On learning visual odometry errors

Andrea De Maio[1] and Simon Lacroix[1]

Abstract— **This paper fosters the idea that deep learning methods can be sided to classical visual odometry pipelines to improve their accuracy and to associate uncertainty models to their estimations. We show that the biases inherent to the visual odometry process can be faithfully learned and compensated for, and that a learning architecture associated to a probabilistic loss function can jointly estimate a full covariance matrix of the residual errors, defining an error model capturing the heteroscedasticity of the process. Experiments on autonomous driving image sequences and micro aerial vehicles camera acquisitions assess the possibility to concurrently improve visual odometry and estimate an error associated to its outputs.**

I. Introduction

Visual odometry (VO) is a well established motion estimation process in robotics [1], successfully applied in a wide range of contexts such as autonomous cars or planetary exploration rovers. Seminal work resorted to stereovision: by tracking point features in images, 3D points correspondences are used to recover the motion between two stereovision acquisitions – the integration of elementary motions yielding an estimate of the robot pose over its course. Continuous work on VO led to a well established processes pipeline, composed of feature extraction, matching, motion estimation, and finally optimization. This scheme has been extended to single camera setups, in which case motions are estimated up to a scale factor, retrieved *e.g.* by fusing inertial information. Direct methods for VO have recently been proposed: they bypass the feature extraction process and optimize a photometric error [2]. These methods overcome the limits of sparse feature-based methods in poorly textured environments or in presence of low quality images (blurred), and they have proven to be on average more accurate.

The advent of convolutional neural networks (CNN) sprouted alternate solutions to VO. The full estimation process can be achieved by deep-learning architectures in an end-to-end fashion (see *e.g.* [3], [4], and especially [5] – note these work consider the monocular version of the problem, leaving the scale estimation untackled). In such approaches, the system has to learn the various information necessary to perform vision-based egomotion estimation, which can be a daunting task for a CNN.

The paper presented here builds upon existing work that exploits a CNN to predict *corrections* to classic stereo VO methods [6], aiming at improving their precision. This concurs with the idea that it is more beneficial to side deep-learning based methods with classical localization estimation processes rather than delegating the full pose estimation to a CNN. Our developments consider that visual odometry estimation errors do not have zero mean, as assessed in *e.g.* [7], [8], and provide corrections that improve the precision of VO. Furthermore, they provide a full error model for each computed motion estimation (in form of a Gaussian model), akin to [9]. This is a significant achievement, as it is generally complex to derive precise error models for geometrical VO methods.

II. Problem statement and related work

Consider a robot moving in a three dimensional environment. Let $\boldsymbol{x}_i \in \mathbb{R}^6$ (3 translations and 3 orientations) be its pose at time i in a given reference frame. The actual motion (ground truth) between time instants i and $i+1$ is represented by a homogeneous 4×4 transformation matrix ${}^{i}\mathbf{T}_{i+1}$.

A vision-based motion estimator uses raw image data $\boldsymbol{\mathcal{I}}_i \in \mathbb{R}^n$ to obtain an estimate ${}^{i}\hat{\mathbf{T}}_{i+1}$. In the VO case, the raw data $\boldsymbol{\mathcal{I}}_i$ is a pair of monocular or stereoscopic images captured at two different time instants i, $i+1$ (*i.e.* 2 or 4 images). The error $\boldsymbol{e}_i$ of VO is:

$$\boldsymbol{e}_i \; = \; {}^{i}\mathbf{T}_{i+1} \; \cdot \; {}^{i}\hat{\mathbf{T}}_{i+1}^{-1} \tag{1}$$

We can create a dataset $\mathcal{D} = \{\boldsymbol{\mathcal{I}}_i, \boldsymbol{e}_i | \forall i \in [1, d]\}$, where d is the size of the dataset. The literature provides two different approaches to leverage this type of dataset. The two approaches side a classic VO process with learning to either estimate *(i)* a *motion correction* to apply to ${}^{i}\hat{\mathbf{T}}_{i+1}^{-1}$, thus improving its accuracy [6], or *(ii)* an *error model* associated to ${}^{i}\hat{\mathbf{T}}_{i+1}^{-1}$ [9], thus allowing its fusion with any other motion or pose estimation process. Alternatively, with the same semantic, substituting errors with actual motion transforms, it is possible to directly learn the motion estimate and associated error [5].

A. Directly learning VO and an error model

The work in [5] introduces an end-to-end, sequence-to-sequence probabilistic visual odometry ("ESP-VO") based on a recurrent CNN. ESP-VO outputs both a motion estimate ${}^{i}\hat{\mathbf{T}}_{i+1}^{-1}$ and an associated error. The learned error model is a diagonal covariance matrix, hence not accounting for possible correlations between the different dimensions of the motions. It is unclear how the probabilistic loss is mixed to the mean squared error of the Euclidean distance between the ground truth and the estimated motions, and they make use of a hand-tuned scaling factor to balance rotation and translation. The article presents significant results obtained on a large variety of datasets, with comparisons to state-of-the-art VO schemes. The results show that ESP-VO is a serious alternative to classic schemes, all the more since

[1] LAAS-CNRS, Université de Toulouse, CNRS 7, Avenue du Colonel Roche, 31031 Toulouse, France, E-mail: {andrea.de-maio,simon.lacroix}@laas.fr

it also provides the variances associated to the estimations. Yet, they are analysed over whole trajectories, which inherit from the random walk effect of motion integration, and as such do not provide thorough statistical insights – *e.g.* on the satisfaction of the gaussianity of the error model.

B. Learning corrections to VO

The work presenting DPC-net [6] learns an estimate of $\boldsymbol{e}_i$, which is further applied to the VO estimate ${}^{i}\hat{\mathbf{T}}_{i+1}^{-1}$ to improve its precision. The authors introduce an innovative pose regression loss based on the SE(3) geodesic distance modelled with a vector in Lie algebra coordinates. Instead of resorting to a scalar weighting parameter to generate a linear combination of the translation and rotation errors, the proposed distance function naturally balances these two types of errors. The loss takes the following form:

$$\mathcal{L}(\boldsymbol{\xi}) = \frac{1}{2} g(\boldsymbol{\xi})^{\mathsf{T}} \boldsymbol{\Sigma}^{-1} g(\boldsymbol{\xi}) \tag{2}$$

where $\boldsymbol{\xi} \in \mathbb{R}^6$ is a vector of Lie algebra coordinates estimated by the network, $g(\boldsymbol{\xi})$ computes the equivalent of (1) in the Lie vector space, and $\boldsymbol{\Sigma}$ is an empirical average covariance of the estimator pre-computed over the training dataset. The paper provides statistically significant results that show DPC-net improves a classic feature-based approach, up to the precision of a dense VO approach. In particular, it alleviates biases (*e.g.* due to calibration errors) and environment factors. Worth to notice is that the authors interlace the corrections estimated at lower rate than the underlying VO process with VO, which processes all the images, using a pose-graph relaxation approach.

C. Learning an error model of VO

Inferring an error model for VO comes to learn the parameters of a predefined distribution to couple VO with uncertainty measures. The work in [9] introduces DICE (Deep Inference for Covariance Estimation), which learns the covariance matrix of a VO process as a maximum-likelihood for Gaussian distributions. Nevertheless, it considers the distribution over measurement errors as a zero-mean Gaussian $\mathcal{N}(0, \boldsymbol{\Sigma})$. Such model is acceptable for unbiased estimators, which unfortunately it is often not the case of VO. Yey, the authors show that their variance estimates are highly correlated with the VO errors, especially in case of difficult environment conditions, such as large occlusions.

III. Simultaneously learning corrections and uncertainties

To jointly estimate a correction to the VO process and a full error model *after having applied the correction*, we expand the network structure of [9] adding a vector $\boldsymbol{\mu}_i \in \mathbb{R}^6$ to the output layer, which is incorporated in the negative log-likelihood loss that is derived as follows. Given a dataset $\mathcal{D}$ of size d, where the observations $\{\boldsymbol{e_1}, \ldots, \boldsymbol{e_d}\}^{\mathsf{T}}$ of VO errors are assumed to be independently drawn from a multivariate Gaussian distribution, we can estimate the parameters of the Gaussian as

$$\underset{\boldsymbol{\mu}_{1:d}, \boldsymbol{\Sigma}_{1:d}}{\arg\max} \sum_{i=1}^{d} p(\boldsymbol{e}_i | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{3}$$

This is equivalent to minimize the negative log-likelihood (NLL)

$$\underset{\boldsymbol{\mu}_{1:d}, \boldsymbol{\Sigma}_{1:d}}{\arg\min} \sum_{i=1}^{d} - \log \left(p(\boldsymbol{e}_i | \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \right) \tag{4}$$

$$= \underset{\boldsymbol{\mu}_{1:d}, \boldsymbol{\Sigma}_{1:d}}{\arg\min} \sum_{i=1}^{d} \log |\boldsymbol{\Sigma}_i| + (\boldsymbol{e}_i - \boldsymbol{\mu}_i)^{\mathsf{T}} \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{e}_i - \boldsymbol{\mu}_i) \tag{5}$$

$$\begin{aligned} \approx \underset{f_{\boldsymbol{\mu}_{1:d}}, f_{\boldsymbol{\Sigma}_{1:d}}}{\arg\min} & \sum_{i=1}^{d} \log |f_{\boldsymbol{\Sigma}_i}(\boldsymbol{\mathcal{I}}_i)| + \\ & (\boldsymbol{e}_i - f_{\boldsymbol{\mu}_i}(\boldsymbol{\mathcal{I}}_i))^{\mathsf{T}} f_{\boldsymbol{\Sigma}_i}(\boldsymbol{\mathcal{I}}_i)^{-1} (\boldsymbol{e}_i - f_{\boldsymbol{\mu}_i}(\boldsymbol{\mathcal{I}}_i)) \end{aligned} \tag{6}$$

We split the output of the network in two different parts: the mean vector $f_{\boldsymbol{\mu}_i}(\boldsymbol{\mathcal{I}}_i)$ and the covariance matrix $f_{\boldsymbol{\Sigma}_i}(\boldsymbol{\mathcal{I}}_i)$, where $f(\boldsymbol{\mathcal{I}}_i)$ represents the full output given a pair of stereo images.

To enforce a positive definite covariance matrix we use the LDL matrix decomposition [9]. $f_{\boldsymbol{\Sigma}_i}(\boldsymbol{\mathcal{I}}_i)$ is reformulated as a vector $\boldsymbol{\alpha}_i = [\boldsymbol{l}_i, \boldsymbol{d}_i]^{\mathsf{T}}$ with $\boldsymbol{l}_i \in \mathbb{R}^{\frac{(n^2-n)}{2}}$ and $\boldsymbol{d}_i \in \mathbb{R}^n$. We have then

$$\boldsymbol{\Sigma}_i \approx L(\boldsymbol{l}_i) D(\boldsymbol{d}_i) L(\boldsymbol{l}_i)^{\mathsf{T}} \tag{7}$$

where $\boldsymbol{l}_i$ and $\boldsymbol{d}_i$ are the vectors containing the elements of the respective $\boldsymbol{L}$ and $\boldsymbol{D}$ matrices. The LDL decomposition is unique and exists as long as the diagonal of D is strictly positive. This can be enforced using the exponential function $exp(\boldsymbol{d_i})$ on the main diagonal. Thanks to some additional properties around the computation of its log determinant the first term of (6) can be simplified as $sum(\boldsymbol{d}_i)$, that is the sum of the elements of the vector $\boldsymbol{d}_i$. In the second term $f_{\boldsymbol{\Sigma}_i}(\boldsymbol{\mathcal{I}}_i)^{-1}$ is replaced by the LDL product. Replacing $f_{\boldsymbol{\mu}_i}(\boldsymbol{\mathcal{I}}_i)$ with the mean output vector $\hat{\boldsymbol{\mu}}_i$ we finally obtain

$$\begin{gathered} \mathcal{L}(\boldsymbol{\mathcal{I}_{1:d}}) = \underset{\hat{\boldsymbol{\mu}}_{1:d}, \boldsymbol{\alpha}_{1:d}}{\arg\min} \sum_{i=1}^{d} sum(\boldsymbol{d}_i) + \\ (\boldsymbol{e}_i - \hat{\boldsymbol{\mu}}_i)^{\mathsf{T}} (L(\boldsymbol{l}_i) D(exp(\boldsymbol{d}_i)) L(\boldsymbol{l}_i)^{\mathsf{T}})^{-1} (\boldsymbol{e}_i - \hat{\boldsymbol{\mu}}_i) \end{gathered} \tag{8}$$

Formulating the problem as in (8), the loss function recalls the formulation of the Lie algebra loss in (2). The covariance matrix in this case is learned in relation to the input, capturing the heteroscedastic uncertainty of each sample. The learned covariance matrix acts as in [10], weighting position and orientation errors. The main difference resides in the nature of the learned uncertainty, homoscedastic vs heteroscedastic: through back-propagation with respect to the input data, [11], we aim to learn a heteroscedastic error.

Assuming that errors can be drawn from a distribution $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, estimating $\boldsymbol{\mu}_i$ corresponds to predicting the maximum likelihood value that the error model can assume. This corresponds to the desired correction in our case. At the same time, we estimate a covariance matrix $\mathbf{\Sigma}_i$, returning an uncertainty measure relative to each particular input and predicted correction.

IV. Experiments

A. Setup

We use the open-source VO implementation *libviso2* [12]. It is a feature-based approach, that uses a Kalman Filter in combination with a RANSAC scheme to produce SE(3) estimates using rectified stereoscopic pairs.

1) Datasets: We exploit two different datasets: the KITTI dataset [13] and the Euroc Micro Air Vehicles dataset [14].

KITTI provides various sequences of rectified images acquired while driving in urban areas. We tested training the network using several combinations and obtained consistent results splitting train and validation trajectories in different configurations: for all results shown here we trained using sequences `04` to `10` excluding one or two for validation purposes. The estimated motions are expressed in camera frame (z axis pointing forward), and the Tait-Bryan angles are defined w.r.t. this reference frame (*e.g.* yaw encodes rotations around the optical axis z).

Euroc contains different stereoscopic sequences recorded in different environments. We rectified the images in order to use them as input for both the VO process and the network. We use the first three sequences `MH_01` to `MH_03`. Note that in some cases VO fails to produce a pose estimate (mainly when images have a strong motion blur): these data have simply been discarded from the dataset. Contrary to the KITTI dataset, Euroc estimations and ground truth are expressed in the vehicle body frame (x pointing forward, y leftward).

2) Network structures: We initially compared the results produced using the architectures in DPC-net [6] and DICE [9]. The first trial was to adapt the loss in (8) to DPC-net. We noticed that the mean output vector was still rather constant throughout entire trajectories, regardless of the dataset, and the same behavior was experienced using the loss in (2). Similar tests were conducted with DICE. We experienced problems in reducing the average mean error along the six dimensions, and an increase in the standard deviation. Alleging these issues as being caused by the shallow architecture of DICE, we modified its network structure, first removing the max pool layers to preserve spatial information [15], and achieving dimension reduction by setting the stride to 2 in early layers. We also increased the number of convolutional filters to tackle the estimation of both the corrections and error model, adding 50% dropout after each layer to prevent over-fitting. We kept respective nonlinear activation function using parametric ReLu for DPC-net and leaky ReLu for DICE. For the rest of the paper, we refer to this network as Deeper-DICE (D-DICE, Table I).

The convolutional layers are followed by two fully connected layer, respectively composed of 2048 and 27 output units. In the six-dimensional case we need 21 values for the LDL decomposition and 6 for the mean vector. No rectification or dropout is applied to the last fully connected layer.

We trained using Adam optimizer with a learning rate of 1e-04 and halted the learning when test and train loss start diverging. All the experiments have been carried out using using an Nvidia GeForce RTX 2080 Ti with a batch size of 64.

Layer	Kernel size	Stride	Number of channels
conv1	5x5	2	64
conv2	5x5	2	128
conv3	3x3	2	256
conv4	3x3	2	512
conv5	3x3	1	1024

TABLE I: D-DICE convolutional architecture.

B. Evaluation

Most of the results presented in this section aim to analyse the improvements brought by the neural network to VO, by complementing it with SE(3) pose corrections and an error estimate.

1) Loss comparison: Here we compare the results in terms of corrections between the loss based on the Lie algebra formulation (2) and our full negative log-likelihood loss (8). Both are evaluated on DPC-net.

Table II shows that corrections learned with the Lie loss and extracted from the Gaussian model learned minimizing the negative log-likelihood. In the first two columns we show the average mean error produced by *libviso2* and its standard deviation. The second and the third column show the same statistics after the corrections produced by DPC-net, respectively optimized using the Lie loss and the NLL, have been applied to VO. These results have been obtained after training without sequences `05`, left for validation and `06`, used for testing.

None of the two losses comes out as a clear winner, even if both improve the mean error along five out of six dimensions compared to VO alone. Note small increases in the average standard deviations using NLL loss, which are not significant compared to the improvements over the average mean error. Of course, the advantage of using the proposed NLL loss is to jointly provide an uncertainty estimation.

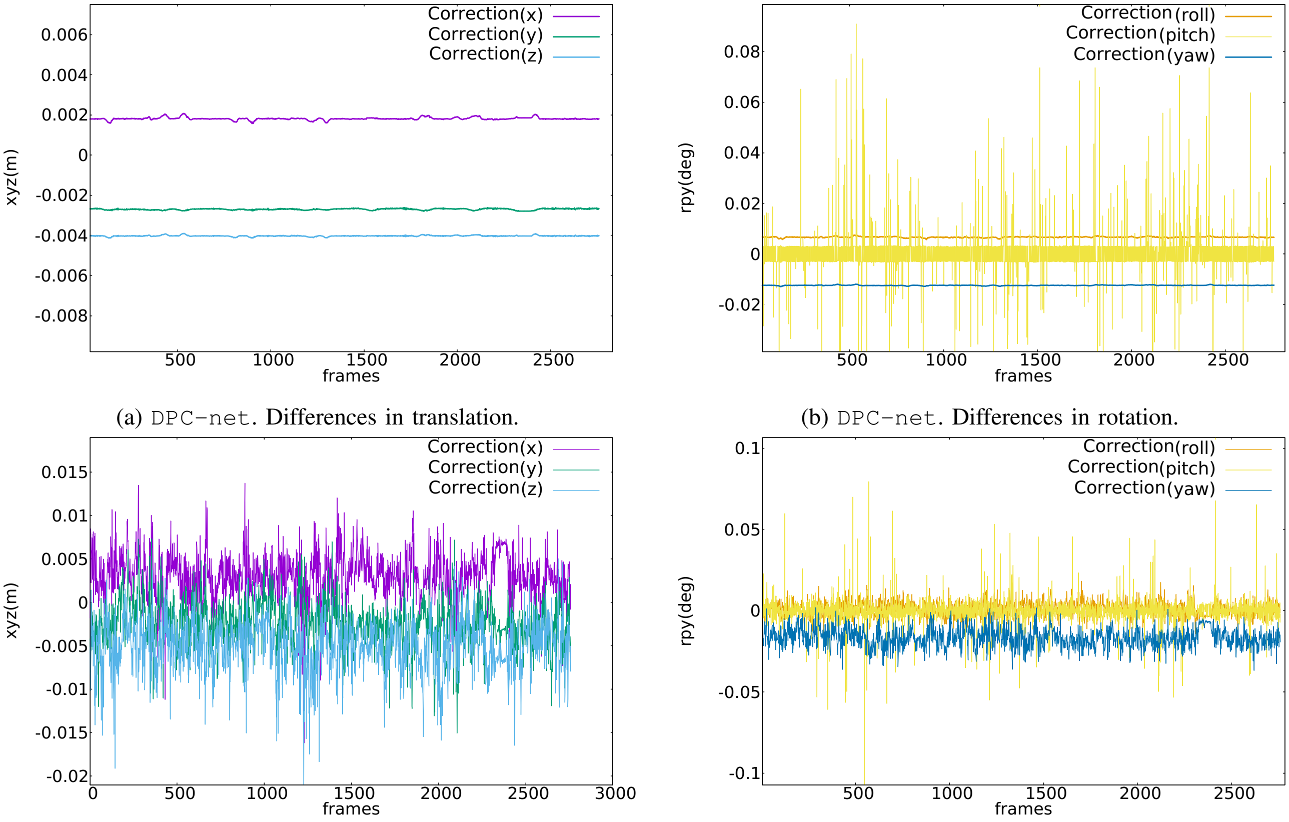
2) Architecture comparison: We noticed that DPC-net tends to output rather constant corrections throughout a whole sequence, certainly compensating biases. This was one of the driving factor for exploring other network architectures. D-DICE behaves differently, as can be seen in Fig. 1, exhibiting more data-dependant corrections.

3) Uncertainty estimation: Here we inspect the error estimates produced by D-DICE. Since we assumed a Gaussian error model $\sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma})$, a common way to measure its relevance is to check the fraction of samples that do not respect the following inequality:

$$\mu_i - n\sigma_i \leq e_i \leq \mu_i + n\sigma_i \tag{9}$$

	μ_{VO}	σ_{VO}	$\mu_{corr}\ Lie$	$\sigma_{corr}\ Lie$	$\mu_{corr}\ NLL$	$\sigma_{corr}\ NLL$
x	0.20	0.60	**0.07**	0.60	-0.92	0.66
y	-0.25	0.58	0.07	0.58	**-0.02**	0.62
z	-0.62	0.95	-0.25	0.95	**-0.12**	0.98
$roll$	1.21	28.17	-0.98	28.16	**-0.2**	28.55
$pitch$	**0.81**	28.16	**0.82**	28.13	1.01	28.06
yaw	-14.32	30.59	**0.28**	30.60	2.26	30.85

TABLE II: `KITTI, sequence 05`. Statistics for VO with and without corrections, applied using DPC-net trained with the two losses in (2) and (8). Units are cm for x, y, z and $mrad$ for the angles.



(a) `DPC-net`. Differences in translation.

(b) `DPC-net`. Differences in rotation.

(c) `D-DICE`. Differences in translation.

(d) `D-DICE`. Differences in rotation.

Fig. 1: The plots show the different nature of the learned corrections using DPC-net and D-DICE (Euroc dataset). Each plot shows the difference between VO estimates and the corrected estimates produced by the network.

where e_i is the error along the dimension i, and μ_i, σ_i respectively are the mean and standard deviation predicted for the input associated to $\boldsymbol{e}$ on the i-th dimension. The parameter n is the number of considered standard deviations. We test against the three sigma interval ($n = 3$) that, in case of samples drawn from a normal distribution, covers $\sim 99.7\%$ of the samples.
Fig. 2 and 3 show the predicted 3-sigma standard deviation, shifted by the predicted mean, against the ground truth error, respectively for a sequence from Euroc and KITTI.

The corresponding statistics are displayed in Table III and IV. The results are positive, proposing an uncertainty model that is not over-pessimistic but still manages to approximate well a normal distribution.

Fig. 4 shows D-DICE uncertainty bounds trained considering the visual odometry error $\sim \mathcal{N}(0, \boldsymbol{\Sigma}_i)$, in pink, and $\sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, in light blue. Fig. 4a and 4b respectively display VO errors without and with corrections and with their predicted standard deviations (along z, in the KITTI case). The error is more zero-centered in the second case and the uncertainty model fits well the corrected error. In Fig. 4c, the two uncertainty models are overlaid. When the loss does not encorporate the mean, the error model is more pessimistic without being significantly more accurate than its counterpart estimating the mean. Our proposed loss already achieves a good precision on the uncertainty estimation in the 3-sigma interval, as shown in Table IV. The results from considering an unbiased error yield a non-significant increase

	σ	2σ	3σ
x	77.07%	95.99%	99.39%
y	69.88%	92.18%	97.20%
z	78.34%	95.00%	99.36%
$roll$	61.85%	86.27%	95.30%
$pitch$	68.67%	94.72%	99.33%
yaw	82.59%	96.90%	98.97%

TABLE III: `Euroc, sequence MH_01`. Percentages of samples that lie in the various sigma-intervals around the mean. Mean and standard deviations are produced by D-DICE and correspond to Fig. 2.

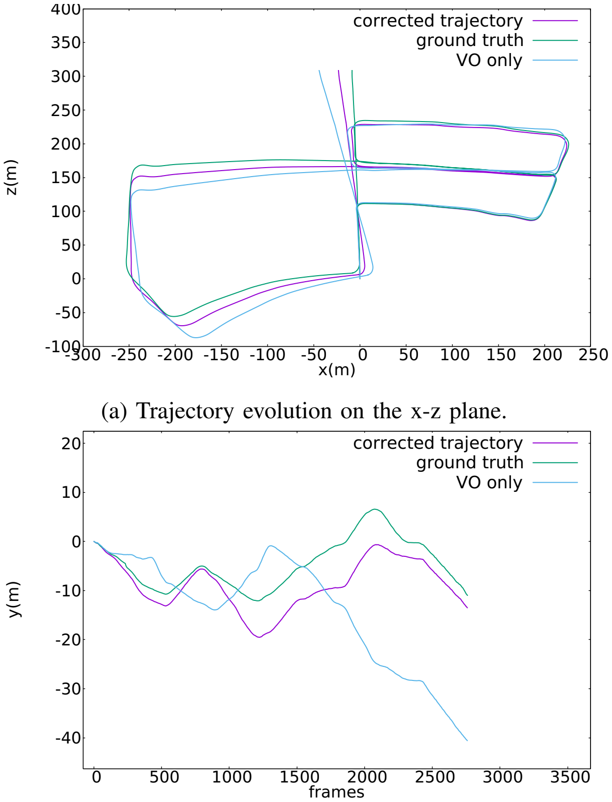
	σ	2σ	3σ
x	87.24%	98.91%	99.63%
y	91.15%	99.42%	99.67%
z	79.49%	96.12%	98.94%
$roll$	75.72%	94.89%	98.69%
$pitch$	72.97%	95.14%	98.76%
yaw	70.57%	93.55%	98.69%

TABLE IV: `KITTI, Sequence 05`. Percentages of samples that lie in the various sigma-intervals around the mean. Mean and standard deviations are produced by D-DICE and correspond to Fig. 3.

of precision, as we already sit around 99%, at the cost of being more uncertain. For sake of completeness, we show the outcome of corrections in Fig. 5a. A major improvement was encountered in several trajectories especially on the vertical axis (y-axis, in KITTI, see Fig. 5b).

V. Conclusions

We presented an insight into the learning of errors in visual odometry. Relying on existing state-of-the-art techniques, we iterated on analysing what type of error and uncertainty can be learned by deep neural networks. We concentrated our efforts on approaches that can be paired with classical visual odometry pipelines, in order to ease the work done by the network and exploiting the well established feature-based processes. We demonstrated that it is possible to assimilate the distribution over visual odometry errors to Gaussians, and proceeded to cast the error prediction to a full maximum likelihood for normal distributions case. Knowing that the errors are biased, we model such Gaussians as non-zero mean distributions, showing the beneficial aspects of this approach compared to works that rely only on the estimation of the covariance matrix. On the other hand, we pair visual odometry corrections with a more precise model, inferred thanks to the assumption of biased distributions. In future we would like to explore similar approaches, with different perception processes that are yet to associated with precise error models, *e.g.* iterative closest points algorithm based on LiDAR scans [16].



(a) Trajectory evolution on the x-z plane.

(b) Y-estimate evolution over time.

Fig. 5: `KITTI, sequence 05`. Trajectory with and without corrections.

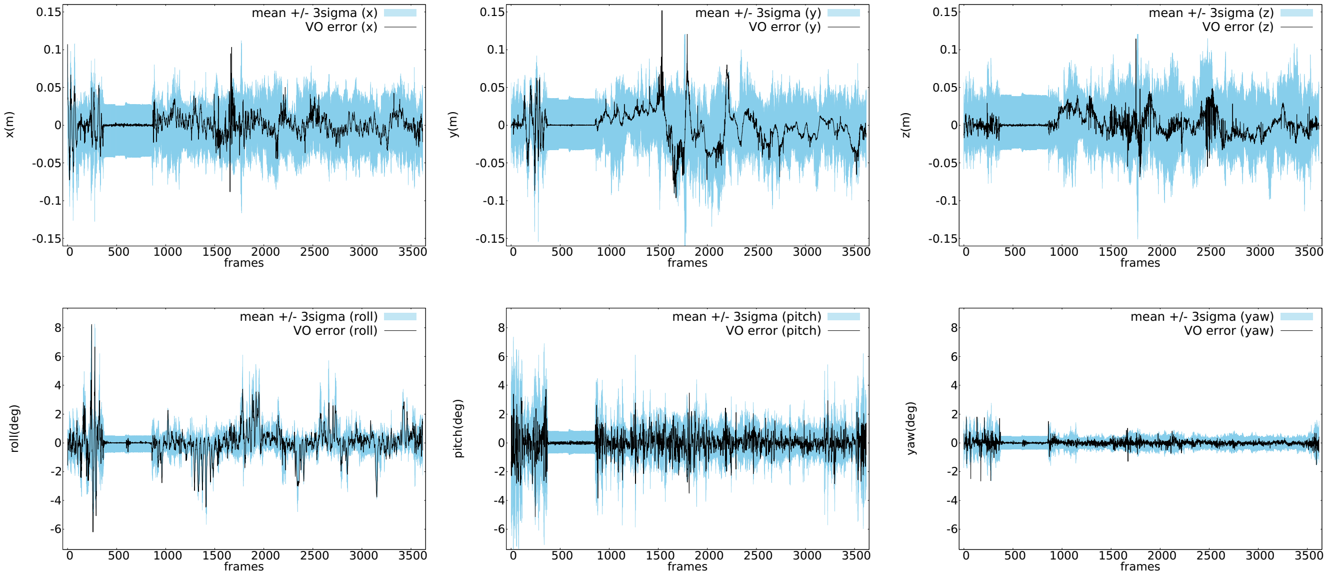


Fig. 2: D-DICE. Euroc, MH_01. Uncertainty prediction in the six dimensions (translation top, rotation bottom).

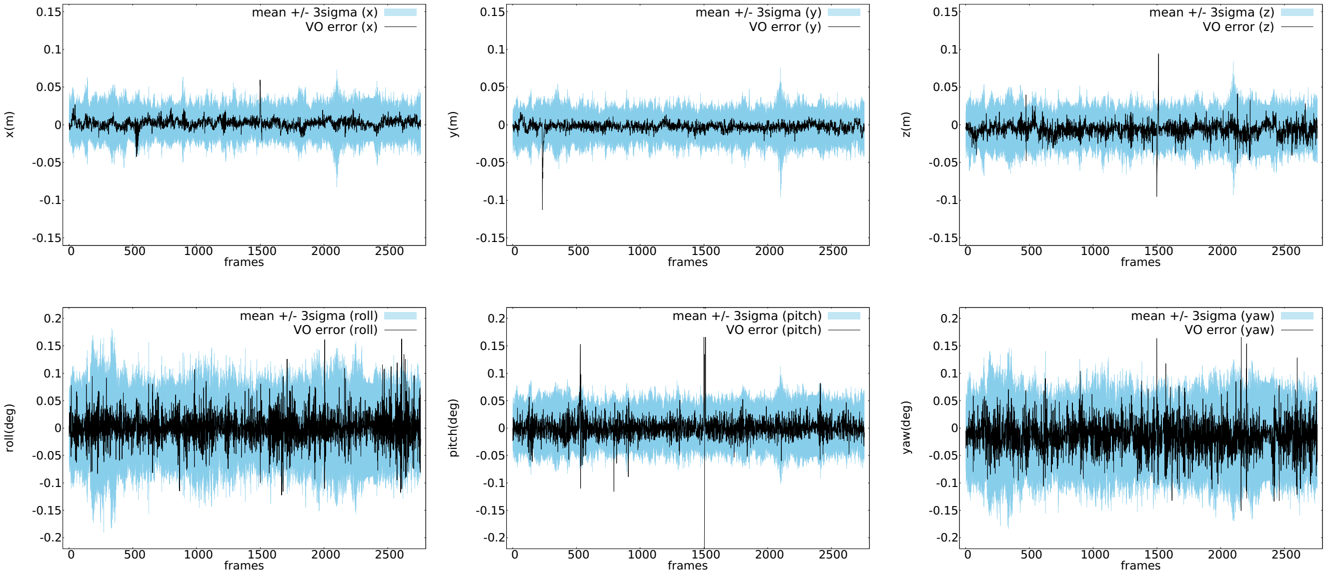
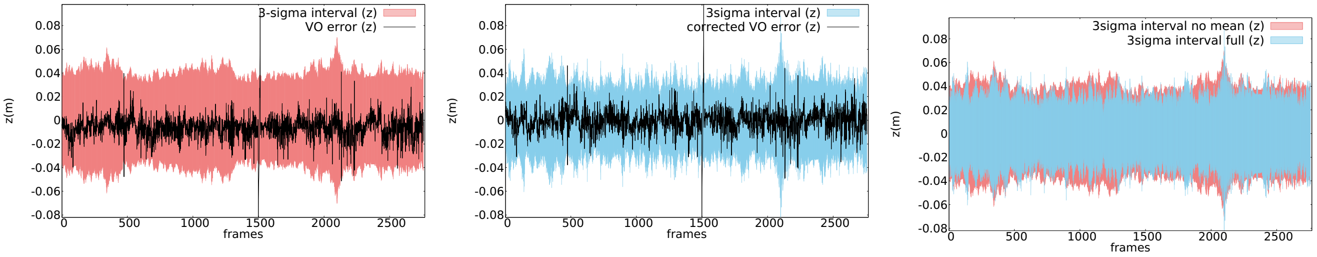


Fig. 3: D-DICE. KITTI, 05. Uncertainty prediction in the six dimensions (translation top, rotation bottom).



(a) KITTI, sequence 05. Standard deviation prediction without mean estimation (pink) and VO errors (black).

(b) KITTI, sequence 05. Standard deviation prediction trained estimating the mean (blue) and corrected VO errors (black).

(c) KITTI, sequence 05. Comparison between the two standard deviations.

Fig. 4: Comparison between the same uncertainty model (on one axis) learned without and with mean estimation.

References

[1] D. Scaramuzza and F. Fraundorfer, “Visual odometry [tutorial],” *IEEE robotics & automation magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[2] J. Engel, V. Koltun, and D. Cremers, “Direct sparse odometry,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[3] K. R. Konda and R. Memisevic, “Learning visual odometry with a convolutional network.,” in *VISAPP (1)*, pp. 486–490, 2015.

[4] R. Li, S. Wang, Z. Long, and D. Gu, “Undeepvo: Monocular visual odometry through unsupervised deep learning,” *arXiv preprint arXiv:1709.06841*, 2017.

[5] S. Wang, R. Clark, H. Wen, and N. Trigoni, “End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks,” *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 513–542, 2018.

[6] V. Peretroukhin and J. Kelly, “Dpc-net: Deep pose correction for visual localization,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2424–2431, 2017.

[7] G. Dubbelman, P. Hansen, and B. Browning, “Bias compensation in visual odometry,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2828–2835, IEEE, 2012.

[8] V. Peretroukhin, J. Kelly, and T. D. Barfoot, “Optimizing camera perspective for stereo visual odometry,” in *2014 Canadian Conference on Computer and Robot Vision*, pp. 1–7, IEEE, 2014.

[9] K. Liu, K. Ok, W. Vega-Brown, and N. Roy, “Deep inference for covariance estimation: Learning gaussian noise models for state estimation,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1436–1443, IEEE, 2018.

[10] A. Kendall and R. Cipolla, “Geometric loss functions for camera pose regression with deep learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5974–5983, 2017.

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

[12] B. Kitt, A. Geiger, and H. Lategahn, “Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme,” in *2010 ieee intelligent vehicles symposium*, pp. 486–492, IEEE, 2010.

[13] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3354–3361, IEEE, 2012.

[14] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, “The euroc micro aerial vehicle datasets,” *The International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.

[15] A. Handa, M. Bloesch, V. Pătrăucean, S. Stent, J. McCormac, and A. Davison, “gvnn: Neural network library for geometric computer vision,” in *European Conference on Computer Vision*, pp. 67–82, Springer, 2016.

[16] F. Pomerleau, F. Colas, R. Siegwart, *et al.*, “A review of point cloud registration algorithms for mobile robotics,” *Foundations and Trends® in Robotics*, vol. 4, no. 1, pp. 1–104, 2015.