# Balanced Covariance Estimation for Visual Odometry Using Deep Networks

Youngji Kim[1], Sungho Yoon[2], Sujung Kim[3], and Ayoung Kim[1*]

***Abstract*— Uncertainty modeling is one of the recent trends in deep learning. Even though the uncertainty modeling is important in many applications, it has been overlooked until recently. In this paper, we propose a method of learning covariance for visual odometry. Unlike the existing supervised learning based uncertainty estimation, we introduce an unsupervised loss for uncertainty modeling. The learned uncertainty includes epistemic (model-driven) and aleatoric (data-driven) uncertainties.**

## I. Introduction

We usually model the state of a robot as a Gaussian distribution with a mean and variance. For a reliable state estimation, we need to consider both the mean and variance. However, the importance of uncertainty is sometimes overlooked and the performance of the estimator is measured only by the mean values. As shown in the examples of utilizing uncertainty for practical robotics applications, variance is as important as mean values. For instance in simultaneous localization and mapping (SLAM), the influence of each measurement is determined by the sensor measurement uncertainty. In active SLAM or belief space planning, the objective function relies heavily on the expected uncertainty. Moreover, uncertainty is required for the safe decision making as in the navigation of self-driving cars.

We propose a method of modeling uncertainty in sensor measurements and its application to SLAM. Among various sensor measurements, our focus is on the camera-based visual odometry (VO), which is particularly challenging to specify uncertainty. This is because camera is an extroverted sensor and uncertainty in VO relies both on the external environment where the image is taken and on the process of matching consecutive image frames. In this work, we propose a method of considering both the uncertainty from the environment (data uncertainty) and the uncertainty from the measurement process model (model uncertainty).

We follow the unified approach of estimating model and data uncertainty using deep networks proposed by Kendall and Gal [1]. Unlike the other supervised learning based approaches, we propose a fully unsupervised uncertainty learning scheme that does not require ground truth measurement error. To the best of our knowledge, it is the first report of unsupervised uncertainty learning for VO. In addition, to overcome the limitation of unsupervised learning of single sensor uncertainty, we provide a covariance balancing scheme that enables the network to learn relative magnitudes of uncertainties from different sensors.

Y. Kim and A. Kim are with the Department of Civil and Environmental Engineering, KAIST, Daejeon, S. Korea [youngjikim, ayoungk]@kaist.ac.kr

S. Yoon is with the Robotics Program, KAIST, Daejeon, S. Korea sungho.yoon@kaist.ac.kr

S. Kim is with Autonomous Driving Group, NAVER LABS. sujung.susanna.kim @naverlabs.com

This work is fully supported by [Deep Learning based Camera and LIDAR SLAM] project funded by Naver Labs Corporation.

## II. Unsupervised Learning of Uncertainty

### A. Supervised Uncertainty Learning

According to Kendall and Gal [1], epistemic (model) and aleatoric (data) uncertainty can be estimated using deep networks as

$$\begin{aligned}\hat{\Sigma}_{\mathbf{y}} &= \hat{\Sigma}_{\mathbf{y},\mathbf{epi}} + \hat{\Sigma}_{\mathbf{y},\mathbf{ale}} \\ &= \frac{1}{T}\sum_{t=1}^{T}\hat{\mathbf{y}}_t\hat{\mathbf{y}}_t^\top - \left(\frac{1}{T}\sum_{t=1}^{T}\hat{\mathbf{y}}_t\right)\left(\frac{1}{T}\sum_{t=1}^{T}\hat{\mathbf{y}}_t\right)^\top \\ &\quad + \frac{1}{T}\sum_{t=1}^{T}\hat{\Sigma}_{\mathbf{y}_t,\mathbf{ale}}. \end{aligned} \tag{1}$$

*1) Epistemic uncertainty:* One practical approach of learning epistemic uncertainty is by using dropout as an approximation of Bayesian Neural Networks (BNNs) [2]. Epistemic uncertainty is obtained by using dropouts also at test time. The empirical variance is computed from $T$ stochastic forward passes as

$$\hat{\Sigma}_{\mathbf{y},\mathbf{epi}} = \frac{1}{T}\sum_{t=1}^{T}\hat{\mathbf{y}}_t\hat{\mathbf{y}}_t^\top - \left(\frac{1}{T}\sum_{t=1}^{T}\hat{\mathbf{y}}_t\right)\left(\frac{1}{T}\sum_{t=1}^{T}\hat{\mathbf{y}}_t\right)^\top, \tag{2}$$

where $\hat{\mathbf{y}}$ denotes the network output.

*2) Aleatoric uncertainty:* Along with the predictive mean value, aleatoric uncertainty can be trained by making the output of the network as

$$[\hat{\mathbf{y}}, \hat{\Sigma}_{\mathbf{y},\mathbf{ale}}] = f(\mathbf{x}), \tag{3}$$

where $f$ indicates the network model and $\mathbf{x}$ is the input data.

Given a dataset $D = \{\mathbf{x}_i, \mathbf{y}_i \mid \forall i \in [1, \cdots, N]\}$, the loss for training aleatoric uncertainty is

$$L_{\text{sup}} = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{y}_i - \hat{\mathbf{y}}_i||^2_{\hat{\Sigma}_{\mathbf{y}_i,\mathbf{ale}}} + \log|\hat{\Sigma}_{\mathbf{y}_i,\mathbf{ale}}|, \tag{4}$$

where $||\cdot||^2_\Sigma$ denotes Mahalanobis distance, normalizing the error with variance as $||\mathbf{e}||^2_\Sigma = \mathbf{e}^\top\Sigma^{-1}\mathbf{e}$.

### B. Unsupervised Uncertainty Learning

We reformulate the described uncertainty learning process to make the two uncertainties trainable in an unsupervised manner. We propose the unsupervised uncertainty learning loss, which consists of two terms as in (1). Similar to the supervised uncertainty, epistemic uncertainty is obtained via dropout sampling as in the same manner in (2).

However, the loss for aleatoric uncertainty should be modified when training it in an unsupervised manner. In (4), the ground truth mean prediction $\mathbf{y}$ is required. To train the network without the ground truth, we modified the loss as

$$L_{\text{unsup}} = \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{z}_i - \hat{\mathbf{z}}_i||^2_{\hat{\Sigma}_{\mathbf{z}_i}} + \log|\hat{\Sigma}_{\mathbf{z}_i}| \tag{5}$$

by switching the ground truth $\mathbf{y}$ and its prediction $\hat{\mathbf{y}}$ into the measurement $\mathbf{z} = g(\mathbf{x})$ and its prediction $\hat{\mathbf{z}} = h(\mathbf{x}, \hat{\mathbf{y}})$. We introduce measruement function $g$ and $h$; $g$ converts input data $\mathbf{x}$ to the measurement $\mathbf{z}$, whereas $h$ converts input data $\mathbf{x}$ and the network prediction $\hat{\mathbf{y}}$ to the predicted measurement $\hat{\mathbf{z}}$.

The network can directly output $\hat{\Sigma}_{\mathbf{z}}$ when only the measurement uncertainty is concerned. However, when the uncertainty of network prediction $\hat{\Sigma}_{\mathbf{y}}$ should be known, we need to reformulate the measurement uncertainty as

$$\hat{\Sigma}_{\mathbf{z}} = \underbrace{\frac{\partial g}{\partial \mathbf{x}}\Sigma_{\mathbf{x}}\frac{\partial g}{\partial \mathbf{x}}^{\top} + \frac{\partial h}{\partial \mathbf{x}}\Sigma_{\mathbf{x}}\frac{\partial h}{\partial \mathbf{x}}^{\top}}_{\text{data-related}} + \underbrace{\frac{\partial h}{\partial \hat{\mathbf{y}}}\hat{\Sigma}_{\mathbf{y}}\frac{\partial h}{\partial \hat{\mathbf{y}}}^{\top}}_{\text{prediction-related}}. \tag{6}$$

The reformulated uncertainty includes partial derivatives of the measurements with respect to the input data $\mathbf{x}$ and network prediction $\hat{\mathbf{y}}$ and their variances. For convenience, we refer to the first term as data-related uncertainty and the second term as prediction-related uncertainty. In training time, the measruement uncertainty $\hat{\Sigma}_{\mathbf{z}}$ should be computed by using elements in (6). We make the network output data-related uncertainty in addition to the prediction uncertainty $\hat{\Sigma}_{\mathbf{y}}$ and compute the partial derivative $\partial h/\partial \hat{\mathbf{y}}$ from the measurement model.

## III. Uncertainty Balancing

Despite successful uncertainty training, a discrepancy between trained uncertainties from each network might occur depending on sensor measurement. This is critical when the uncertainty is trained in an unsupervised manner since no absolute scale is obtainable.

To solve this issue, this paper proposes covariance balancing that occurs during the training. To do so, we define the balancing loss as below.

$$\begin{aligned} L_{\text{balancing}} =& \frac{1}{N}\sum_{i=1}^{N} ||\mathbf{z}_i^a - \hat{\mathbf{z}}_i^a||^2_{\hat{\Sigma}_{\mathbf{z}_i^a}} + \log|\hat{\Sigma}_{\mathbf{z}_i^a}| \\ &+ \frac{1}{M}\sum_{j=1}^{M} ||\mathbf{z}_j^b - \hat{\mathbf{z}}_j^b||^2_{\hat{\Sigma}_{\mathbf{z}_j^b}} + \log|\hat{\Sigma}_{\mathbf{z}_j^b}| \\ &+ \underbrace{\frac{1}{K}\sum_{(i,j)\in\mathcal{K}} ||\hat{\mathbf{z}}_i^a - \hat{\mathbf{z}}_j^b||^2_{\hat{\Sigma}_{\mathbf{z}_i^a - \mathbf{z}_j^b}}}_{\text{inter-sensor consistency loss}}. \end{aligned} \tag{7}$$

This loss is defined as the sum of the unsupervised loss from each sensor measurement and the sensor consistency loss. This derivation allows the normalized uncertainties based on the direct comparison between sensors.

Here $\mathcal{K}$ is a set of indices of corresponding measurements between sensor $a$ and sensor $b$. The sensor consistency loss is computed when the measurements are from the same sensors. In some cases, we need conversion between measurements. For this purpose, we use the transformation between observations by using a transfer function $g_{a\to b}(\cdot)$ as

$$\hat{\mathbf{z}}^{b*} = g_{a\to b}(\hat{\mathbf{z}}^a). \tag{8}$$

In the above equation, $a$ and $b$ indicates each sensor modality.

## IV. Experiment

Follow the literature (UnDeepVO [4]), we initially use the depth and pose networks. Additional to these two networks, we add fully connected layers for the pose uncertainty and decoders for the data-related uncertainty. Next, we refine VO uncertainty via covariance balancing between two sensors.

The performance of the uncertainty estimation is provided in comparison to other methods. During the evaluation, the mean values were kept the same while changing uncertainty estimation methods. Unsupervised uncertainty means the estimated uncertainty without balancing. Unsupervised uncertainty consists of epistemic and aleatoric uncertainty. For supervised uncertainty, we additionally trained our network for 30 epochs using the supervised loss in (4) using the ground truth pose as a label. For the comparison baseline, we chose DICE [3] by implementing a DICE network predicting 6-DOF pose uncertainty from a single image.

The average log-likelihood of the estimated odometry is given in Table. I when verified over the KITTI test sequences

TABLE I: Average log-likelihood

| method | translation | rotation | all |
|---|---|---|---|
| Epistemic | $-42.8$ | $-4.98$ | $-54.5$ |
| Aleatoric | $-3.04 \times 10^6$ | $-8.10 \times 10^2$ | $-1.19 \times 10^{10}$ |
| Unsupervised | $-16.7$ | $-0.53$ | $-28.7$ |
| Supervised | 0.56 | 4.37 | 0.51 |
| Proposed | 0.63 | 3.54 | $-0.62$ |
| DICE [3] | $-16.43$ | 2.20 | $-20.26$ |

Average log-likelihood of uncertainty estimation methods computed from KITTI test dataset (sequence `09` and `10`).

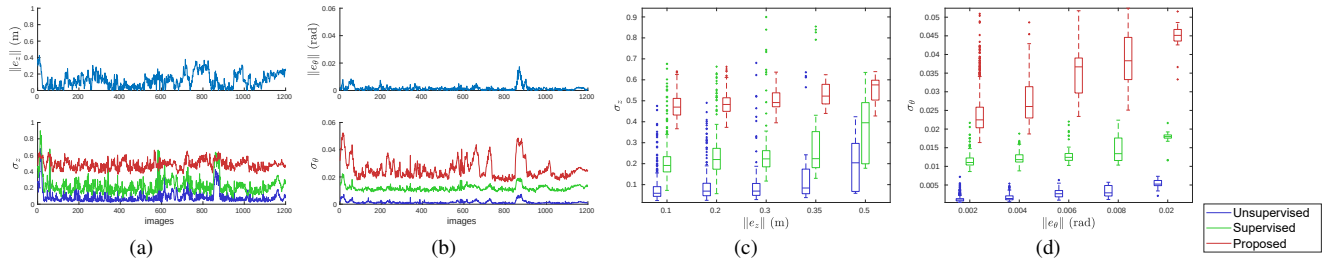


Fig. 1: Estimated pose errors and uncertainties. We compare the estimation among unsupervised, supervised and balanced approaches.

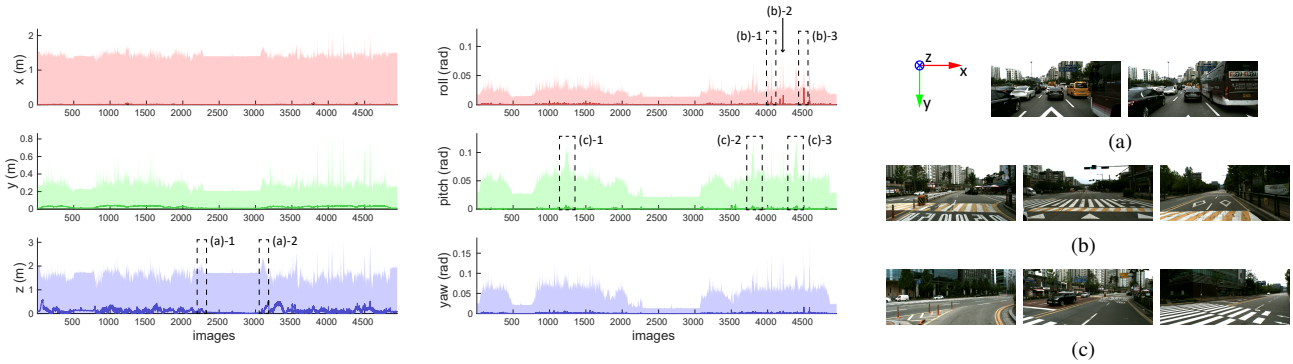


Fig. 2: The estimated covariance of the proposed method on KAIST urban dataset. The graphs shows translational and rotational errors in each axis and their $3\sigma$ bounds depicted with shaded regions. Thumbnail images on the right illustrate situations where large uncertainty occurs. (a) shows highly dynamic environments with moving cars where large z-axis uncertainties are captured. (b) represents when the car encounters with a speed bump, causing large uncertainty in roll motion. Large pitch errors occur at curved roads as shown in (c) and the estimated uncertainty reflects these errors.

[5]. Average log-likelihood reveals how the estimated uncertainty captures error magnitudes on average. The larger the value the better performance. Supervised learning (e.g., DICE) shows better performance since the supervised loss is negative of the average log-likelihood itself. Note that the proposed approach yields comparable numbers even when trained in an unsupervised manner. The balancing process enabled the network to learn absolute error magnitudes. Please note that the proposed uncertainty even better catches the error fluctuations than the supervised uncertainty does as seen in Fig. 1. The box plots (Fig. 1(c) and Fig. 1(d)) shows the uncertainty with respect to the actual error. As can be seen, the proposed method shows a steady increase.

Fig. 2 illustrates the learned VO uncertainty on the KAIST urban dataset [6]. The estimated uncertainty follows error fluctuations as seen in the $3\sigma$ value around large error variation. For example, the thumbnail images represent situations where uncertainty increases because of dynamic environments (Fig. 2(a)) and sudden motions (Fig. 2(b) and Fig. 2(c)). Also, the uncertainty is plausible because it captures relative magnitude of errors in each axis. For instance, larger uncertainty in the z-axis is measured since the driving data has large errors in the travel direction (z-axis).

## V. Conclusion

This paper proposed a general unsupervised uncertainty estimation using deep networks. We aimed to overcome the limitation of single sensor uncertainty learning by proposing balancing uncertainties between different sensors. As a validation, we applied the uncertainty estimation and balancing methods to end-to-end learning-based VO.

## References


[1] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in neural information processing systems*, 2017, pp. 5574–5584.

[2] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[3] K. Liu, K. Ok, W. Vega-Brown, and N. Roy, "Deep inference for covariance estimation: Learning gaussian noise models for state estimation," in *Proc. IEEE Intl. Conf. on Robot. and Automat.* IEEE, 2018, pp. 1436–1443.

[4] R. Li, S. Wang, Z. Long, and D. Gu, "UnDeepVO: Monocular visual odometry through unsupervised deep learning," in *Proc. IEEE Intl. Conf. on Robot. and Automat.*, 2018, pp. 7286–7291.

[5] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2012, pp. 3354–3361.

[6] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments," *International Journal of Robotics Research*, vol. 38, no. 6, pp. 642–657, 2019.