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 only the existing image stream, where the sun is typically not visible. 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. 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 a median error of less than 13 degrees on the KITTI odometry benchmark training set, and yields improvements of up to 38% in translational ARMSE and 30% in rotational ARMSE compared to standard VO. An open source implementation of our Bayesian CNN sun estimator (Sun-BCNN) using Caffe is available at** `https://github.com/utiasSTARS/sun-bcnn-vo`.

I. Introduction

Egomotion estimation is a fundamental building block of mobile autonomy. Although there exist an array of possible algorithm-sensor combinations that can estimate motion in unknown environments (e.g., LIDAR-based point-cloud matching [1] and visual-inertial navigation [2]), egomotion estimation remains a dead-reckoning technique that accumulates unbounded estimation error over time in the absence of global information such as GPS or a known map.

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 38% in translational average root mean squared error (ARMSE) and 30% 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`.

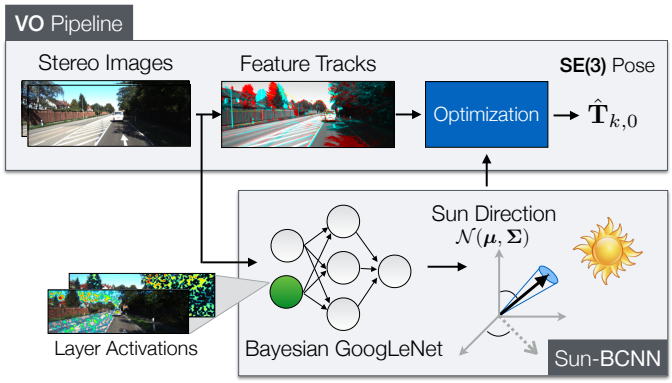


Fig. 1: Sun-BCNN (Sun Bayesian Convolutional Neural Network) incorporated into a visual odometry (VO) pipeline. A Bayesian CNN infers sun direction estimates as a mean and covariance, which are then incorporated into a sliding window bundle adjuster to produce a final trajectory estimate.

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 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 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 (VO), a technique to estimate the egomotion 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) [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], adaptive robust methods [6], [7], or operate directly on pixel intensities [8].

Independent of the estimator, VO exhibits super-linear error growth [9], and is particularly sensitive to errors in orientation [5], [9]. One way to reduce orientation error is to incorporate observations of a landmark whose position or

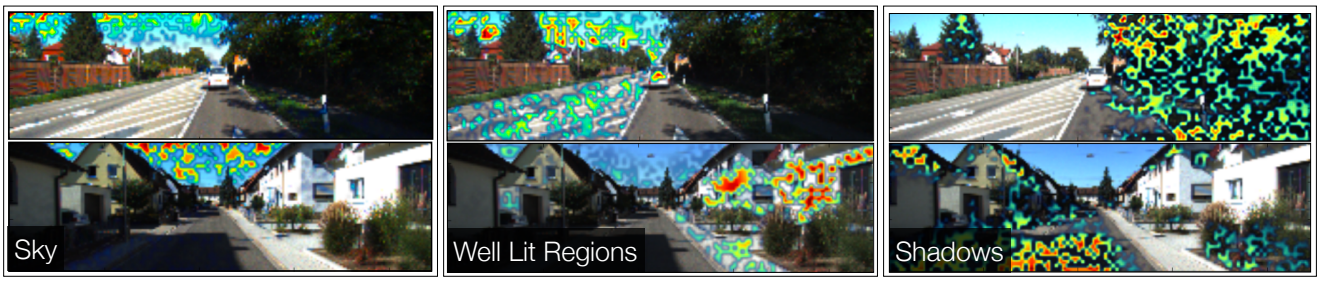


Fig. 2: Three `conv1` layer activation maps superimposed on two images from the KITTI [3] odometry benchmark `00` and `04` for three selected filters. Each filter picks out salient parts of the image that aid in sun direction inference.

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 [10], [11], and were also incorporated into the MERs [12], [13]. 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 [14]. Some of these methods have been based on hand-crafted illumination cues [14], [15], while others have attempted to learn such cues from data using deep Convolutional Neural Networks (CNNs) [16].

CNNs have been applied to a wide range of classification, segmentation, and learning tasks in computer vision [17]. 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 [16], [18], [19]. 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 [20] 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 et al. [19] who used it to improve their prior work on camera pose regression.

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. We also extend the work of Clement et al. [14] who demonstrated that virtual sun sensors can improve VO accuracy. 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 in [16]); and 3) we incorporate the sun direction covariance into a VO estimator that accounts for growth in pose uncertainty over time (unlike [14]). Furthermore, our Bayesian CNN includes a dropout layer after every convolutional and fully connected layer (as outlined in [20] but not done in [19]), which produces more principled covariance outputs.

III. Indirect Sun Detection using a Bayesian Convolutional Neural Network

We use a Convolutional Neural Network (CNN) to infer the direction of the sun. We motivate the choice of a deep model through the empirical findings of Clement et al. [14] and Ma et al. [16], who demonstrated that a CNN-based sun detector can substantially outperform hand-crafted models such as that of Lalonde et al. [15].

We choose a deep neural network structure based on GoogLeNet [21] due to its use in past work that adapted it for orientation regression [18]. 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 concerned with classifying physical locations rather than objects.

A. Cost Function

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

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

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 $\mathbf{s}_k$ and $\hat{\mathbf{s}}_k$. Since both vectors have unit length, this is equivalent to minimizing Equation (1):

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

B. Uncertainty Estimation

To output principled covariances for sun direction estimates, we adopt Bayesian Convolution Neural Networks (BCNNs) [20], [24], [25]. BCNNs rely on a connection between stochastic regularization (e.g. dropout, a widely adopted technique in deep learning) and approximate variational inference of a Bayesian Neural Network. We outline the technique here briefly, and refer the reader to [24] for more details.

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

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

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

Here, i indexes a particular layer in the neural network with K_i weights, $\mathbf{M}_i$ are the weights to be optimized, b_j^i 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 results in the same $\boldsymbol{w}$ as minimizing the Kullback-Leibler (KL) divergence between the variational distribution and the true posterior: $\text{KL}(p(\boldsymbol{w}|\mathbf{X},\mathbf{S})||q(\boldsymbol{w}))$. At test time, the first two moments of the predictive distribution are approximated using Monte Carlo integration over the weights $\boldsymbol{w}$:

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

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

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 [24], 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).

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 [26]–[29]. 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 provides 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 [28] to detect and track keypoints between stereo image pairs. Based on these keypoint tracks, a three-point Random Sample Consensus (RANSAC) algorithm generates an initial guess of the inter-frame 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 [30]. 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 $\mathbf{e}_{\mathbf{y}_{k,j}} = \hat{\mathbf{y}}_{k,j} - \mathbf{y}_{k,j}$ 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 $SE(3)$ matrix logarithm as $\mathbf{e}_{\check{\mathbf{T}}_{k_1,0}} = \log\left(\check{\mathbf{T}}_{k_1,0}^{-1}\hat{\mathbf{T}}_{k_1,0}\right)$. The 6×6 matrix $\mathbf{R}_{\check{\mathbf{T}}_{k_1,0}}$

is the covariance associated with $\check{\mathbf{T}}_{k_1,0}$ in its local tangent space, and is obtained as part of the previous window's bundle adjustment solution. This prior term allows consecutive windows of pose estimates to be combined in a principled way that appropriately propagates global pose uncertainty from window to window, which is essential in the context of optimal data fusion.

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. [11] 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, based on a solar ephemeris model that computes the sun direction for a given date, time, and location on Earth. We define the world frame $\underset{\rightarrow}{\mathcal{F}}_0$ 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 us to account for the apparent motion of the sun over long trajectories. Note that here we are using the notation $\mathbf{s}_k$ to represent the sun vector predicted by our sun sensing apparatus (denoted $\hat{\mathbf{s}}_k$ in Section III), not the ground truth training vector.

By transforming the global sun direction into each camera frame $\underset{\rightarrow}{\mathcal{F}}_k$ in the window, we obtain expected 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 expected sun direction $\hat{\mathbf{s}}_k$ to the estimated sun direction $\mathbf{s}_k$ 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 cost term constrains the orientation of the camera, which helps limit drift in the VO result due to orientation error [11].

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 $\underset{\rightarrow}{\mathcal{F}}_k$ to a zenith-azimuth parameterization:

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

where $\mathbf{s}_k = \begin{bmatrix}s_{k,x} & s_{k,y} & s_{k,z}\end{bmatrix}^T$. We can then define the term $\mathbf{e}_{\mathbf{s}_k} = \mathbf{f}(\mathbf{s}_k) - \mathbf{f}(\hat{\mathbf{s}}_k)$ 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.

V. Experiments

To train and test Sun-BCNN we used the KITTI odometry benchmark training sequences [3]. 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 consist of the union of the remaining nine sequences.

A. Training Sun-BCNN

We implemented our network in Caffe [31] (for the normalization layers, we used the `L2Norm` layer from the Caffe-SL fork[1]) 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. We set all dropout probabilities to 0.5.

1) Data Preparation & Transfer Learning: We resized the KITTI images from their original, rectified size of $[1242 \times 378]$ pixels to $[224 \times 224]$ pixels to achieve the image size expected by GoogleLeNet. We experimented with preserving the aspect ratio of the original image (padding zeros to the top and bottom of the resized image), but found that preserving the vertical resolution (as in [16]) resulted in better test-time accuracy. We performed no additional cropping or rotating of the images.

2) Model Precision: We found an empirically optimal model precision τ (see Equation (6)) by optimizing the Average Normalized Estimation Error Squared (ANEES) on each of our training datasets. Without this tuning, the BCNN uncertainty estimates tended to underestimate the true uncertainty. We believe this behaviour is due to two factors: 1) variational inference is known to underestimate predictive variance [25]; and 2) the observation noise is assumed to be homoscedastic. As noted by Gal [25], the BCNN can be made heteroscedastic by learning the model precision during training, but this extension is outside the scope of this work.

B. Testing Sun-BCNN

Once trained, we analyzed the accuracy and consistency of Sun-BCNN mean ($\mathbf{s}_k$) and covariance ($\mathbf{R}_{\mathbf{s}_k}$) estimates.

1) Computing $\mathbf{s}_k$: We evaluated Equation (4) (setting $N = 25$) and then renormalized the resulting mean vector to preserve unit length.

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

TABLE I: 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]			NEES[2]
Sequence	Mean	Median	Stdev	Mean	Median	Stdev	Mean	Median	Stdev	Mean
00	-1.11	0.16	5.03	2.37	4.11	30.84	16.02	12.04	15.87	1.44
01	-9.58	-7.32	8.73	13.89	23.57	35.60	27.78	27.00	13.72	1.03
02	-4.50	-3.28	6.29	4.01	5.43	39.50	19.22	12.19	20.66	1.44
04	-2.01	-1.98	0.64	2.09	1.78	3.44	3.44	2.77	1.86	0.27
05	-3.75	-1.88	6.34	2.06	-0.10	38.20	20.44	16.24	17.75	1.03
06	-2.46	-2.40	3.19	-9.99	-11.13	27.99	18.96	16.84	13.17	1.03
07	-0.24	0.26	3.41	1.76	5.36	25.72	15.50	13.27	11.59	1.44
08	-2.33	-0.09	7.01	3.86	1.59	40.90	20.44	13.09	20.21	0.85
09	-0.45	0.71	6.90	1.29	-2.71	27.02	15.05	10.02	13.89	1.27
10	1.73	2.19	4.40	5.21	4.39	18.95	11.27	8.67	10.43	1.66
	-2.61	-1.28	6.47	2.94	2.73	34.94	18.25	12.91	17.62	-

[1] We compute Average Normalized Estimation Error Squared (ANEES) values with all sun directions that fall below a cosine distance threshold of 0.1 (to emulate our visual odometry pipeline). For each test sequence, we select the model precision τ by optimizing ANEES on training data.

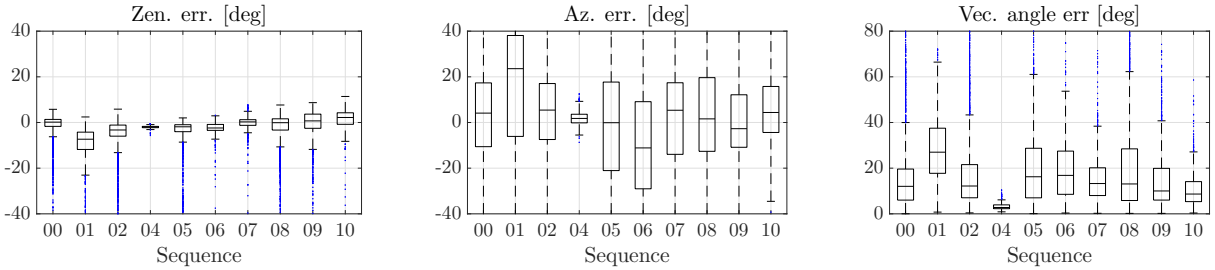


Fig. 3: Box-and-whiskers plot of final test errors on all ten KITTI odometry sequences (c.f. Table I).

2) Computing $\mathbf{R}_{\mathbf{s}_k}$*:* To obtain the required covariance on azimuth and zenith angles (recall that the BCNN outputs unit-length direction vectors), we sampled the vector outputs, converted them to azimuth and zenith angles using Equation (13), and then applied Equation (5). It is also possible to retain samples in unit vector form, apply Equation (5), and then propagate this covariance through a linearized Equation (13). In this paper we used the former approach, leaving a comparison of these two uncertainty propagation schemes to future work.

3) Results: Table I summarizes the test errors numerically, while Figures 3 and 4 plot the error distributions for azimuth, zenith, and angular distance for all ten KITTI odometry sequences. Table I also lists the Average Normalized Estimation Error Squared (ANEES) values for each sequence. Figure 5 shows three characteristic plots of the azimuth and zenith predictions over time. Sun-BCNN achieved median vector angle errors of less than 17 degrees on every sequence except sequence `01`, which was particularly difficult due to its lack of distinct shadows. As illustrated in Figure 2, Sun-BCNN relies on strong shadows to estimate the sun direction.

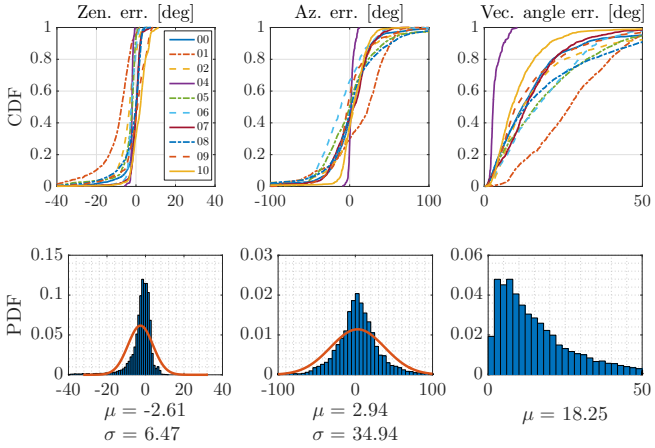


Fig. 4: Distributions of azimuth error, zenith error, and angular distance for Sun-BCNN compared to ground truth over each test sequence. *Top row*: Cumulative distributions of errors for each test sequence individually. *Bottom row:* Histograms and Gaussian fits of aggregated errors.

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 obtained these ground truth sun vectors by transforming the ephemeris vector into each camera frame using ground truth vehicle poses. We selected our noise levels such that the mean angular error of each simulated dataset was 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.

Figures 6a to 6c show the results we obtained using simulated sun measurements on the 2.2 km odometry sequence

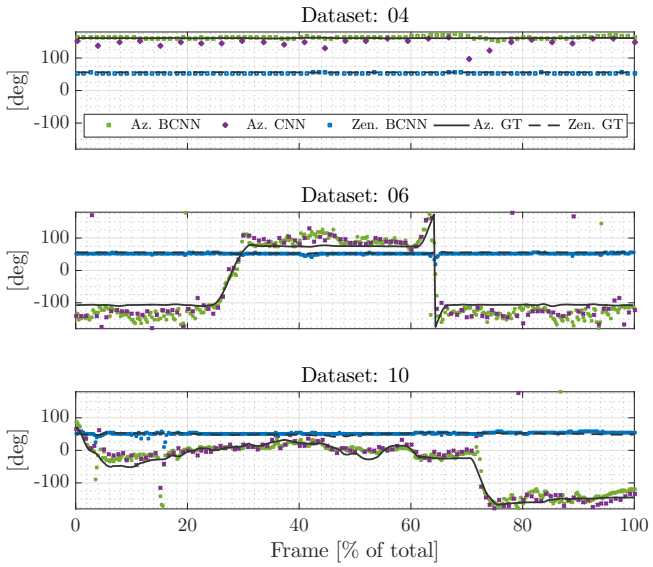


Fig. 5: Azimuth (Sun-BCNN and Sun-CNN [16]) and zenith (Sun-BCNN only) predictions over time for KITTI test sequences `00`, `04` and `10`. Sun-CNN is trained and tested on every tenth image, whereas Sun-BCNN is trained and tested on all frames (in our VO experiments, we use the Sun-BCNN predictions of every tenth image to make a fair comparison).

`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, 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 II 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. We speculate that incorporating a motion prior in our VO pipeline may mitigate these additional translational errors, and leave such an investigation to future work.

D. Visual Odometry with Vision-based Sun Sensing

Figures 6d to 6f show the results we obtained for sequence `05` using the Sun-CNN of Ma et al. [16], 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. in its original [15] and VO-informed [14] 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. [16] and Clement et al. [14].

Table II 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.

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 train the network using a relatively small training set while achieving median test error rates of less than 13 degrees. 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 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.

References

[1] J Zhang and S Singh, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *Proc. IEEE Int. Conf. Robot. Automat. (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. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.

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

[4] D Scaramuzza and F Fraundorfer, "Visual odometry [tutorial]," *IEEE Robot. Automat. Mag.*, vol. 18, no. 4, pp. 80–92, Dec. 2011.

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

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

TABLE II: 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	3.60	4.61	29.70	10.49	**5.48**
Lalonde-VO	4.87	199.03	29.41	2.48	9.74	**3.29**	4.52	27.82	11.06	5.59
Sun-CNN	4.36	**192.50**	26.58	2.48	8.92	3.51	**4.30**	26.99	10.15	5.58
Sun-BCNN	5.06	198.46	**23.17**	**2.47**	**8.48**	3.79	4.55	**25.80**	**9.12**	5.62
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	4.08	5.52	27.84	11.41	**5.87**
Lalonde-VO	5.38	231.33	33.68	1.82	11.13	**3.58**	5.42	32.24	12.41	6.00
Sun-CNN	4.56	**224.91**	24.65	1.82	9.99	3.94	**5.16**	30.09	11.21	5.99
Sun-BCNN	5.32	229.76	**20.01**	**1.78**	**7.13**	4.37	5.39	**26.46**	**8.57**	6.02
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	25.12	50.49	64.22	26.27	**20.49**
Lalonde-VO	27.91	185.52	69.52	12.98	68.09	22.45	49.74	65.35	28.82	22.10
Sun-CNN	24.05	**177.45**	**58.32**	13.00	61.48	23.11	**47.77**	60.55	26.19	21.99
Sun-BCNN	27.67	188.50	72.62	**12.89**	**48.90**	**22.10**	49.56	**58.74**	**21.03**	21.40

[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.

[7] V Peretroukhin, W Vega-Brown, N Roy, and J Kelly, "PROBE-GK: Predictive robust estimation using generalized kernels," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 817–824.

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

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

[10] 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.

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

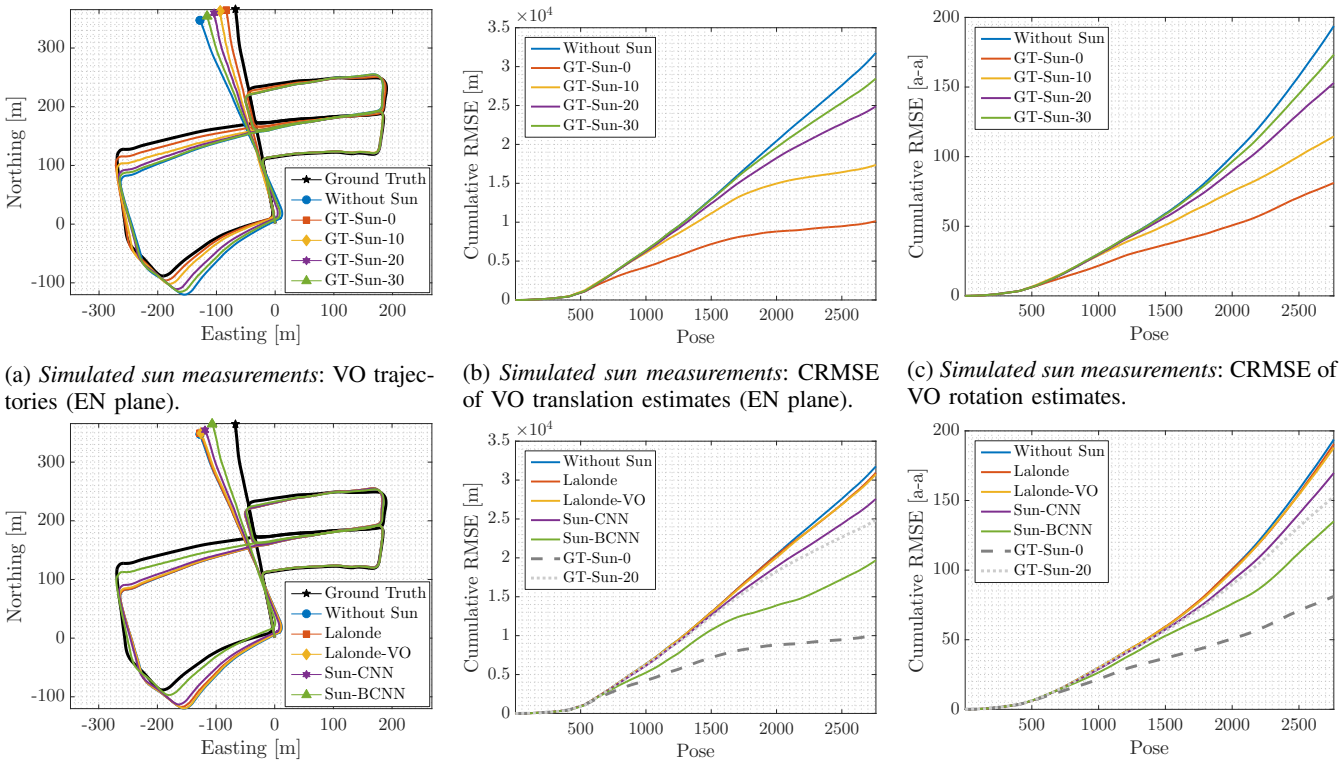
[12] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *J. Field Robot.*, vol. 24, no. 3, pp. 169–186, 2007.

[13] 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.

[14] L. Clement, V. Peretroukhin, and J. Kelly, "Improving the accuracy of stereo visual odometry using visual illumination estimation," in *Proc. Int. Symp. Experimental Robot. (ISER)*, Oct. 2016.

[15] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan, "Estimating the natural illumination conditions from a single outdoor image," *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," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2017.



(a) *Simulated sun measurements*: VO trajectories (EN plane).

(b) *Simulated sun measurements*: CRMSE of VO translation estimates (EN plane).

(c) *Simulated sun measurements*: CRMSE of VO rotation estimates.

(d) *Estimated sun measurements*: VO trajectories (EN plane).

(e) *Estimated sun measurements*: CRMSE of VO translation estimates (EN plane).

(f) *Estimated sun measurements*: CRMSE of VO rotation estimates.

Fig. 6: VO results for sequence `05`: top down trajectory plots in the Easting-Northing (EN) plane and Cumulative Root Mean Squared Error (CRMSE) plots for translational and rotational error. *Top row*: Results using a selection of simulated sun measurements of varying accuracy (c.f. Section V-C). *Bottom row*: Results using different sun estimation techniques, with selected simulated measurements added for reference. The sun direction estimates provided by Sun-BCNN significantly improve the VO solution, while the Lalonde [15], Lalonde-VO [14], and Sun-CNN [16] methods provide modest reductions in estimation error. In both simulated and estimated measurements, sun directions are computed at every tenth pose.

[17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

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

[19] A Kendall and R Cipolla, "Modelling uncertainty in deep learning for camera relocalization," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 4762–4769.

[20] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learning (ICML)*, 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 *Proc. IEEE Conf. Comput. 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 Inform. Process. Syst. (NIPS)*, 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 *Proc. IEEE Conf. Comput. Vision and Pattern Recognition, (CVPR)*, 2009, pp. 248–255.

[24] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with Bernoulli approximate variational inference," in *4th Int. Conf. on Learning Representations (ICLR) Workshop Track*, 2016.

[25] Y. Gal, "Uncertainty in deep learning," PhD thesis, University of Cambridge, 2016.

[26] 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. Automat. Mag.*, vol. 13, no. 2, pp. 54–62, Jun. 2006.

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

[28] 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.

[29] J. Kelly, S. Saripalli, and G. S. Sukhatme, "Combined visual and inertial navigation for an unmanned aerial vehicle," in *Proc. Field and Service Robot. (FSR)*, 2008, pp. 255–264.

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

[31] 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 *Proc. ACM Int. Conf. Multimedia (MM)*, 2014, pp. 675–678.