MDN-VO: Estimating Visual Odometry with Confidence

Nimet Kaygusuz, Oscar Mendez, Richard Bowden

Abstract— **Visual Odometry (VO) is used in many applications including robotics and autonomous systems. However, traditional approaches based on feature matching are computationally expensive and do not directly address failure cases, instead relying on heuristic methods to detect failure. In this work, we propose a deep learning-based VO model to efficiently estimate 6-DoF poses, as well as a confidence model for these estimates. We utilise a CNN - RNN hybrid model to learn feature representations from image sequences. We then employ a Mixture Density Network (MDN) which estimates camera motion as a mixture of Gaussians, based on the extracted spatio-temporal representations. Our model uses pose labels as a source of supervision, but derives uncertainties in an unsupervised manner. We evaluate the proposed model on the KITTI and nuScenes datasets and report extensive quantitative and qualitative results to analyse the performance of both pose and uncertainty estimation. Our experiments show that the proposed model exceeds state-of-the-art performance in addition to detecting failure cases using the predicted pose uncertainty.**

I. Introduction

Traditional VO approaches have been studied for decades and supported real world applications in robotics, computer vision and autonomous driving. Even though these algorithms perform well in ideal conditions, they are not robust, perform poorly in low texture environments and are prone to failure under fast motion [26]. They also do not have built-in reliability estimation which limits their ability to recover from failure cases. As traditional VO algorithms do not provide a reliability measure, systems that depend upon their output, such as path planning [39], vehicle state estimation [33] etc., are therefore susceptible to failure which can lead catastrophic outcomes.

Recent years have seen a move to pose estimation using learning based approaches [31], [15], [34]. Unlike traditional VO algorithms, learning-based methods exploit the availability of large scale datasets to learn from the data itself [14]. This enables them to be robust to conditions such as low texture areas and challenging lighting conditions without requiring accurate camera calibration [4].

Motivated by the recent success of deep learning based approaches and to address the pose estimation reliability, in this work we present a novel deep learning based VO estimation approach that includes uncertainty. We work on raw images and extract image representations using a Convolutional Neural Network (CNN) [14]. We then model the vehicle motion in the image sequences using an Recurrent Neural Network (RNN) [11]. Unlike most approaches, which regress a single pose [19], [31], we employ a Mixture Density Network (MDN) [2] to regress a mixture model of 6-DoF poses (see Figure 1). This approach enables us to estimate pose and uncertainty using the predicted mixture model characteristics which are means (μ), standard deviations (σ) and mixture coefficients (α).

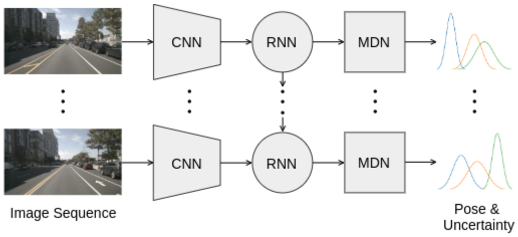


Fig. 1: An overview of the proposed monocular VO and uncertainty approach using MDN.

We evaluate our approach on two challenging public datasets, namely KITTI [8] and nuScenes [3]. We report quantitative and qualitative pose estimation and VO reliability results. We show that uncertainty estimates negatively correlate with pose estimation accuracy. This indicates that the proposed approach can successfully predict VO reliability without requiring additional supervision.

The contributions of this paper can be summarised as:

- We propose a learning-based VO estimation approach that employs a mixture of probability distributions to enable uncertainty estimation.
- We demonstrate state-of-the-art performance on the KITTI and the recently released nuScenes datasets, which includes challenging scenarios such as night-time and fast motion.
- We provide extensive analysis for uncertainty estimations and demonstrate how they reflect pose estimation failures.

The rest of the paper is structured as follows: In Section II we discuss traditional and learning based approaches to VO. We then introduce our approach in Section III. In Section IV we share our experimental setup and report qualitative and quantitative results. We conclude in Section V by discussing our findings and possible future work.

II. Related Work

Traditional feature based methods to VO estimation have been studied for the last two decades [22], [20]. They estimate the motion of the camera by extracting and matching a group of hand crafted feature points (e.g. SIFT [17], SURF [1] or ORB [24]) on consecutive frames. However, these approaches are prone to failure in scenarios where there are

All authors are with the University of Surrey {n.kaygusuz, o.mendez, r.bowden}@surrey.ac.uk

insufficient features, such as in low texture environments or poor illumination [26]. To address this issue, Newcombe et al. [21] proposed a dense direct approach, which estimates the motion by optimising image pixel intensities. However, this approach requires extensive computational power. To mitigate the heavy computational requirements, sparse direct methods have been introduced. Engel et al. [7] proposed only using image pixels which have high gradient intensities. Compared to feature based approaches, direct methods are more suited to low texture environments but require a good initialisation and are sensitive to rapid motion and photometric change.

More recently, deep learning based approaches have dominated computer vision [14]. Inspired by this, the field has adapted CNNs to extract and match feature points and to estimate VO [30]. DeTone et al. [5] present a learning-based approach to detect interest point detectors and descriptors. Sarlin et al. [25] propose a neural network to learn to match two sets of pre-existing features. End-to-end VO approaches have also been studied: Mohanty et al. [19] propose combining two CNNs to extract features from sequential images. They estimate pose by concatenating features and passing them through fully connected layers. To enhance the temporal modelling capabilities, Wang et al. [31], [32] propose using RNNs to model changes in image features and achieve competitive results to traditional VO based approaches.

Most deep learning based VO approaches are trained in a supervised manner, which requires labelled data. However, collecting and annotating large amounts of data is a laborious task. Alternatively, unsupervised learning based approaches [15], [35], [36], [37] have the ability to exploit vast amounts of unlabelled data. Zhou et al. [38] use view synthesis and learning depth and pose estimation together. They use a photometric error based loss function to train their networks. However, their approach is not able to recover the global scale. To solve this issue, Zhan et al. [36] and Li et al. [15] propose using stereo image pairs to estimate scaled VO in an unsupervised manner. Li et al. [16] present a meta-learning algorithm for a better adaptation to unseen environments. Although unsupervised VO methods achieve promising results and allow the models to be trained on a large variety of unlabelled data, their current performance is lower than supervised methods.

More recently Yang et al. [34] proposed a hybrid approach, which combines unsupervised deep learning and classical direct VO estimation methods. They estimate depth, pose and photometric uncertainty from images. They then feed these outputs to a direct VO estimation method [7] to obtain the final trajectories.

Uncertainty estimation has been studied for similar tasks, such as camera localisation. Kendall et al. [13] model camera localisation uncertainty by utilising dropout at test time. However this approach can be considered as model uncertainty estimation [4], whereas in this work we approach uncertainty estimation as a sensor reliability metric, which can ultimately be used for sensor fusion.

Inspired by the recent success of deep learning based VO models, in this work we propose a novel architecture where we utilise MDNs [2] to regress the 6-DoF poses as a mixture of distributions, which allows the network to learn multiple modes from the data.

III. Methodology

In this section, we introduce the proposed end-to-end approach for estimating visual odometry and its confidence/uncertainty. Our model takes in a sequence of images, $\mathcal{V} = \{I_0, ..., I_T\}$, with $(T+1)$ number of frames, and predicts a mixture of distributions, $\mathcal{G}$, with M components, that are most likely to produce the ground truth relative poses, $Y = \{y_{(0,1)}, ..., y_{(T-1,T)}\}$. We use Gaussian distributions, $\mathcal{N}$, as mixture components to model relative poses. Thus, we can notate our mixture model $\mathcal{G}$ as:

$$\mathcal{G} = \{\alpha_t^1 \mathcal{N}_t(\mu^1, \sigma^1), ..., \alpha_t^M \mathcal{N}_t(\mu^M, \sigma^M)\}^{t=1:T} \tag{1}$$

We break the proposed architecture into three main modules, namely *Visual Feature Extraction*, *Temporal Modelling* and *Mixture Density Estimation*. In the visual feature extraction module, we extract features from consecutive image pairs using a CNN. We then pass the extracted features to the temporal modelling layer, which utilises an RNN to model the changes in visual features over time. Finally, the RNN outputs are fed into the mixture density estimation module, which estimates the relative poses for each frame with respect to its predecessor. An overview of the proposed approach can be seen in Figure 2. In the rest of this section we describe each component in detail.

A. Visual Feature Extractor

Traditional VO approaches start by extracting geometric features from image sequences. By matching the feature points between image pairs, these models estimate the camera motion. Following the same concept, learning based approaches [19], [31], propose learning representations of distinctive image features using CNNs.

We develop our CNN backbone structure similar to Flownet [6]. It is designed to learn the features that are most suitable to represent image motion in the context of optical flow, which is obviously a related task to VO estimation. The network consists of 9 convolutional layers with a receptive filter size of $7x7$ for the first layer, $5x5$ for the next two and $3x3$ for the rest. We utilise batch normalisation [12], Leaky ReLU [18] and Dropout [28] after each convolutional layer.

Given an image sequence, $\mathcal{V}$, our model starts by concatenating consecutive image pairs (I_{t-1}, I_t) and extracting features, f_t as:

$$f_t = \text{CNN}([I_{t-1}, I_t]) \tag{2}$$

where $[.]$ is a concatenation operation over the image colour channels. Extracted image features $\{f_1, ..., f_T\}$ are then passed to the next module, which learns to model the temporal changes in the latent space.

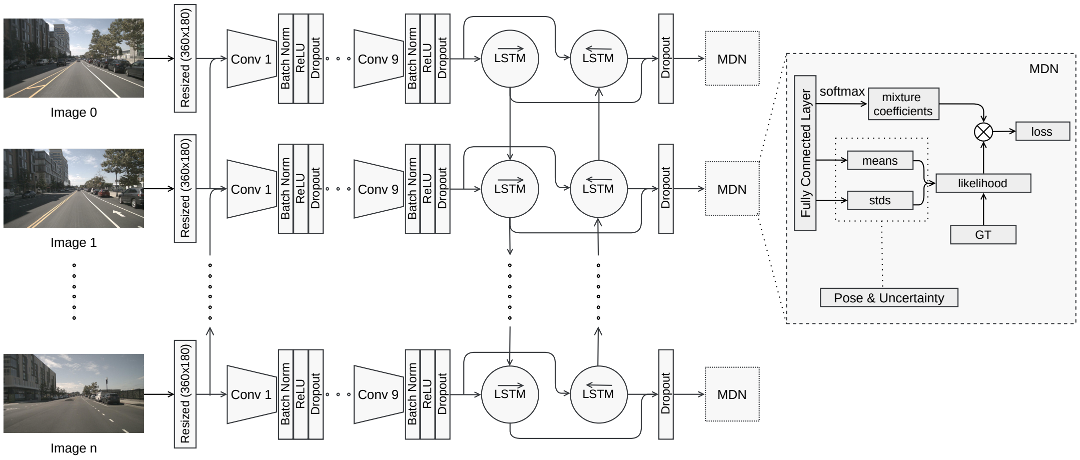


Fig. 2: Architecture of the proposed VO and uncertainty estimation framework, consisting of visual feature extractor, temporal modelling and mixture model based pose and uncertainty regression.

B. Temporal Modelling

We track the changes of distinguishable patterns on the image plane to estimate the camera motion [26]. Unlike commonly used hand-crafted features [27], [10], [17], [1], [23], in this work we utilise learnt image representations, and model their changes over time using RNNs.

Given image features, $\{f_1 ..., f_T\}$, RNN produces outputs r_t, for each frame I_t as:

$$r_t, h_t = \mathrm{RNN}(f_t, h_{t-1}) \tag{3}$$

where h_t is the hidden states of the RNN units after producing r_t, and h_0 is a zero vector.

We employ Long Short-Term Memory (LSTM) units [11] as our RNN structure, which has been shown to be successful in modelling long-term dependencies. We utilise a bi-directional LSTM, to enhance the representation by using a small temporal window of past frames.

The LSTM outputs, $\{r_1, ..., r_T\}$, are then passed to the next module, which estimates the mixture of relative pose distributions, conditioned on these features.

C. Mixture Density Estimation

Unlike other deep learning based VO methods [19], [31], which directly regresses a 6-DOF pose given an image sequence, our approach estimates a mixture model over the 6-DoF pose by using an MDN.

The MDN module takes temporally modelled features $\{r_1, ..., r_T\}$ and outputs the parameters of a mixture density model with M components for each time step t, namely means $\{\mu_1, ..., \mu_M\}^t$, standard deviations $\{\sigma_1, ..., \sigma_M\}^t$ and mixture coefficients $\{\alpha_1, ..., \alpha_M\}^t$. We use a fully connected layer to predict these parameters from LSTM outputs.

Given a set of temporally modelled features, $\{r_1, ..., r_T\}$, we model the conditional probability of pose change, $y_{(t-1,t)}$, at time t as:

$$p\left(y_{(t-1,t)}|r_t\right) = \sum_{i=1}^{M} \alpha_i\left(r_t\right) \phi\left(y_{(t-1,t)}|r_t\right) \tag{4}$$

where M is the number of mixture units, $\alpha_i\left(r_t\right)$ is the mixture coefficient which represent the probability of the pose, $y_{(t-1,t)}$, being in the i^{th} component given r_t. The conditional density, $\phi\left(y_{(t-1,t)}|r_t\right)$, of the pose $y_{(t-1,t)}$, for the i^{th} component, can be expressed as a Gaussian distribution:

$$\phi\left(y_{(t-1,t)}|r_t\right) = \frac{1}{\sigma_i\left(r_t\right)\sqrt{2\pi}} e^{\left\{-\frac{\|y_{(t-1,t)}-\mu_i(r_t)\|^2}{2\sigma_i(r_t)}\right\}} \tag{5}$$

where $\mu_i\left(r_t\right)$ and $\sigma_i\left(r_t\right)$ show the mean and standard deviation of the i^{th} mixture.

We can calculate the likelihood of a pose estimate, y_t, with K variables, i.e. $K = 6$ for 6 degree of freedom pose, given r_t, as the product of the likelihood of each variable as:

$$\mathcal{L}_t = \prod_{k=1}^{K} p\left(y_{(t-1,t)}[k]|r_t\right) \tag{6}$$

where $[k]$ is an indexing operation to access k^{th} variable of the pose.

The average likelihood for a sequence with T time steps, $\mathcal{L}$, is then calculated by summation over the time axis t as:

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}_t \tag{7}$$

We train our network to maximise the likelihood. Hence, we use the negative log likelihood as our error signal:

$$E = -\log(\mathcal{L}) \tag{8}$$

We use ground truth relative poses, $Y = \{y_{(0,1)}, ..., y_{(T-1,T)}\}$, to train our network. During inference, we derive the relative poses and uncertainties using the mean and standard deviations of the mixture components, respectively. We recover absolute pose, $\mathcal{Y}_T$, by accumulating the relative poses over time as:

$$\mathcal{Y}_T = \mathbf{y}_{(T-1,T)} \cdots \mathbf{y}_{(1,2)} \mathbf{y}_{(0,1)} \tag{9}$$

where $\mathbf{y}_{(t-1,t)}$ is the $4x4$ homogeneous transformation matrix representation of pose change $y_{(t-1,t)}$.

IV. Experiments

We evaluate our approach on the KITTI [8] and nuScenes [3] datasets and report quantitative and qualitative experiment results.

KITTI is one of the most popular autonomous driving datasets. It provides rectified camera images and ground truth motion of the vehicle. The driving scenarios include fast motions and sharp turns, presenting a challenging benchmark for egomotion estimation.

nuScenes is a recently released large-scale autonomous driving dataset which contains significant environmental variation. Compared to the KITTI dataset, nuScenes contains more varied sequences, including night time driving and rainy weather. We believe these challenging driving scenarios and environmental conditions present a good baseline for evaluating the necessity of VO uncertainty estimation.

A. *Implementation Details*

Our network is implemented using the PyTorch framework and trained on an NVIDIA TITAN X GPU. The training of the network takes approximately 40 epochs to converge. We used the Adam optimiser with parameters $(\beta_1 = 0.9, \beta_2 = 0.999)$. We utilise a plateau learning rate scheduler with a starting learning rate of 10^{-3}, patience of 8 and decay factor of 0.7. We also use dropout with a rate of 0.1 on CNN layers and a 0.2 drop rate on LSTM layers to prevent over-fitting. We use pre-trained Flownet weights [6] to initialise CNN backbone. For the rest of the parameters we used Xavier initialisation. We use the *evo* python package [9] to measure the performance of our approach, using relative pose error (RPE).

B. *Experiments on the KITTI Dataset*

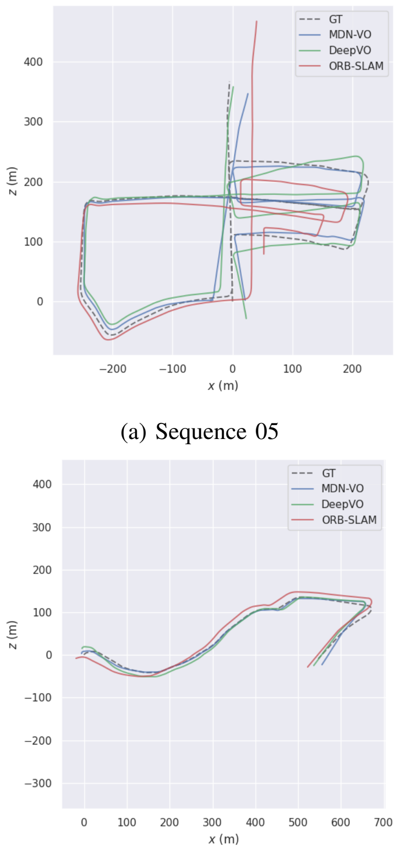
In our first set of experiments, we compared the proposed approach against the state-of-the-art on the KITTI dataset. We only considered the left colour camera as input to our network. For training, we used sequences 00, 01, 02, 04, 08, 09 and tested our model using sequences 03, 05, 06, 07, 10.

We compare our approach with monocular ORB-SLAM [20] and DeepVO [31]. We ran ORB-SLAM without loop closure, to make it more comparable to the proposed approach in the context of VO. Unlike the proposed approach, monocular ORB-SLAM estimates the trajectory up to a scale thus we aligned its trajectory with ground truth using [29]. We used the PyTorch implementation of DeepVO[1].

As can be seen in Table I, the proposed approach (MDN-VO) outperforms both DeepVO and ORB-SLAM approaches in overall mean RMSE score. ORB-SLAM mostly suffers on manoeuvres that include sharp turns, as the algorithm relies on maintaining sufficient 3D-to-2D feature matches to obtain scaled poses between consecutive frames.

Overall, DeepVO performs better than ORB-SLAM but the proposed approach (MDN-VO) yields the best results. We believe this is due to modelling the target pose changes with a mixture model. A network trained by least squares approximates the conditional averages of the target data. However, simply modelling the mean is insufficient for multimodal distributions [2]. Thus, using a mixture model enables our model to learn multiple modes from the training data, overcoming the limitations of direct regression.

We also share aligned trajectories for the test sequences 5 and 10 in Figure 3. As can be seen, ORB-SLAM performs drastically worse than learning based approaches in sequence 5. We believe this is caused by the inconsistencies in estimated local scales due to the lack of global loop closure. As the proposed approach learns to estimate scale during training as a prior, we do not suffer from this issue.



(a) Sequence 05

(b) Sequence 10

Fig. 3: Estimated Trajectories on the KITTI Dataset.

C. *Experiments on the nuScenes Dataset*

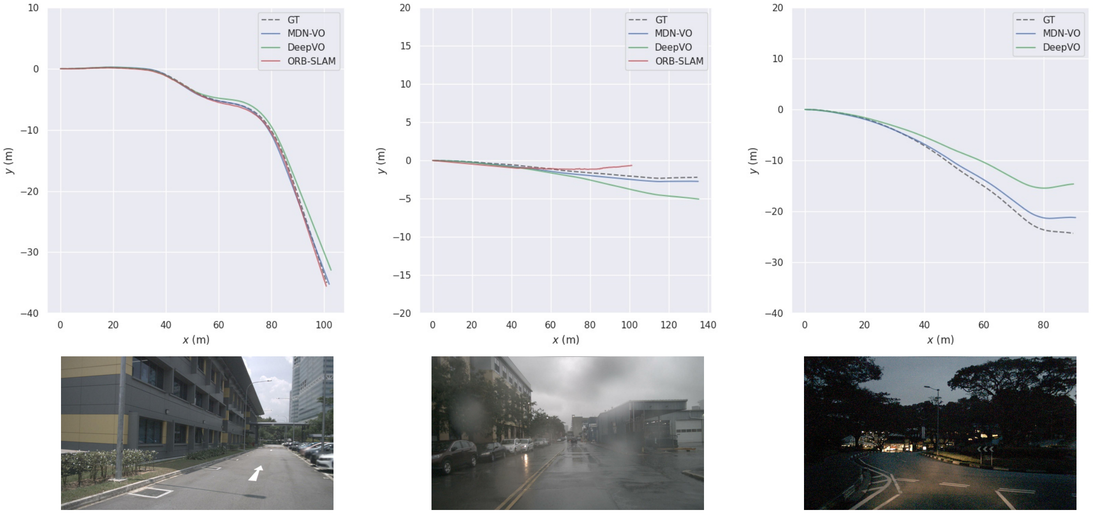
Next, we used the nuScenes dataset to evaluate our approach and examine its effectiveness in difficult conditions. As the nuScenes dataset does not have a defined VO benchmark protocol, we define our own data split which includes scenarios from the challenging environmental conditions, such as direct sunlight, lack of illumination etc, in both training and validation sets.

In Figure 4, we share the estimated trajectories in three different weather/lighting conditions, namely daylight, rain and

[1] https://github.com/ChiWeiHsiao/DeepVO-pytorch/

TABLE I: Quantitative Results on the KITTI Dataset

Sequence	ORB-SLAM			DeepVO			**MDN-VO (ours)**		
	RMSE	Max	Mean ± std	RMSE	Max	Mean ± std	RMSE	Max	Mean ± std
03	**0.03**	**0.20**	**0.03 ± 0.02**	0.08	0.23	0.08 ± 0.04	0.12	0.41	0.11 ± 0.06
05	0.25	0.67	0.20 ± 0.15	0.24	0.57	0.21 ± 0.12	**0.16**	**0.36**	**0.15 ± 0.08**
06	0.34	0.65	0.30 ± 0.17	**0.16**	**0.31**	**0.14 ± 0.08**	0.20	0.45	0.18 ± 0.09
07	0.17	**0.35**	0.13 ± 0.09	0.14	**0.35**	0.12 ± 0.07	**0.08**	0.51	**0.07 ± 0.05**
10	0.30	0.95	0.23 ± 0.20	0.21	0.47	0.19 ± 0.08	**0.14**	**0.32**	**0.13 ± 0.06**
mean	0.22	0.56	0.18 ± 0.13	0.17	**0.39**	0.15 ± 0.08	**0.14**	0.41	**0.13 ± 0.07**



(a) Daylight / scene-0972

(b) Rain / scene-0572

(c) Night / scene-0999

Fig. 4: Estimated trajectories (top) and a sample frame (bottom) from the corresponding nuScenes sequences.

TABLE II: Quantitative Results on the nuScenes Test Sequences

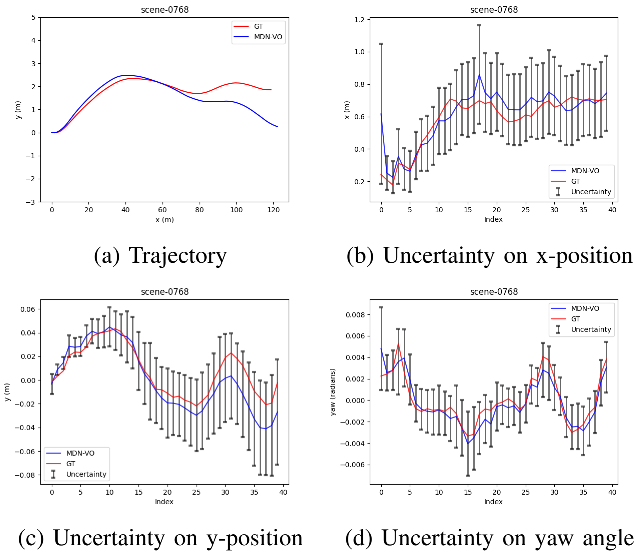
Condition	ORB-SLAM			DeepVO			MDN-VO (ours)		
	RMSE	Max	Mean ± std	RMSE	Max	Mean ± std	RMSE	Max	Mean ± std
Daylight	0.40	3.28	0.12 ± 0.38	0.09	0.54	0.06 ± 0.07	**0.07**	**0.53**	**0.04 ± 0.05**
Rain	0.76	10.19	0.14 ± 0.74	0.07	**0.39**	0.05 ± 0.05	**0.06**	0.43	**0.04 ± 0.05**
Night	-	-	-	0.12	0.74	0.09 ± 0.09	**0.11**	**0.73**	**0.07 ± 0.08**

night time. The estimated trajectories and the ground truth can be seen in the top row, while we display sample frames from the corresponding sequences in the bottom row. As can be seen, the proposed method (MDN-VO) successfully produces trajectories for all sequences. However, ORB-SLAM failed to initialise in the night time scenarios, hence we could not report ORB-SLAM's qualitative (See Figure 4c) and quantitative (See Table II) results for these scenarios. Compared to ORB-SLAM and DeepVO, our method (MDN-VO) achieves more accurate trajectory estimates, even in challenging conditions such as rain and night scenes. Qualitative results are reflected in our quantitative experiments, where we use the Relative Pose Error (RPE) as our error metric and report performance of each method with respect to different weather conditions. As can be seen in Table II, the proposed approach surpasses the performance of both ORB-SLAM and DeepVO models.

We investigate uncertainty estimation of the proposed method in detail on a sample sequence, namely scene-0768. We plot the estimated trajectory against ground truth in Figure 5a. As can be seen, the error between the estimated trajectory and ground truth increases towards the end of the trajectory. To investigate if this result is captured by the uncertainty estimation, we visualise the relative pose estimations and the ground truth between consecutive frames along with the corresponding uncertainty estimations. We plot x, y and yaw angles separately in Figure 5b, Figure 5c, Figure 5d, respectively, to give the reader more insight. As can be seen, the error between ground truth and our model's pose estimations are bounded by the uncertainty intervals (3σ). While the error in y estimate increases, the uncertainty estimate also increases. This shows that the failure on the y estimate is captured by the model and is reflected to the y uncertainty estimates (See Figure 5c). These results show that failure cases can be captured by the proposed model's uncertainty estimation.

V. Conclusions and Future Work

This paper presented an end-to-end deep learning based VO and uncertainty estimation method. We utilised a CNN-RNN hybrid architecture combined with an MDN. We



(a) Trajectory

(b) Uncertainty on x-position

(c) Uncertainty on y-position

(d) Uncertainty on yaw angle

Fig. 5: Uncertainty estimation on a test sample from nuScenes dataset (scene-0768).

evaluated our approach on two public autonomous driving datasets, namely KITTI and nuScenes. Our experiments demonstrate that our method can successfully estimate VO and uncertainty. Furthermore, our model achieves promising results under challenging conditions, such as rain and night scenes, where traditional VO approaches would fail. As future work, we plan to extend our work to utilise multiple cameras and fuse their estimations based on the estimated uncertainties.

References

[1] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, 2006.

[2] Christopher M Bishop. Mixture density networks. 1994.

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[4] Changhao Chen, Bing Wang, Chris Xiaoxuan Lu, Niki Trigoni, and Andrew Markham. A survey on deep learning for localization and mapping: Towards the age of spatial machine intelligence. *arXiv:2006.12567*, 2020.

[5] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPRW*, 2018.

[6] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015.

[7] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3), 2017.

[8] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11), 2013.

[9] Michael Grupp. evo: Python package for the evaluation of odometry and slam. `https://github.com/MichaelGrupp/evo`, 2017.

[10] Christopher G Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15. Citeseer, 1988.

[11] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8), 1997.

[12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[13] Alex Kendall and Roberto Cipolla. Modelling uncertainty in deep learning for camera relocalization. In *ICRA*, 2016.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[15] Ruihao Li, Sen Wang, Zhiqiang Long, and Dongbing Gu. Undeepvo: Monocular visual odometry through unsupervised deep learning. In *ICRA*, 2018.

[16] Shunkai Li, Xin Wang, Yingdian Cao, Fei Xue, Zike Yan, and Hongbin Zha. Self-supervised deep visual odometry with online adaptation. In *CVPR*, 2020.

[17] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 2004.

[18] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, volume 30, 2013.

[19] Vikram Mohanty, Shubh Agrawal, Shaswat Datta, Arna Ghosh, Vishnu Dutt Sharma, and Debashish Chakravarty. Deepvo: A deep learning approach for monocular visual odometry. *arXiv:1611.06069*, 2016.

[20] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5), 2015.

[21] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. Dtam: Dense tracking and mapping in real-time. In *ICCV*, 2011.

[22] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *CVPR*, 2004.

[23] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *ECCV*, 2006.

[24] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, 2011.

[25] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, 2020.

[26] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. *IEEE robotics & automation magazine*, 18(4), 2011.

[27] Jianbo Shi et al. Good features to track. In *CVPR*, 1994.

[28] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 2014.

[29] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Computer Architecture Letters*, 13(04), 1991.

[30] Ke Wang, Sai Ma, Junlan Chen, Fan Ren, and Jianbo Lu. Approaches challenges and applications for deep visual odometry toward to complicated and emerging areas. *IEEE Transactions on Cognitive and Developmental Systems*, 2020.

[31] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *ICRA*, 2017.

[32] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. *The International Journal of Robotics Research*, 37(4-5), 2018.

[33] Stephan Weiss, Markus W Achtelik, Simon Lynen, Michael C Achtelik, Laurent Kneip, Margarita Chli, and Roland Siegwart. Monocular vision for long-term micro aerial vehicle state estimation: A compendium. *Journal of Field Robotics*, 30(5), 2013.

[34] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *CVPR*, 2020.

[35] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *CVPR*, 2018.

[36] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *CVPR*, 2018.

[37] Cheng Zhao, Li Sun, Pulak Purkait, Tom Duckett, and Rustam Stolkin. Learning monocular visual odometry with dense 3d mapping from dense 3d flow. In *IROS*, 2018.

[38] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.

[39] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *ICRA*, 2017.