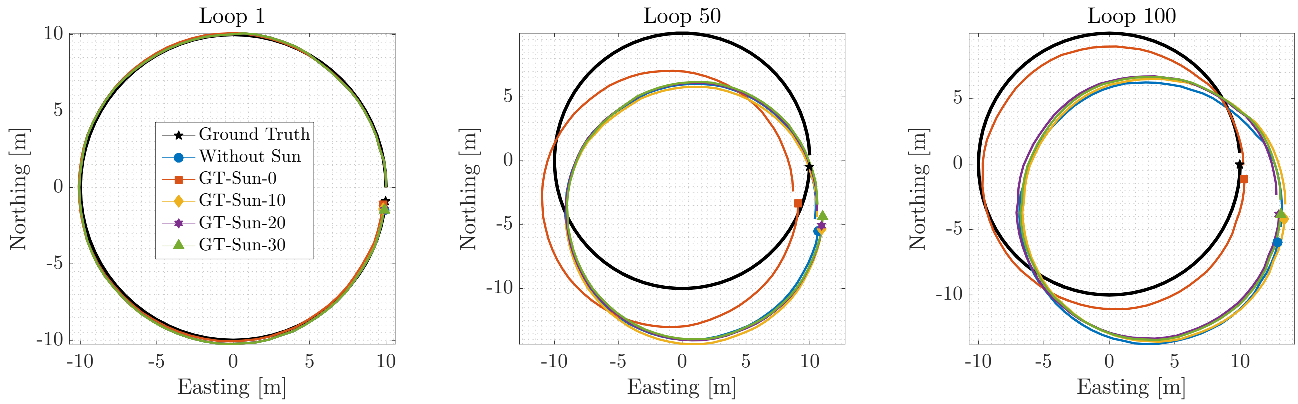
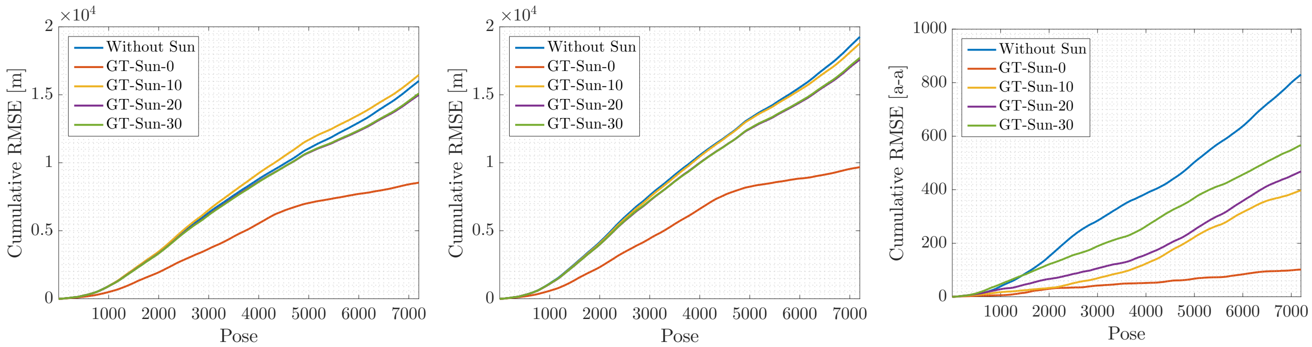


Figure 5. Selected segments of a 100-loop "Circle" trajectory, without sun corrections, and with sun corrections corrupted by varying levels of artificial Gaussian noise. The effect of VO drift can be clearly seen, as well as the benefit of incorporating observations of a directional landmark such as the sun.



(a) Translational CRMSE (ENU) **(b)** Translational CRMSE (EN plane) **(c)** Rotational CRMSE

Figure 6. Cumulative root mean squared error (CRMSE) of a simulated 100-loop circular trajectory, without sun corrections, and with sun corrections corrupted by varying levels of artificial Gaussian noise. The accumulated estimation error is greatly reduced by incorporating observations of the sun, and the benefit decreases as these observations become noisier.

experiments consisting of a stereo camera moving along loopy trajectories of varying shapes through a simulated field of point landmarks, with a single static directional landmark representing the sun. Figure 4 shows several such loopy trajectories. We simulate the sun at 45° of zenith and an arbitrary azimuth angle, and corrupt observations of the ground truth sun vector with artificial noise such that the mean angular distance between the observed and true sun direction is 0°, 10°, 20°, and 30°, labeling these conditions *GT-Sun-0*, *GT-Sun-10*, *GT-Sun-20*, and *GT-Sun-30*, respectively. In our experiments, we treated the measurement noise as an additive quantity sampled from a zero-mean isotropic 3D Gaussian distribution, and renormalized the resulting vectors to enforce the unit-norm constraint. We also experimented with sampling simulated measurements from a Von Mises-Fisher distribution (Fisher 1953), which is approximately analogous to an isotropic Gaussian distribution that respects the geodesics on the unit 2-sphere. However, we observed that the resulting distributions on azimuth and zenith error were severely non-Gaussian, which violated the assumption of zero-mean Gaussian noise in our VO pipeline and interfered with our VO experiments.

Since our VO pipeline does not incorporate loop closures, the effects of drift in the VO solution can be clearly seen by examining individual loops in the camera trajectory. Figure 5 shows three loops from the "Circle" trajectory, demonstrating that the VO solution drifts significantly from

Table 1. Comparison of translational and rotational average root mean squared errors (ARMSE) on simulated sequences.

Loop Shape	Circle	Triangle	Square	Star
# Loops	100	100	100	100
Trans. ARMSE [m]				
Without Sun	2.22	2.00	2.33	1.41
GT-Sun-0	1.19	1.62	2.13	0.75
GT-Sun-10	2.29	2.07	2.05	1.32
GT-Sun-20	2.08	2.12	2.31	1.33
GT-Sun-30	2.10	1.95	2.16	1.38
Trans. ARMSE (EN-plane) [m]				
Without Sun	2.67	1.88	2.57	1.10
GT-Sun-0	1.34	1.89	2.56	0.83
GT-Sun-10	2.61	2.04	2.26	0.99
GT-Sun-20	2.44	2.03	2.57	0.88
GT-Sun-30	2.46	2.00	2.35	1.25
Rot. ARMSE ($\times 10^{-3}$) [axis-angle]				
Without Sun	115.32	144.56	107.27	111.19
GT-Sun-0	14.10	113.58	59.21	30.69
GT-Sun-10	55.22	115.03	75.62	39.17
GT-Sun-20	65.02	121.11	80.41	49.75
GT-Sun-30	78.73	145.22	100.91	72.39

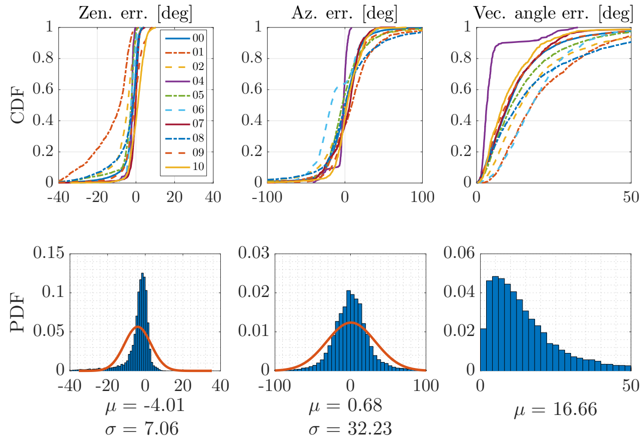


Figure 7. Distributions of azimuth error, zenith error, and angular distance for Sun-BCNN compared to ground truth over each test sequence in the KITTI dataset. *Top row*: Cumulative distributions of errors for each test sequence individually. *Bottom row:* Histograms and Gaussian fits of aggregated errors.

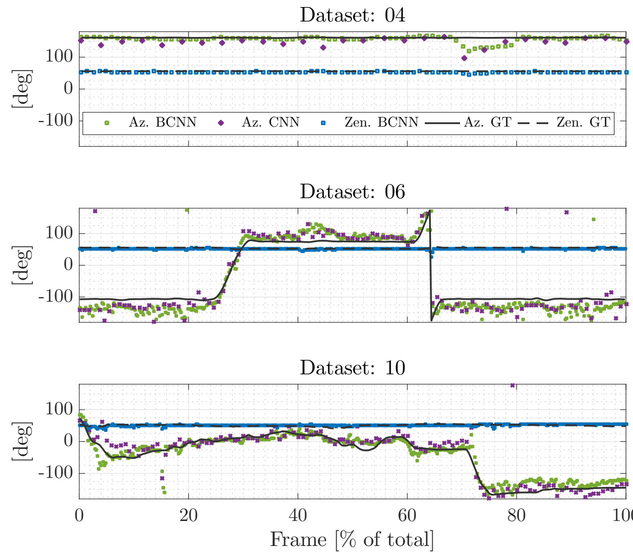


Figure 8. Azimuth (Sun-CNN and Sun-BCNN) and zenith (Sun-BCNN only) predictions over time for KITTI test sequences `04`, `06` and `10`. Sun-CNN is trained and tested on every tenth image, whereas Sun-BCNN is trained and tested on every image. In our VO experiments, we use the Sun-BCNN predictions of every tenth image to make a fair comparison.

the true trajectory by the 100th loop. Figure 6 plots the translational and rotational cumulative root mean squared error (CRMSE) for this trajectory, which measures the growth in total estimation error over time. Figure 6c in particular highlights the significant effect of sun sensing on rotational error, where we see a clear progression in estimation error as the sun direction observations become more noisy.

Table 1 shows that while all four simulation trajectories display consistent and predictable reductions in rotational average root mean squared error (ARMSE), this is not always the case for translational ARMSE. This is because translational errors are only partially induced by rotational errors, with the remainder made up of 'sliding' motions orthogonal to the direction of travel. These non-rotational errors are highly dependent on the specific trajectory, where more or less of the observed feature tracks can be explained by a sliding motion instead of a rotation. While we do not implement this in our work, we speculate that incorporating an appropriate motion model into our VO formulation would significantly mitigate the impact of these errors by, for example, imposing a nonholonomic constraint on a ground vehicle or accounting for the dynamics of a quadcopter.

Urban Driving Experiments: The KITTI Odometry Benchmark

We investigated the performance of Sun-BCNN on the KITTI odometry benchmark training set (Geiger et al. 2012, 2013), which consists of 21.6 km of urban driving data[†]. Importantly, the dataset includes 6-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 $\underrightarrow{\mathcal{F}}_0$ (which is an ENU coordinate system) into the camera coordinate frame $\underrightarrow{\mathcal{F}}_k$. 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.

[†]Because 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.

Table 2. Test Errors for Sun-BCNN on KITTI odometry sequences with estimates computed at every image.

	Zenith Error [deg]			Azimuth Error [deg]			Vector Angle Error [deg]			
Sequence	Mean	Median	Stdev	Mean	Median	Stdev	Mean	Median	Stdev	**ANEES**[1]
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 (ANEES) values with all sun directions that fall below a cosine distance threshold of 0.3 (relative to ground truth) and set $\tau^{-1} = 0.015$.

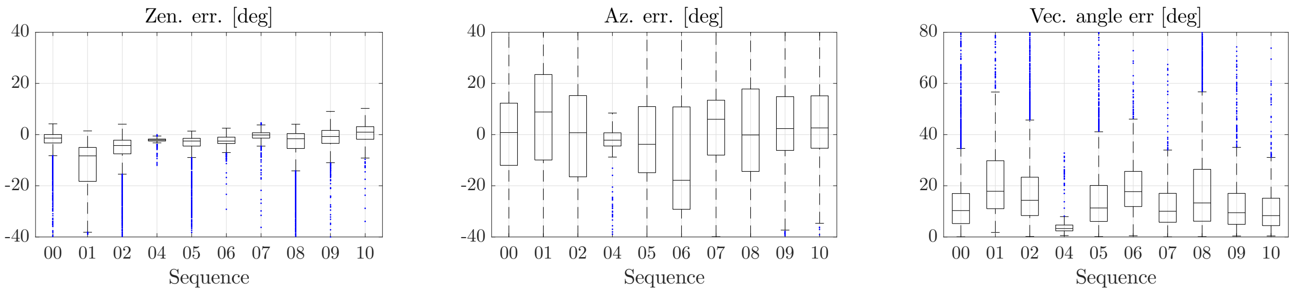
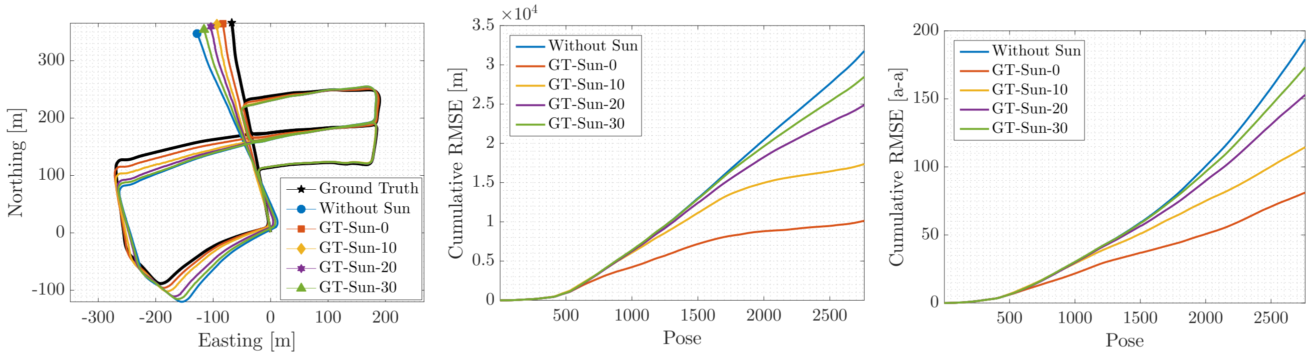


Figure 9. Box-and-whiskers plot of final test errors on all ten KITTI odometry sequences (c.f. Table 2).



(a) VO and ground truth trajectories **(b)** Translational CRMSE (EN-plane) **(c)** Rotational CRMSE

Figure 10. 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.

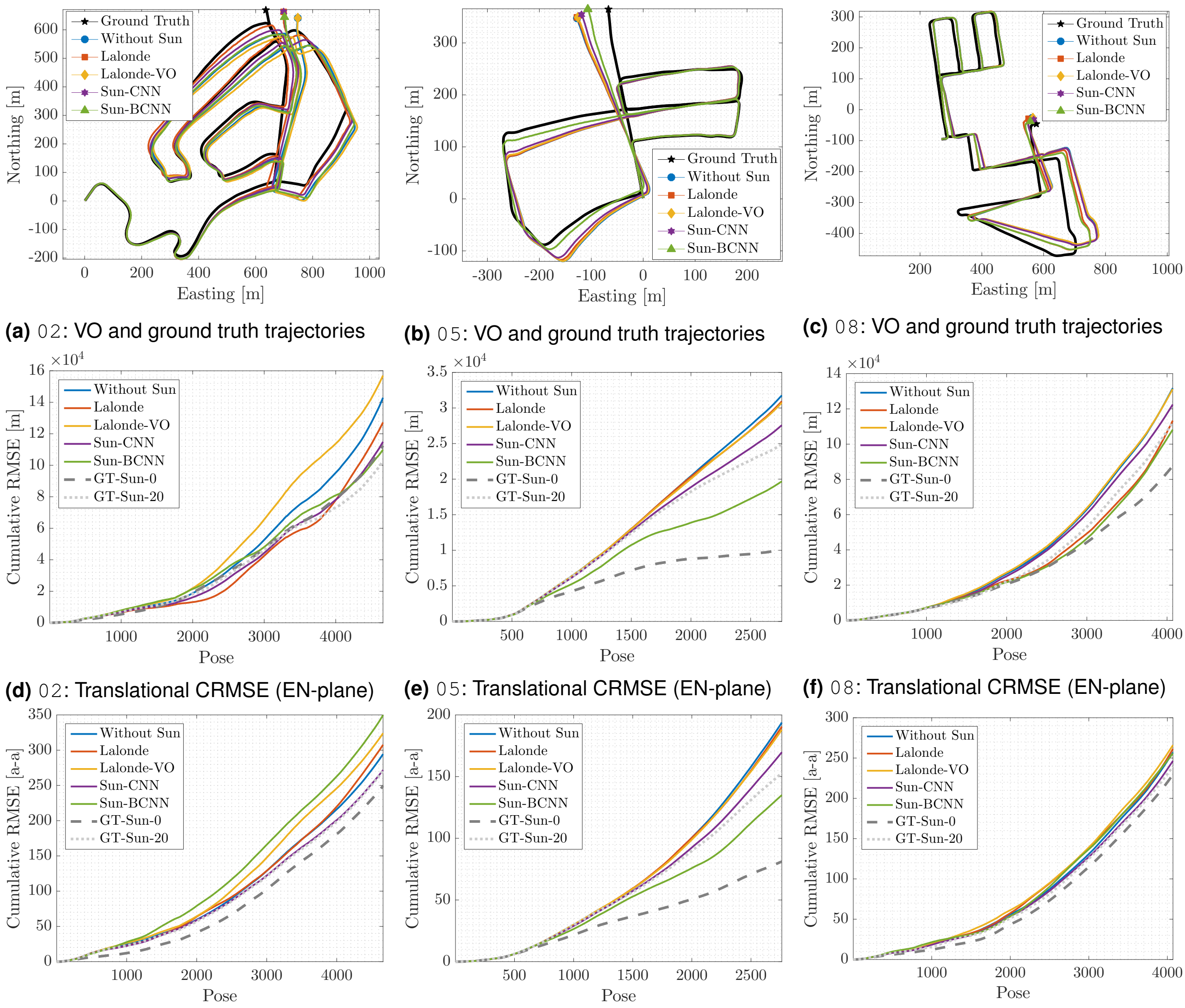
(2017), against whose Sun-CNN we directly compare, and allows us to evaluate each sequence using the maximum amount of training data.

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 (4) 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 (13), and then applied Equation (5). We investigate the impact of this parametrization (as opposed to working in azimuth and zenith coordinates directly) later in this paper. As shown in Table 2, 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 7 and 9 plot the error distributions for azimuth, zenith, and angular distance for all ten KITTI odometry sequences, while Figure 8 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

Figure 11. 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.



(a) `02`: VO and ground truth trajectories

(b) `05`: VO and ground truth trajectories

(c) `08`: VO and ground truth trajectories

(d) `02`: Translational CRMSE (EN-plane)

(e) `05`: Translational CRMSE (EN-plane)

(f) `08`: Translational CRMSE (EN-plane)

(g) `02`: Rotational CRMSE

(h) `05`: Rotational CRMSE

(i) `08`: Rotational CRMSE

Figure 12. 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. 2016), and Sun-CNN (Ma et al. 2017) methods provide modest reductions in estimation error. The remaining sequences are less clear, but Sun-BCNN generally provides some benefit.

Table 3. 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 `libviso2`. The base VO result is of low quality, although we note that including global orientation from the sun nevertheless improves the VO result.

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 2 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 ANEES 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.

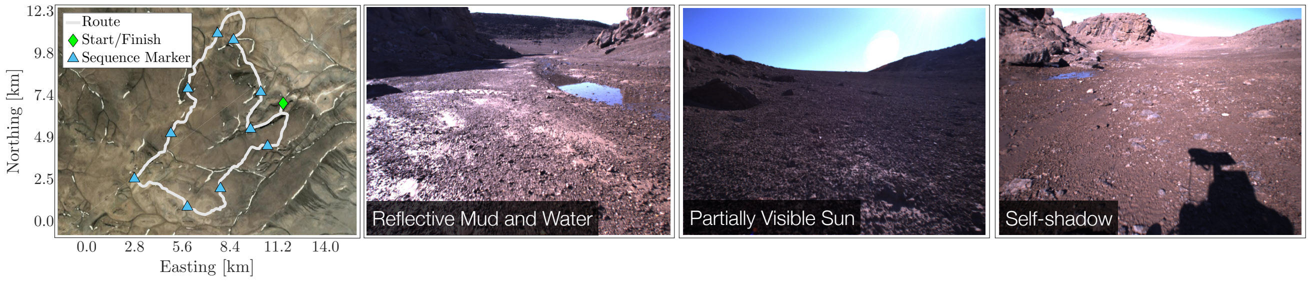


Figure 13. GPS track and sample images from the Devon Island traverse, with the start of each sequence highlighted. The Devon Island dataset is conducive to visual sun sensing due to the presence of strong environmental shadows, reflective surfaces such as mud and water, occasionally visible sun, and self-shadowing by the sensor platform. (Map data: Google, DigitalGlobe)

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. 2016), and the Sun-CNN of Ma et al. (2017).

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 error.

Figure 10 shows the results we obtained using simulated sun measurements on sequence `05`, in which the basic VO suffers from substantial orientation drift.[‡] Incorporating absolute orientation 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 3, 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 3 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 11 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 12 shows the results we obtained for sequences `02`, `05`, and `08` using the Sun-CNN of Ma et al. (2017), 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. 2016) 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. (2017) and Clement et al. (2016).

Table 3 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 – an improvement that comes without the need for additional sensors or a specially oriented camera.

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

[‡] In order to make a fair comparison to the Sun-CNN of Ma et al. (2017), 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.

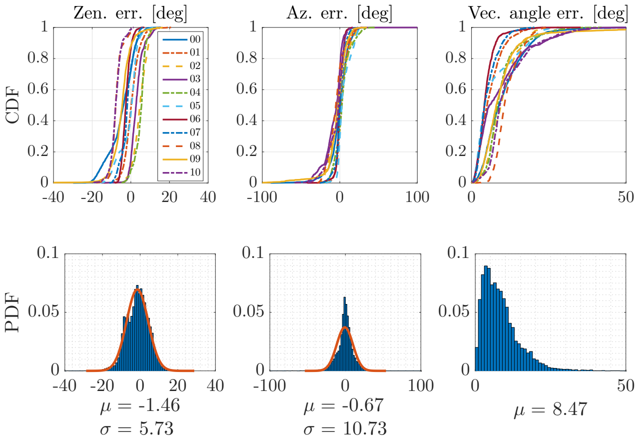


Figure 14. (Devon Island) Distributions of azimuth error, zenith error, and angular distance for Sun-BCNN compared to ground truth over each test sequence. *Top row*: Cumulative distributions of errors for each test sequence individually. *Bottom row:* Histograms and Gaussian fits of aggregated errors.

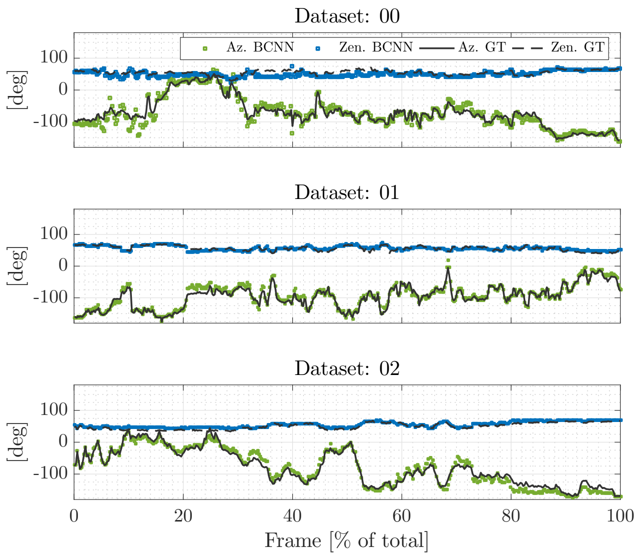


Figure 15. Azimuth (Sun-BCNN azimuth and zenith predictions over time for Devon Island test sequences `00`, `01` and `12`. Sun-BCNN is trained and tested on all frames (in our VO experiments, we use the Sun-BCNN predictions of every tenth image to make a fair comparison).

traversing a 10 km loop on Devon Island in the Canadian High Arctic (Figure 13). The rugged landscape of Devon Island (Figure 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 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 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.

Sun-BCNN Test Results

As in our experiments with the KITTI odometry benchmark, we obtained the mean estimated sun vector by evaluating Equation (4) 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 (13), and then applied Equation (5). As shown in Table 4, 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 14 and 16 plot the error distributions for azimuth, zenith, and angular distance for all 11 Devon Island odometry sequences, while Figure 15 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 7), 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 4 summarizes the test errors and ANEES of each sequence numerically, while Figures 14 and 16 plot the error distributions for azimuth, zenith, and angular distance for each sequence. Figure 15 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.

Table 4. Test Errors for Sun-BCNN on Devon Island odometry sequences with estimates computed at every image.

	Zenith Error [deg]			Azimuth Error [deg]			Vector Angle Error [deg]			
Sequence	Mean	Median	Stdev	Mean	Median	Stdev	Mean	Median	Stdev	**ANEES**[2]
`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 (ANEES) values with all sun directions that fall below a cosine distance threshold of 0.3 (relative to ground truth) and set $\tau^{-1} = 0.01$.

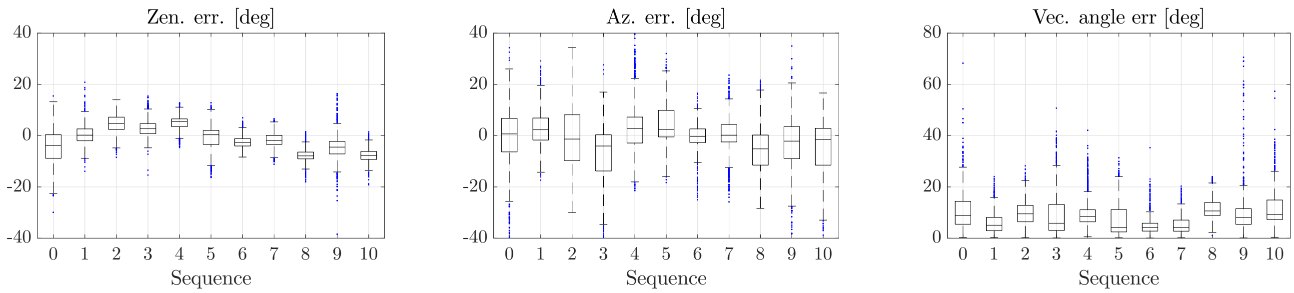


Figure 16. Box-and-whiskers plot of final test errors on Devon Island odometry sequences (c.f. Table 4).

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. (2017) 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. 2016) methods.

Figure 17 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.

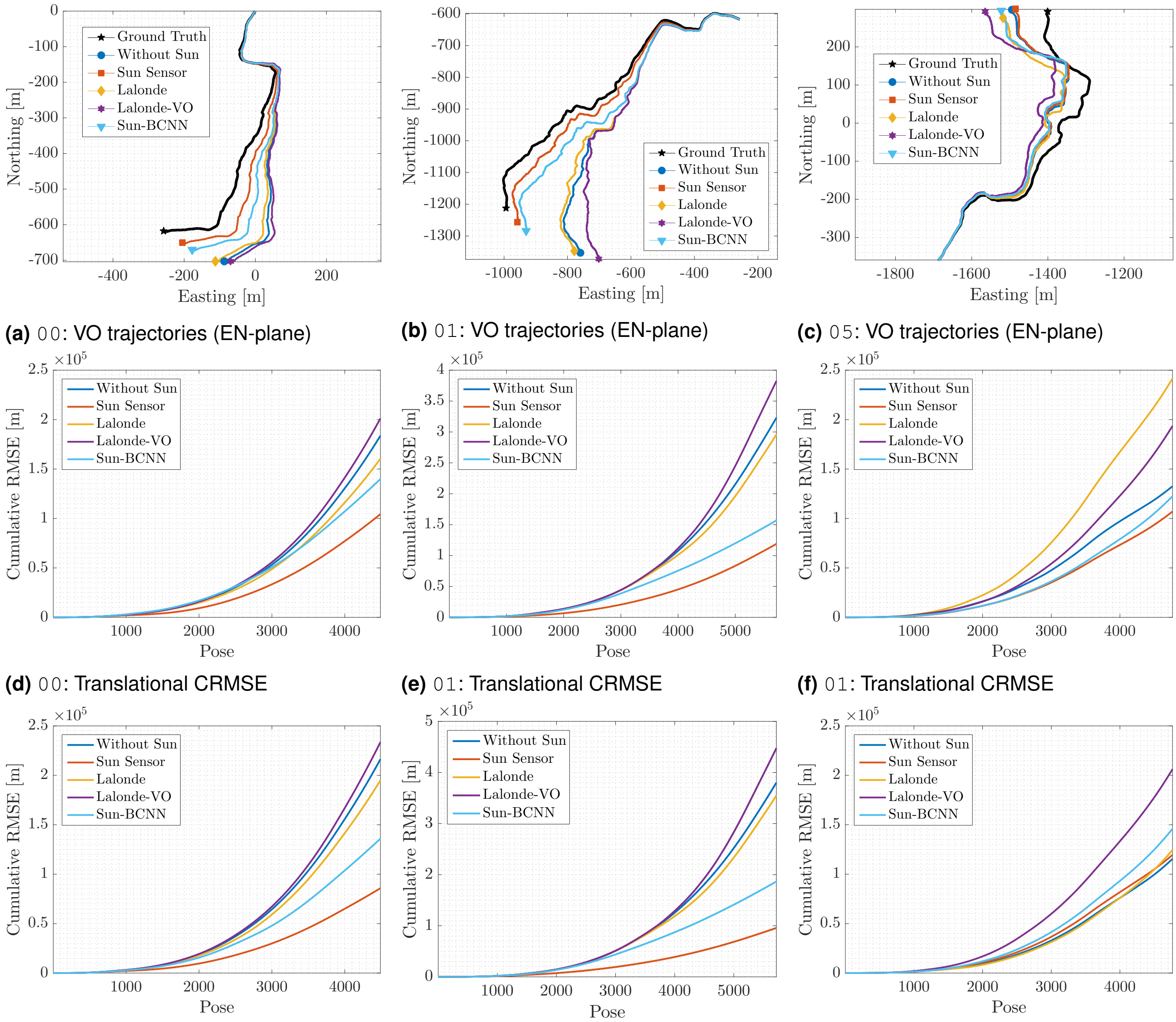
Table 5 summarizes these results numerically for all 11 sequences in the dataset. While the addition of sun sensing using either the hardware sensor or Sun-BCNN generally results in significant reductions in error, we note that in certain cases (e.g., sequence `05`), sun sensing has little or no impact on the VO result. We suspect that the translation errors in these cases are dominated by non-rotational effects, 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.

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.

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 3, 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.



(a) 00: VO trajectories (EN-plane) **(b)** 01: VO trajectories (EN-plane) **(c)** 05: VO trajectories (EN-plane)

(d) 00: Translational CRMSE **(e)** 01: Translational CRMSE **(f)** 01: Translational CRMSE

(g) 00: Translational CRMSE (EN-Plane) **(h)** 01: Translational CRMSE (EN-Plane) **(i)** 05: Translational CRMSE (EN-Plane)

Figure 17. 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).

Table 5. 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**

Figure 18. 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.

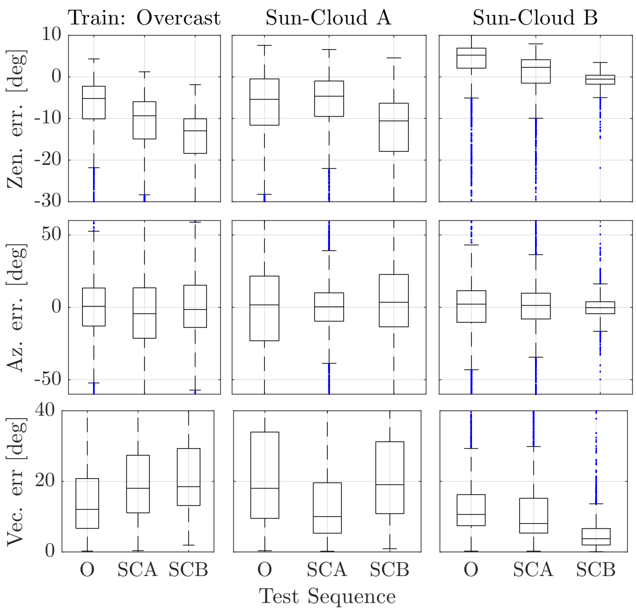


Figure 19. Box-and-whiskers plot for zenith, azimuth and vector angle errors for nine different combinations of train-test sequences taken from the Oxford Robotcar dataset. Each column corresponds to a different training sequence, and each plot contains three different test sequences. In the bottom legend, we use the labels O: *Overcast*, SCA: *Sun-Cloud A*, SCB: *Sun-Cloud B*.

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 18 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 19 shows the results of these experiments with box and whisker plots for azimuth, zenith and vector angle errors while Table 6 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 produces vector angle errors on the *Overcast* test set that are lower than even 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.

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 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.

Results Figure 20 shows the results of these experiments with box and whisker plots for azimuth, zenith and vector angle errors while Table 7 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

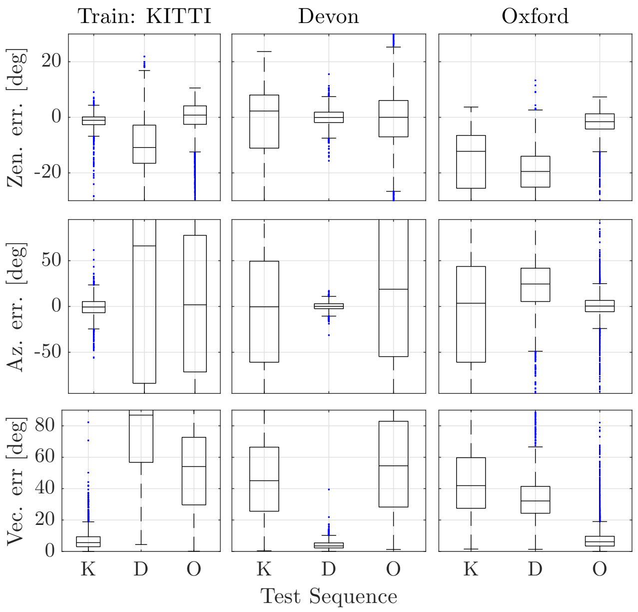


Figure 20. Box-and-whiskers plot for zenith, azimuth and vector angle errors for nine different combinations of train-test datasets. Each column corresponds to a different training sequence, and each plot contains three different test sequences. In the bottom legend, we use the labels K: KITTI, D: Devon 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.

camera. Furthermore, differences in dynamic range may also significantly affect the ability of Sun-BCNN to treat shading variations consistently.

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 (4) and (5) 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 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 (4) 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 (13).
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 (13). We then evaluate Equation (4) 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 8 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) 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 (13).
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) 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 9 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.

Conclusion

In this work, we have presented Sun-BCNN, a Bayesian CNN applied to the problem of sun direction estimation from a single RGB image in which the sun may not be visible. By leveraging the principled uncertainty estimates of the BCNN, we incorporated the sun direction estimates into a stereo visual odometry pipeline and demonstrated significant reductions in error growth over 21.6 km of urban driving data from the KITTI odometry benchmark and a further 10 km of visual navigation data from the Devon Island Rover Navigation Dataset, achieving median test error rates of approximately 12° and 7° respectively. Further, we demonstrated Sun-BCNN's ability to deal with cloud cover on the Oxford Robotcar Dataset, analyzed the effect of transferring a model across different terrain and cameras, and compared two different methods of computing the mean

Table 6. Test Errors for Sun-BCNN on three different Oxford Robotcar sequences collected on the same day with different lighting conditions.

		Zenith Error [deg]			Azimuth Error [deg]			Vector Error [deg]		
Train	**Test**	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 7. Test Errors for Sun-BCNN on different training and test datasets.

		Zenith Error [deg]			Azimuth Error [deg]			Vector Error [deg]		
Train	**Test**	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 8. A comparison of prediction errors from different mean estimation methods.

		Zenith Error [deg]			Azimuth Error [deg]			Vector Error [deg]		
Sequence	**Mean Type**	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

and covariance of a norm-constrained vector. We stress that although we integrated Sun-BCNN into a visual odometry pipeline in this work, it can just as readily be used to inject global orientation information into any egomotion estimator.

Possible avenues for future work include incorporating an explicit motion model into the visual odometry pipeline in order to better connect rotational and translational motion, amending the BCNN to produce temporally consistent estimates by leveraging the sequential nature of the test data (e.g., using a Recurrent Neural Network), and exploring ways in which the BCNN can account for different camera models to improve the generalizability of the trained models.

Acknowledgements

The authors would like to thank Wei-Chiu Ma, Shenlong Wang, and Raquel Urtasun for sharing their Sun-CNN data with us, NVIDIA for providing the Titan X GPU we use to train and test our model, and the Natural Science and Engineering Research Council of Canada (NSERC) for financially supporting Valentin Peretroukhin and Lee Clement's doctoral work.

References

Agarwal S, Mierle K et al. (2016) Ceres solver. URL http://ceres-solver.org.

Alcantarilla PF and Woodford OJ (2016) Noise models in feature-based stereo visual odometry .

Cheng Y, Maimone MW 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.

Table 9. 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

Clement L, Peretroukhin V and Kelly J (2016) Improving the accuracy of stereo visual odometry using visual illumination estimation. In: *Proc. Int. Symp. Experimental Robot. (ISER)*.

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

Deng J, Dong W, Socher R, Li LJ, Li K and Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: *Proc. IEEE Conf. Comput. Vision and Pattern Recognition, (CVPR)*. pp. 248–255.

Eisenman AR, Liebe CC and Perez R (2002) Sun sensing on the mars exploration rovers. In: *Aerosp. Conf. Proc.*, volume 5. IEEE, pp. 5–2249–5–2262 vol.5.

Engel J, Stuckler J and Cremers D (2015) Large-scale direct SLAM with stereo cameras. In: *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Syst. (IROS)*. pp. 1935–1942.

Fischler MA and Bolles RC (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automatated cartography. *Commun. ACM* 24(6): 381–395.

Fisher R (1953) Dispersion on a sphere. In: *Proc. Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 217. The Royal Society, pp. 295–305.

Furgale P and Barfoot TD (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 TD (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.

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)*. pp. 1050–1059.

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

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

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

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)*. pp. 675–678.

Kelly J, Saripalli S and Sukhatme GS (2008) Combined visual and inertial navigation for an unmanned aerial vehicle. In: *Proc. Field and Service Robot. (FSR)*. pp. 255–264.

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

Kendall A, Grimes M and Cipolla R (2015) Posenet: A convolutional network for real-time 6-dof camera relocalization. In: *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*.

Lalonde JF, Efros AA and Narasimhan SG (2011) Estimating the natural illumination conditions from a single outdoor image. *Int. J. Comput. Vis.* 98(2): 123–145.

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

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

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

Ma WC, Wang S, Brubaker MA, Fidler S and Urtasun R (2017) Find your way by observing the sun and other semantic cues. In: *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*.

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.

Olson CF, Matthies LH, Schoppers M and Maimone MW (2003) Rover navigation using stereo ego-motion. *Robot. Auton. Syst.* 43(4): 215–229.

Peretroukhin V, Clement L and Kelly J (2017) Reducing drift in visual odometry by inferring sun direction using a bayesian convolutional neural network. In: *Proc. IEEE Int. Conf. Robot. Automat. (ICRA), to appear*.

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)*. pp. 817–824.

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

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)*. pp. 1–9.

Zhang J and Singh S (2015) Visual-lidar odometry and mapping: low-drift, robust, and fast. In: *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*. pp. 2174–2181.

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)*. pp. 487–495.