Reducing Drift in Visual Odometry by Inferring Sun Direction using a Bayesian Convolutional Neural Network

Valentin Peretroukhin†, Lee Clement†, and Jonathan Kelly

***Abstract*— We present a method to incorporate global orientation information from the sun into a visual odometry pipeline using the existing image stream only. We leverage recent advances in Bayesian Convolutional Neural Networks to train and implement a sun detection model that infers a three-dimensional sun direction vector from a single RGB image (where the sun is typically not visible). 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. Our Bayesian sun detection model achieves median errors of less than 10 degrees on the KITTI odometry benchmark training set, and yields improvements of up to 37% in translational ARMSE and 32% in rotational ARMSE compared to standard VO. An implementation of our Bayesian CNN sun estimator (Sun-BCNN) is available as open-source code at** `https://github.com/utiasSTARS/sun-bcnn-vo`.

I. Introduction

Egomotion estimation is a fundamental building block of mobile autonomy. Although there exists an array of possible algorithm-sensor combinations that can estimate motion within unknown environments (including, for example, LIDAR-based point-cloud matching [1] and visual-inertial navigation [2]), egomotion estimation remains a dead-reckoning technique that will accumulate error over time without the injection of global information.

In this work, we focus on one technique to infer global orientation information without a known map: computing the direction of the sun. By leveraging recent advances in Bayesian Convolutional Neural Networks (BCNNs), we demonstrate how we can train a deep model to compute a direction vector from a single RGB image using only 20,000 training images. Furthermore, we show that our network can produce a principled covariance estimate that can readily be used in an egomotion estimation pipeline. We demonstrate one such use by incorporating sun direction estimates into a stereo Visual Odometry (VO) pipeline and report significant error reductions of up to 37% in translational average root mean squared error (ARMSE) and 32% in rotational ARMSE compared to plain VO on the KITTI odometry benchmark training set [3].

†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 {`lee.clement, v.peretroukhin`}`@mail.utoronto.ca`, `jkelly@utias.utoronto.ca`.

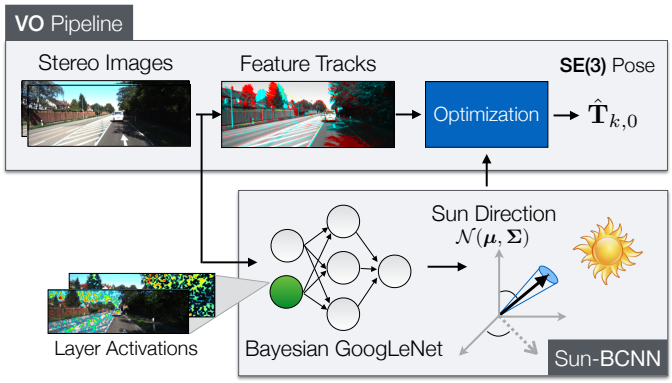


Fig. 1: Sun-BCNN (Sun Bayesian Convolutional Neural Network) incorporated into a visual odometry (VO) pipeline. Sun direction estimates are inferred as a mean and covariance by a Bayesian-CNN, and are incorporated into a sliding window bundle adjuster to produce a final trajectory estimate.

Our main contributions are 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 full dropout (dropout layers after all convolutional layers) 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 21.6 km of urban driving data from the KITTI odometry benchmark training set [3]; and
5) We release our Bayesian CNN sun estimator (Sun-BCNN) as open-source code.

II. Related Work

Visual odometry, a technique to estimate the egomotion of a moving platform, has a rich history of research including a notable implementation onboard the Mars Exploration Rovers (MERs) [4]. Modern approaches to VO can achieve estimation errors below 1% of total distance traveled [3]. To achieve such accurate and robust estimates, modern techniques use careful visual feature pruning [5], [6], adaptive robust methods [7], [8], and operate directly on pixel intensities [9]

Independent of the estimator, VO exhibits super-linear error growth [10], and is particularly sensitive to errors in orientation [5], [10]. One way to reduce orientation error is to incorporate observations of an object with a known

relative orientation (e.g., the sun). Physical sun sensors were incorporated into the MERs [11], [12], and have been shown to improve the accuracy of VO in planetary analogue environments [13], [14]. More recently, software-based alternatives have been developed that can estimate the direction of the sun from illumination cues within a single image [15]–[17], making sun-aided navigation possible without additional sensors or a specially oriented camera [17].

Convolutional Neural Networks have recently been adopted across the computer vision community for a wide range of classification, segmentation, and learning tasks [18]. Recent work has shown that CNNs can learn orientation information directly from images [16], [19] by modifying the loss functions of existing discrete classification-based CNN architectures into continuous regression losses. Despite their success in improving prediction accuracy, existing CNN-based models do not report principled uncertainty estimates, which are important in the context of data fusion. However, Gal et al. [20] recently demonstrated that it is possible to achieve principled covariance outputs with only minor modifications to existing CNN architectures.

Our method is similar in spirit to the work of Ma et al. [16] who built a CNN-based sun sensor as part of a relocalization pipeline, and builds upon the work of Clement et al. [17] who demonstrated that virtual sun sensors can improve VO accuracy. Our model makes three important improvements: 1) We output not only a point estimate of the sun direction, but also use a principled covariance estimate that is incorporated into our estimator; 2) We use a full 3D observation 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 in [16]); and 3) We incorporate the sun direction covariance into a VO estimator that accounts for growth in pose uncertainty over time (unlike [17]). Furthermore, our Bayesian CNN includes a dropout layer after every convolutional and fully connected layer (as outlined in [20] but not done in [19] due to reduced test accuracy), which produces more principled covariance outputs without significantly affecting test accuracy.

III. Indirect Sun Detection using a Bayesian Convolutional Neural Network

To infer the direction of the Sun, we use a Convolutional Neural Network. We motivate the choice of a deep model through the empirical findings of Clement et al. [17] and Ma et al. [16], who demonstrated that hand-crafted models such as [15] do not generalize well to other environments.

We choose a deep neural network structure based on GoogLeNet [21] due to its use in past work that adapted it for orientation regression [19]. Unlike Ma et al. [16], we choose to transfer weights trained on the MIT Places dataset [22] rather than ImageNet [23]. We believe the MIT Places dataset is a more appropriate starting point for localization tasks than ImageNet since it includes outdoor scenes and is more concerned with classifying physical locations rather than objects.

1) Cost Function: We train the network by minimizing the cosine distance between the training target sun unit norm direction vector $\hat{\mathbf{s}}_{\text{target}}$ and the estimate $\hat{\mathbf{s}}$:

$$\mathcal{L}(\hat{\mathbf{s}}) = 1 - (\hat{\mathbf{s}} \cdot \hat{\mathbf{s}}_{\text{target}}), \tag{1}$$

Note that in our implementation, we do not formulate the cosine distance loss explicitly, but instead minimize half the square of the Euclidian distance between the normalized estimate, $\hat{\mathbf{s}}$, and the target direction. Since both $\hat{\mathbf{s}}$ and $\hat{\mathbf{s}}_{\text{target}}$ have unit length, this is equivalent to minimizing Equation (1):

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

2) Uncertainty Estimation: Crucially, our sun detection model produces not only point estimates of the sun direction, but also an associated covariance. For this approach, we adopt a recent approach called Bayesian Convolution Neural Networks (BCNN) [20] that exploits the relationship between sampling a network with dropout layers, and variational inference of a Bayesian neural network where the weights are the conditioning variables. We outline the method here briefly, and refer the reader to [20] for more details.

The method begins with a prior on the weights within a deep neural network, $p(\boldsymbol{w}) \in \mathcal{N}(\mathbf{0}, l^{-2}\mathbf{1})$ with l set to a characteristic length scale, and attempts to compute a posterior distribution given training inputs and targets, $p(\boldsymbol{w}|\mathbf{X}, \hat{\mathbf{S}}_{\text{target}})$. In general, this posterior is intractable, so instead BCNN relies on variational inference to approximate the posterior using a distribution $q(\boldsymbol{w})$:

$$q(\boldsymbol{w}_i) = \mathbf{M}_i \text{ diag}\left\{\{b_j^i\}_{j=1}^{K_i}\right\} \tag{2}$$

$$b_j^i \in \text{Bernoulli}(p_i) \tag{3}$$

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

BCNN then computes the posterior by minimizing the Kullback Leibler (KL) divergence between the variational distribution and the true posterior: $\text{KL}(p(\boldsymbol{w}|\mathbf{X}, \hat{\mathbf{S}}_{\text{target}})||q(\boldsymbol{w}))$. This final posterior over the weights still remains intractable, and is instead computed using Monte Carlo integration over the weights, $\boldsymbol{w}$. The first moment of the final predictive distribution is then estimated through moment matching as:

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

where $\boldsymbol{w}^n$ is a sample from $q(\boldsymbol{w})$. The predictive variance is computed as:

$$\begin{aligned}\text{Covar}(\hat{\mathbf{s}}^*) \approx{}& \tau^{-1}\mathbf{1} \\ &+ \frac{1}{N}\sum_{n=1}^{N} \hat{\mathbf{s}}^*(\mathbf{x}^*, \boldsymbol{w}^n)\hat{\mathbf{s}}^*(\mathbf{x}^*, \boldsymbol{w}^n)^T - \bar{\hat{\mathbf{s}}}^*\bar{\hat{\mathbf{s}}}^{*T}\end{aligned} \tag{5}$$

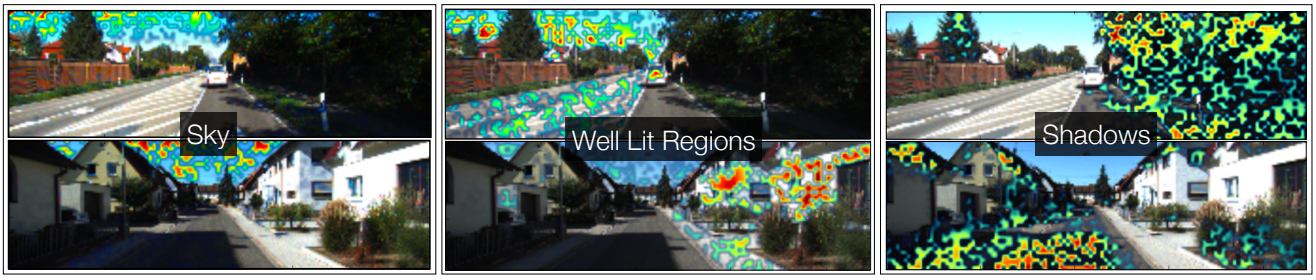


Fig. 2: Three `conv1` layer activation maps superimposed on two images from dataset `00` and `04` for three selected filters. Each of the filters picks out different salient parts of the image that aids in sun direction inference.

where $\mathbf{1}$ is the identity matrix,

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

with p our dropout probability, M the number of samples in our training data, and λ our weight decay.

Following [20], we build our BCNN by incorporating 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 and obtain the relevant statistical quantities using Equations (4) and (5).

IV. Sliding Window Stereo Visual Odometry

We adopt a sliding window sparse stereo VO technique that has been used in a number of successful mobile robotics applications [24]–[27]. Our task is to estimate a window of $SE(3)$ 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 observed to provide good VO accuracy at low computational cost. As discussed in Section IV-C, 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.

A. Observation Model

We assume that our 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.

B. Sliding Window Bundle Adjustment

We use the open-source `libviso2` package [26] to detect and track keypoints between stereo image pairs. Based on these keypoint tracks, a three-point Random Sample Consensus (RANSAC) algorithm [28] 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 [29]. 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

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

The quantity $\mathbf{e}_{\mathbf{y}_{k,j}} = \tilde{\mathbf{y}}_{k,j} - \mathbf{y}_{k,j}$ 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 $\tilde{\mathbf{y}}_{k,j} = \mathbf{g}\left(\tilde{\mathbf{T}}_{k,0}\tilde{\mathbf{p}}_0^j\right)$, where $\tilde{\mathbf{T}}_{k,0}$ and $\tilde{\mathbf{p}}_0^j$ 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 $\tilde{\mathbf{T}}_{k_1,0}$ of this pose compared to the prior can be computed using the $SE(3)$ matrix logarithm as $\mathbf{e}_{\check{\mathbf{T}}_{k_1,0}} = \log\left(\check{\mathbf{T}}_{k_1,0}^{-1}\tilde{\mathbf{T}}_{k_1,0}\right)$. The 6×6 matrix $\mathbf{R}_{\check{\mathbf{T}}_{k_1,0}}$ is the covariance associated with $\tilde{\mathbf{T}}_{k_1,0}$ 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.

C. Sun-based Orientation Correction

In order to combat drift in the VO estimate produced by accumulated orientation error, we adopt the technique of Lambert et al. [14] 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 [14].

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 parameterization, taking care to first rotate the coordinate system by a rotation $\mathbf{C}$ to avoid singularities:

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

where $\mathbf{s}'_k = \begin{bmatrix}s'_{k,x} & s'_{k,y} & s'_{k,z}\end{bmatrix}^T = \mathbf{C}\mathbf{s}_k$. 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 rotated 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.

V. Experiments

To train and test Sun-BCNN we used the KITTI odometry benchmark training sequences. 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. In this section, the test datasets simply correspond to each odometry sequence, while the corresponding training datasets are formed through the union of the remaining nine sequences.

A. Training Sun-BCNN

We implemented our network in Caffe [30] (for the normalization layers, we use the `L2Norm` layer from the Caffe-SL[1] fork of Caffe) and trained the network using stochastic gradient descent, performing 30,000 iterations with a batch size of 64. This results in approximately 1000 epochs of training on an average of roughly 20,000 images. Figure 5 plots the training and test loss as a function of iteration. We set all dropout probabilities to 0.5.

1) Data Preparation & Transfer Learning: We resized the KITTI images from their original, rectified size of $[1242 \times 378]$ to $[224 \times 68]$ and then padded the top and bottom 78 pixels with a constant intensity to achieve the $[224 \times 224]$ image size expected by GoogleLeNet. Note that unlike Ma et al. [31] we opted to preserve the aspect ratio of the image, in exchange for a lower vertical resolution. We performed no additional cropping or rotating of the images.

2) Covariance Estimation: To obtain useful covariance estimates for our VO estimator from the BCNN, we need to compute a covariance on the azimuth and zenith angles. To do this, we sampled the network to obtain unit-length direction vectors, converted them to azimuth and zenith angles by applying Equation (13), and finally applied Equation (5). We also experimented with retaining the samples in unit vector form, applying Equation (5), and then linearly propagating this covariance through Equation (5). We found that the former resulted in more consistent estimates, while the latter increased test accuracy. All results reported in this paper use the latter approach.

3) Model Precision: In our results, we found an empirically optimal model precision (τ in Equation (6)) by setting the weight decay $\lambda = 0.0005$ and using a combination of cross validation and Normalized Estimation Error Squared (NEES) analysis to arrive at a final length scale parameter l. We found l to be significantly larger in our experiments than that prescribed for classification tasks [20]. We leave a detailed analysis of these hyperparameters to future work.

B. Testing Sun-BCNN

Once trained, we analyzed the accuracy of Sun-BCNN by evaluating the mean prediction vectors (Equation (4), with N set to `100`). Figure 3 plots error distributions for the azimuth, zenith, and angular distance for all ten test datasets. Figure 4 plots the test errors using a Box-and-Whiskers plot, while Table I summarizes the results numerically. Sun-BCNN achieved median vector angle errors of less than 15 degrees on all KITTI datasets, with the exception of dataset

[1] `https://github.com/wanji/caffe-sl`

TABLE I: Test Errors for Sun-BCNN on KITTI Odometry Sequences.

Sequence	Zenith Error [deg]			Azimuth Error [deg]			Vector Angle Error [deg]		
	Mean	Median	Stdev	Mean	Median	Stdev	Mean	Median	Stdev
00	-5.08	-1.58	16.65	3.44	1.02	17.87	16.49	11.94	15.79
01	15.09	18.33	13.05	17.34	14.33	16.19	26.33	23.83	12.60
02	4.02	4.19	20.49	0.01	1.46	23.53	21.48	14.81	19.45
04	-0.10	-0.90	4.50	8.18	10.38	8.14	6.70	6.61	4.25
05	-3.75	-2.02	11.25	-1.94	-1.17	15.82	13.57	10.42	12.01
06	-14.22	-13.92	10.96	3.14	1.89	7.08	16.04	14.67	10.90
07	-4.87	-3.75	10.88	-0.14	0.51	7.86	10.36	8.00	8.74
08	-4.58	-1.77	21.50	-1.08	-1.11	22.10	20.19	12.28	21.26
09	-0.97	-0.94	11.59	-2.25	-0.22	16.98	13.45	10.12	12.45
10	0.87	2.05	14.03	1.84	-2.60	17.32	13.27	9.58	13.16

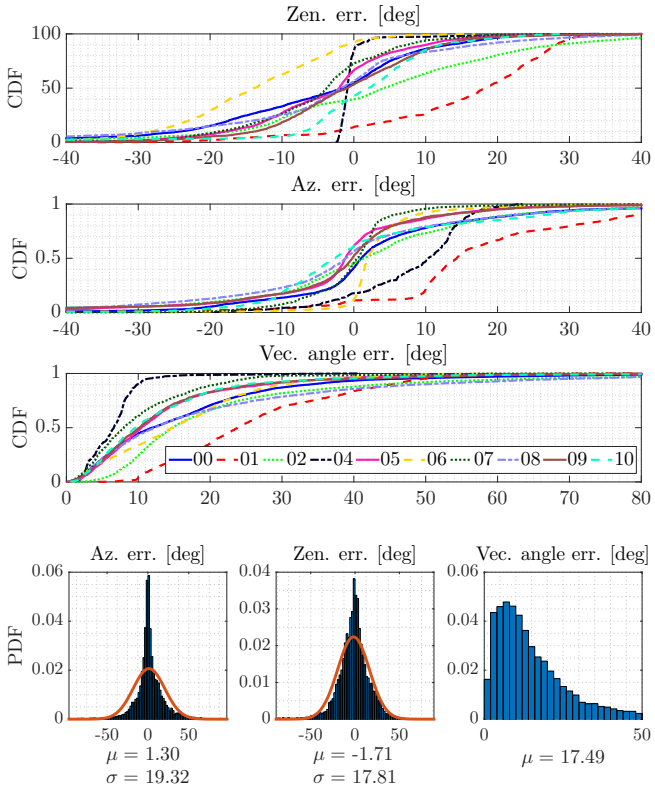


Fig. 3: Distributions over azimuth error, zenith error, and angular distance for the Sun-BCNN outputs compared to ground truth over all ten test sequences.

01, which was particularly difficult due to its lack of distinct shadows. As illustrated in Figure 2, the presence of strong shadows is one of the cues that Sun-BCNN relies on to estimate the sun direction.

C. Visual Odometry with 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 selected our noise levels such that the mean angular error of each simulated dataset was

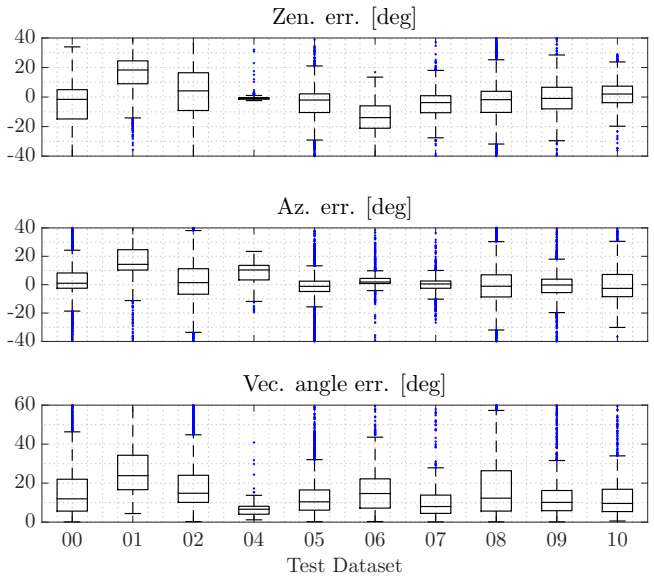


Fig. 4: Box-and-Whiskers plot for the final test errors on all ten KITTI odometry sequences. Table I summarizes these results numerically.

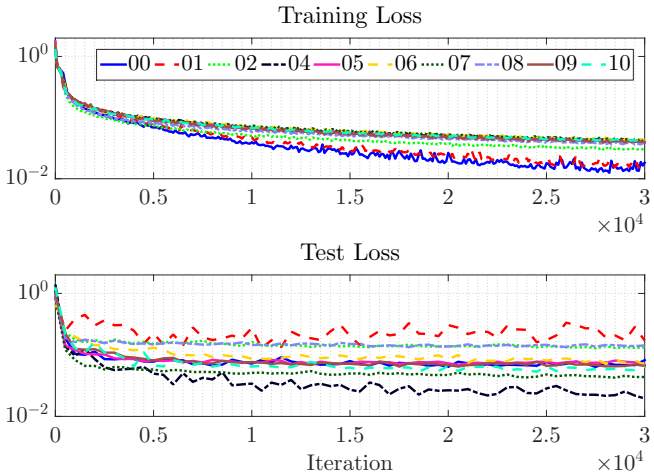


Fig. 5: Training and test loss for all ten training and test datasets. Note that the test error for dataset 01 remains particularly high due to the open, shadowless nature of the environment.

approximately 0, 10, 20, and 30 degrees, and denote each such dataset as "GT-Sun-0", "GT-Sun-10", "GT-Sun-20", and "GT-Sun-30", respectively.

Figure 6 shows the results we obtained using simulated sun measurements on the 2.2 km odometry 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. As shown in Table II, which summarizes our VO results for all ten sequences, these results are typical of sequences where orientation drift is the dominant source of error in the VO estimate.

While the VO solutions for sequences such as 00 and 10 do not improve in terms of translational ARMSE, Table II shows that rotational ARMSE improves on all ten sequences when low-noise sun measurements are included. This implies that the estimation errors of the basic VO solutions for these particular sequences are dominated by non-rotational effects.

D. Visual Odometry with Vision-based Sun Sensing

Figure 7 shows the results we obtained for sequence 05 using both the Sun-CNN of Ma et al. [31], which estimates only the azimuth angle of the sun, and 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. A selection of results using simulated sun measurements are also displayed for reference. Figure 7 shows that both Sun-CNN and Sun-BCNN succeed in reducing translational ARMSE on this sequence, with Sun-BCNN performing better overall and reducing the final translational drift significantly more than Sun-CNN. Table II bears out this conclusion for sequences 05 and 06, where Sun-BCNN reduces translational ARMSE substantially more than Sun-CNN. We stress that the interplay of rotational and translational error is complex, and although both Sun-CNN and Sun-BCNN appear to achieve lower translational RMSE than GT-Sun-0 on certain segments of sequence 05, Table II shows that, on average, neither Sun-CNN nor Sun-BCNN outperforms GT-Sun-0 in terms of translational or rotational error.

Table II shows results for all ten sequences using Sun-BCNN. 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 on sequences where sun sensing had an impact. As previously noted, sequences such as 00 do not benefit 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 the context of VO – an improvement that comes without the need for additional sensors or a specially oriented camera.

VI. Conclusion & Future Work

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. By using a full complement of dropout layers, we were able to successfully train the network on a relatively small set of training images, while achieving median test error rates of less than 10 degrees. We stress that while we integrated Sun-BCNN into a visual odometry pipeline, it can just as readily be used to inject global orientation information into any egomotion estimation pipeline.

Possible avenues for future work include investigating the effect of cloud cover on sun direction estimates, an analysis of the effect of hyperparameters such as length scale and weight decay on the final model, and the use of multiple cameras with non-overlapping fields of view to compute and combine sun direction estimates from multiple perspectives, which would be useful in scenarios such as [32] where multiple cameras are desirable for localization.

References

[1] J Zhang and S Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 2174–2181.

[2] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Rob. Res.*, vol. 34, no. 3, pp. 314–334, 2015, ISSN: 0278-3649.

[3] A Geiger, P Lenz, C Stiller, and R Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Rob. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[4] D Scaramuzza and F Fraundorfer, "Visual odometry [tutorial]," *IEEE Robot. Autom. Mag.*, vol. 18, no. 4, pp. 80–92, Dec. 2011, ISSN: 1070-9932.

[5] I Cvišić and I Petrović, "Stereo odometry based on careful feature selection and tracking," in *Mobile Robots (ECMR), 2015 European Conference on*, 2015, pp. 1–6.

[6] M Buczko and V Willert, "How to distinguish inliers from outliers in visual odometry for high-speed automotive applications," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2016, pp. 478–483.

[7] P. F. Alcantarilla and O. J. Woodford, "Noise models in feature-based stereo visual odometry," 2016. arXiv: 1607.00273 [cs.RO].

[8] V Peretroukhin, W Vega-Brown, N Roy, and J Kelly, "Probe-gk: Predictive robust estimation using generalized kernels," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 817–824.

[9] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," 2016. arXiv: 1607.02565 [cs.CV].

[10] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone, "Rover navigation using stereo ego-motion," *Rob. Auton. Syst.*, vol. 43, no. 4, pp. 215–229, 2003.

[11] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *J. Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007.

[12] A. R. Eisenman, C. C. Liebe, and R Perez, "Sun sensing on the mars exploration rovers," in *Aerosp. Conf. Proc.*, vol. 5, IEEE, 2002, 5–2249–5–2262 vol.5.

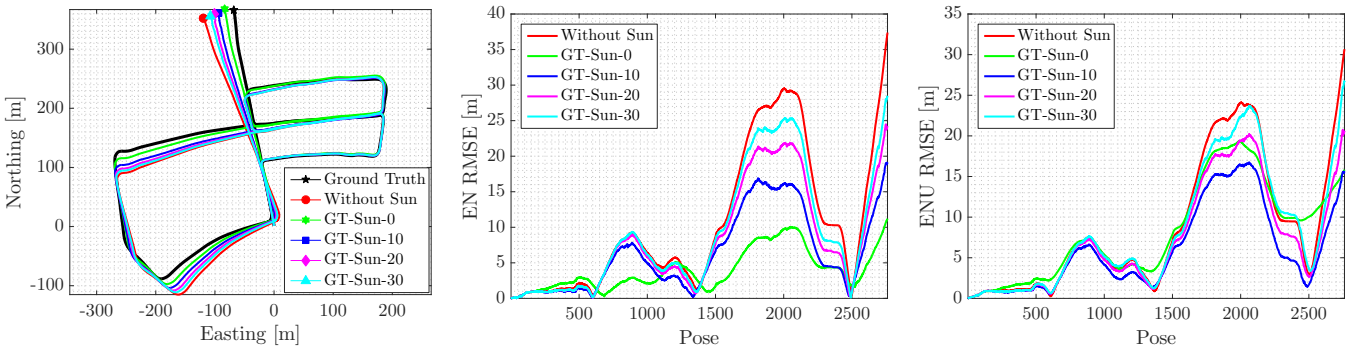
TABLE II: Comparison of translational and rotational average root mean squared errors (ARMSE) on KITTI odometry sequences without sun sensing, with simulated sun sensing, and with CNN-based sun sensing.

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	4.37	140.23	15.94	1.07	8.74	5.05	3.43	17.51	6.68	2.08
GT-Sun-0	7.05	116.36	13.00	1.20	5.76	4.57	1.47	12.49	6.18	2.59
GT-Sun-10	5.46	118.01	11.74	0.94	5.99	4.67	1.65	11.34	7.85	2.51
GT-Sun-20	4.46	117.83	13.58	1.01	7.24	4.73	2.29	12.63	8.54	2.68
GT-Sun-30	5.39	119.35	20.76	1.03	8.33	4.78	2.56	15.54	10.02	3.68
Sun-CNN[3]	4.81	-	-	-	7.71	5.00	-	-	-	-
Sun-BCNN	5.48	120.39	26.68	1.23	6.27	4.52	2.77	10.98	7.49	3.27
Trans. ARMSE (EN-plane) [m]										
Without sun	3.86	142.40	17.73	1.09	10.50	5.79	3.89	17.64	8.02	1.88
GT-Sun-0	3.30	124.76	11.76	1.09	4.75	4.76	1.52	11.54	6.66	2.62
GT-Sun-10	4.26	125.30	11.23	0.94	6.35	4.91	1.80	11.79	6.83	2.61
GT-Sun-20	4.45	126.04	11.43	1.05	8.10	5.16	2.69	13.29	7.38	2.32
GT-Sun-30	4.62	126.84	11.34	1.09	9.12	5.27	2.96	14.96	7.96	2.21
Sun-CNN[3]	4.43	-	-	-	5.95	4.69	-	-	-	-
Sun-BCNN	4.22	127.73	10.17	1.31	3.89	3.30	3.09	8.00	7.73	3.29
Rot. ARMSE $(\times 10^{-3})$ **[axis-angle]**										
Without sun	22.68	156.37	40.41	8.65	59.48	28.90	41.39	51.37	19.45	21.59
GT-Sun-0	8.28	97.00	34.86	7.00	38.90	23.85	20.54	36.74	15.91	9.64
GT-Sun-10	17.04	100.78	39.35	9.77	41.66	25.94	23.95	39.04	17.79	12.54
GT-Sun-20	19.75	103.98	44.94	9.09	49.01	26.72	30.90	44.61	19.60	14.70
GT-Sun-30	33.22	112.31	60.89	9.23	56.81	27.43	33.51	55.66	25.21	17.11
Sun-CNN[3]	27.59	-	-	-	46.83	25.44	-	-	-	-
Sun-BCNN	27.70	105.89	98.67	8.83	44.93	25.62	36.10	42.65	22.41	19.79

[1] Because we rely on the 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.

[3] We do not currently have access to the Sun-CNN model. The authors of [31] previously provided us with model outputs for sequences `00`, `05` and `06` only, but we plan to include a full comparison over all 10 sequences in the final paper for both Sun-CNN and the method of Lalonde et al. [15], [17]

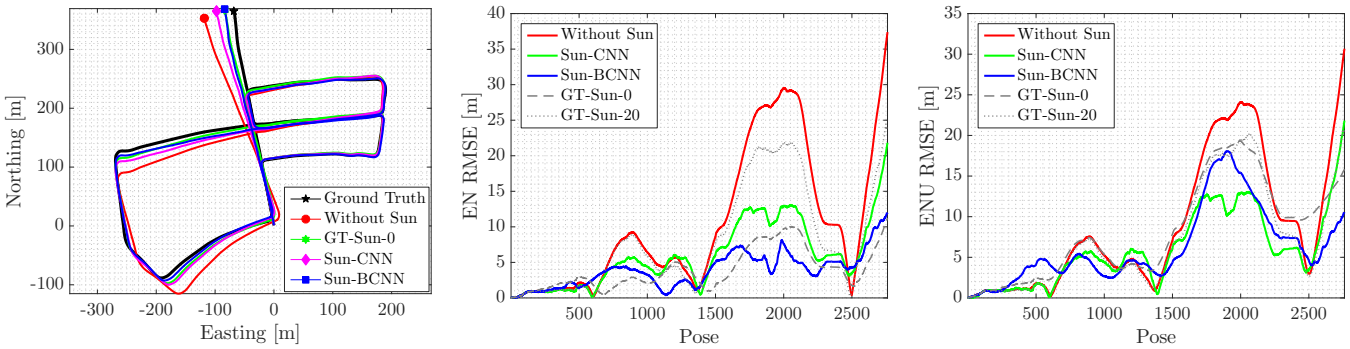


(a) Top-down view of estimated and ground truth trajectories in the EN-plane.

(b) Root Mean Squared Error (RMSE) of the VO estimates in the EN-plane.

(c) Root Mean Squared Error (RMSE) of the VO estimates.

Fig. 6: VO results for sequence `05`, in which the 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.



(a) Top-down view of estimated and ground truth trajectories in the EN-plane.

(b) Root Mean Squared Error (RMSE) of the VO estimates in the EN-plane.

(c) Root Mean Squared Error (RMSE) of the VO estimates.

Fig. 7: Sample VO results for sequence `05`, in which the VO suffers from substantial orientation drift. The sun direction estimates provided by both Sun-CNN [31] and Sun-BCNN significantly improve the VO solution. A selection of results using simulated sun measurements are shown for reference. The interplay of rotational and translational error is complex, and although both Sun-CNN and Sun-BCNN at times achieve lower translational RMSE than GT-Sun-0, on average, neither Sun-CNN nor Sun-BCNN outperforms GT-Sun-0 in terms of translational or rotational RMSE (Table II).

[13] P Furgale, J Enright, and T Barfoot, "Sun sensor navigation for planetary rovers: Theory and field testing," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 3, pp. 1631–1647, Jul. 2011.

[14] A. Lambert, P. Furgale, T. D. Barfoot, and J. Enright, "Field testing of visual odometry aided by a sun sensor and inclinometer," *J. Field Robotics*, vol. 29, no. 3, pp. 426–444, 2012.

[15] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan, "Estimating the natural illumination conditions from a single outdoor image," en, *Int. J. Comput. Vis.*, vol. 98, no. 2, pp. 123–145, 2011.

[16] W.-C. Ma, S. Wang, M. A. Brubaker, S. Fidler, and R. Urtasun, "Find your way by observing the sun and other semantic cues," 2016. arXiv: `1606.07415 [cs.CV]`.

[17] L. Clement, V. Peretroukhin, and J. Kelly, "Improving the accuracy of stereo visual odometry using visual illumination estimation," in *Proceedings of the International Symposium on Experimental Robotics, to appear*, Oct. 2016.

[18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," en, *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, ISSN: 0028-0836, 1476-4687.

[19] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2938–2946.

[20] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of The 33rd International Conference on Machine Learning*, 2016, pp. 1050–1059.

[21] C Szegedy, W. Liu, Y. Jia, P Sermanet, S Reed, D Anguelov, D Erhan, V Vanhoucke, and A Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9.

[22] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in neural information processing systems*, 2014, pp. 487–495.

[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, IEEE, 2009, pp. 248–255.

[24] Y. Cheng, M. W. Maimone, and L Matthies, "Visual odometry on the mars exploration rovers - a tool to ensure accurate driving and science imaging," *IEEE Robot. Autom. Mag.*, vol. 13, no. 2, pp. 54–62, Jun. 2006.

[25] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *J. Field Robotics*, vol. 27, no. 5, pp. 534–560, 2010.

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

[27] J. Kelly, S. Saripalli, and G. S. Sukhatme, "Combined visual and inertial navigation for an unmanned aerial vehicle," en, in *Field and Service Robotics*, ser. Springer Tracts in Advanced Robotics, C. Laugier and R. Siegwart, Eds., Springer Berlin Heidelberg, 2008, pp. 255–264.

[28] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[29] S. Agarwal, K. Mierle, *et al.*, *Ceres solver*. [Online]. Available: `http://ceres-solver.org`.

[30] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 675–678.

[31] W.-C. Ma, S. Wang, M. A. Brubaker, S. Fidler, and R. Urtasun, "Find your way by observing the sun and other semantic cues," 2016. arXiv: `1606.07415 [cs.CV]`.

[32] M. Paton, F. Pomerleau, and T. D. Barfoot, "Eyes in the back of your head: Robust visual teach & repeat using multiple stereo cameras," in *Computer and Robot Vision (CRV), 2015 12th Conference on*, 2015, pp. 46–53.