Learning monocular visual odometry with dense 3D mapping from dense 3D flow

Cheng Zhao[1], Li Sun[2], Pulak Purkait[3], Tom Duckett[2] and Rustam Stolkin[1]

***Abstract*— This paper introduces a fully deep learning approach to monocular SLAM, which can perform simultaneous localization using a neural network for learning visual odometry (L-VO) and dense 3D mapping. Dense 2D flow and a depth image are generated from monocular images by sub-networks, which are then used by a 3D flow associated layer in the L-VO network to generate dense 3D flow. Given this 3D flow, the dual-stream L-VO network can then predict the 6DOF relative pose and furthermore reconstruct the vehicle trajectory. In order to learn the correlation between motion directions, the Bivariate Gaussian modeling is employed in the loss function. The L-VO network achieves an overall performance of 2.68% for average translational error and $0.0143^{\circ}/m$ for average rotational error on the KITTI odometry benchmark. Moreover, the learned depth is leveraged to generate a dense 3D map. As a result, an entire visual SLAM system, that is, learning monocular odometry combined with dense 3D mapping, is achieved.**

I. Introduction

Simultaneous localization and mapping (SLAM) is an essential technique for mobile robot applications. In the past few decades, a substantial amount of research has been devoted to visual SLAM systems that enable robots to localize robustly and accurately in different environments. One of the most challenging branches of visual SLAM is monocular SLAM, which often suffers critically from absolute scale drift. Usually, some prior knowledge such as the height of the camera is necessary to alleviate scale drift. Moreover, these methods require hand-coded engineering efforts and excellent parameter tuning skills.

In recent years, deep learning techniques for visual odometry and SLAM have attracted considerable attention in the SLAM community. These methods not only provide good performance in challenging environments but also rectify the incorrect scale estimation of monocular SLAM. Supervised learning approaches formulate visual odometry (VO) as a regression problem. They explore the ability of CNN [1] or RNN [2][3] to learn ego-motion estimation using the change of RGB value features [4], deep flow [5] and non-deep flow [6] features. These methods are calibration-free but require a lot of expensive ground truth data for training.

On the other hand, some networks for predicting VO take advantage of geometric constraints, e.g. similarity constraints, epipolar constraints, etc., by integrating them into the loss function and training the network in an unsupervised manner. Although the trajectory ground truth is not required for these methods, consecutive frames [7][8][9] or stereo image pairs [10] along with the above geometric constraints are enough to provide sufficient supervision to train the network. However, these methods usually require the intrinsic parameters of the camera.

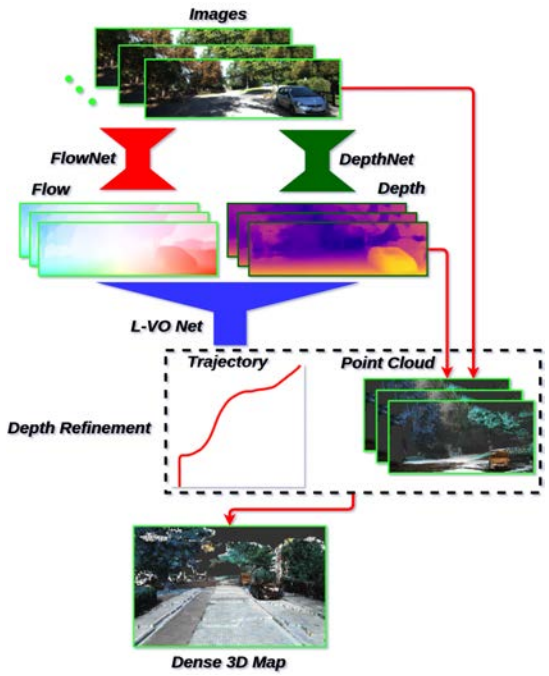


Fig. 1: The pipeline of the proposed learning monocular SLAM system. More detail can be found in Sec. III-A.

The main limitation of the above methods is that they all suffer from high dataset bias and require domain similarity between the training and testing sequences. Moreover, most of the deep learning geometry research only focus on visual odometry for localization without mapping. CNN-SLAM [11] is the forerunner to integrate learning of depth prediction with monocular SLAM to generate an accurate dense 3D map. But the odometry in CNN-SLAM is still based on the conventional method. Therefore, it is still not a pure deep learning SLAM method. In addition, some researches [12][13][14][15] integrate deep semantic information into a conventional SLAM system.

In this paper, a learning system for monocular SLAM is developed, which can simultaneously perform localization and dense 3D mapping through an end-to-end neural network. A learning visual odometry (L-VO) network with a 3D association layer is proposed for ego-motion estimation,

[1] Extreme Robotics Lab, University of Birmingham, Birmingham, UK, B15 2TT. IRobotCheng@gmail.com. [2] Lincoln Centre for Autonomous Systems (L-CAS), University of Lincoln, UK, LN6 7TS. [3] Cambridge Research Lab, Toshiba Research Europe, Cambridge, UK, CB4 0GZ.

which achieves an overall performance of 2.68% for average translational error and $0.0143°/m$ for average rotational error on the KITTI[1] odometry benchmark. The main contributions can be briefly summarized as follows: i) A new baseline L-VO method with a 3D association layer is proposed for ego-motion estimation, ii) a Bivariate Gaussian loss function is used to learn the correlation between motion directions, iii) L-VO is extended to a learning monocular SLAM system. An overview of the proposed architecture is shown in Fig. 1.

II. Related work

A. Learning based visual odometry (Pre-deep learning era)

In the recent past, some learning-based visual odometry estimation methods [16][17][18][19][20] were explored, before deep learning began to dominate many computer vision and robotics tasks. These learning-based methods mainly explored different pre-deep learning methods such as SVM, Gaussian Processes, etc. using sparse optical flow features for camera localisation and motion estimation.

B. Supervised deep learning for visual odometry

One of the pioneering works on deep learning for visual odometry estimation was proposed by Costante *et al.* [6]. They employed convolutional neural networks (CNNs) for ego-motion estimation from dense optical flow obtained by a non-deep method [21]. Then, Muller *et al.* [5] proposed Flowdometry, which combines FlowNet [22] and CNNs to obtain an end-to-end odometry system. Gabriele *et al.* [1] proposed Latent Space Visual Odometry (LS-VO) to find a non-linear representation of the optical flow manifold.

Tuomas *et al.* [4] explored LSTM for visual odometry. They utilized CNNs on the temporal change of RGB values (temporal derivatives) between two adjacent images. They utilized LSTM as a baseline in their work and proposed a back-propagation method for a Kalman Filter to learn the discriminative deterministic state estimators. Another seminal work on learning visual odometry was proposed by Wang *et al.* [2][23]. They utilized FlowNet features with LSTM for an end-to-end visual odometry system. Clark *et. al* [3] used the same network but fused the features of the monocular RGB camera with additional IMU readings for improved performance. Mehmet *et al.* [24] adopted a similar architecture – CNNs with LSTM – to develop a visual odometry system for endoscopic capsule robots.

C. Unsupervised deep learning for visual odometry

Most of the unsupervised visual odometry estimation methods predict the depth and ego-motion simultaneously. These methods do not require the trajectory ground truth but need camera parameters and often some additional information such as stereo images for training. Benjamin *et al.* [8] proposed the DeMoN architecture, which estimates not only depth and motion but also the surface normals and optical flow from a pair of images. They employed an unsupervised training loss function based on the relative spatial differences. Zhou *et al.* [7] also used a training loss function which minimizes image warping error of an image sequence for unsupervised depth prediction and ego-motion estimation. SfM-Net [9] predicts depth, segmentation, camera and rigid object motions, and transforms these to obtain frame-to-frame dense optical flow. Li *et al.* [10] combined the loss functions from [7] and [25] to obtain an unsupervised visual odometry method that can recover the absolute scale.

D. Learning visual SLAM

Most of the deep learning geometry research only focuses on VO for localization without mapping. The only forerunner of deep learning SLAM, CNN-SLAM [11], integrates CNN-style depth prediction with monocular SLAM to recover the absolute scale, and meanwhile generates a dense 3D map. However the odometry in CNN-SLAM is still based on the conventional method. As the estimated odometry of CNN-SLAM is not based on learning methods, it is not a complete end-to-end approach for learning SLAM.

E. Discussion

Conventional monocular visual odometry suffers from scale drift. Pioneering researchers [5][23][3][1] show that this problem can be mitigated via learning from 2D flow features. Inspired by RGBD-SLAM, the relative transform can be estimated directly from solving the PnP problem when the depth is given. In this paper, we model the visual odometry problem as a probabilistic regression problem. Multi-modal features, i.e. 3D flow (derived from the 2D flow and depth flow), are used to enhance the observation of the learning visual odometry. We further explore the correlation of motion directions and learn the translation with a multi-variate Gaussian rather than isotropic Gaussian [2]. Moreover, the learned depth is leveraged to generate a dense 3D map. As a result, an entire visual SLAM system, that is, learning monocular odometry combined with dense 3D mapping, is achieved.

III. Methodology

A. Overview

The pipeline of the proposed learning monocular SLAM is shown in Fig. 1. The proposed L-VO network is an end-to-end neural network for simultaneous monocular visual odometry and dense 3D mapping. To be more specific, L-VO Net takes a pair of consecutive images as input and predicts Ego-motion. The dense 2D flow and depth are obtained with FlowNet2 [26] and DepthNet [25] respectively. The estimated dense 2D flow and depth are further associated to obtain the 3D flow. Next, the 3D flow is fed into two separate regressors to predict the 6DOF relative pose (including scale) transform between each pair of images. As a consequence, the 6DOF camera trajectory can be obtained by accumulating relative poses. The point cloud is simultaneously generated and mapped incrementally from the given RGB image and the estimated depth. Furthermore, a 3D refinement is employed to remove the outliers and incorrect predictions. Finally, a dense 3D map is generated.

[1] http://www.cvlibs.net/datasets/kitti

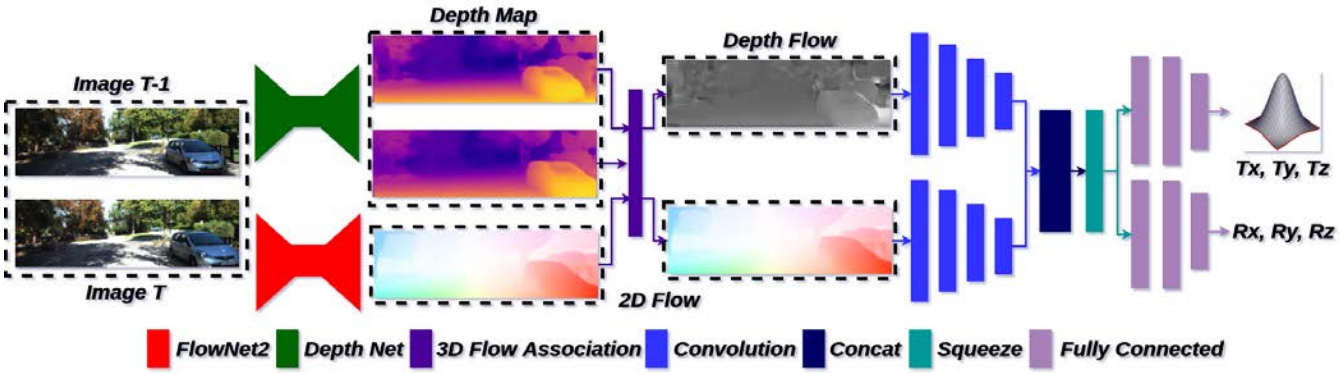


Fig. 2: The architecture of the proposed learning visual odometry (L-VO) network.

B. 2D optical flow and depth prediction

For dense 2D optical flow prediction, the state-of-the-art approach FlowNet2 [26] is employed. FlowNet2 is a stacked architecture composed of a series of FlowNet-S [22], FlowNet-C [22] and FlowNet-SD [26]. It can deliver robust 2D dense optical flow, which is of significant importance for learning odometry. We fine-tune this network using the training KITTI data (as described in IV-C) and then transplant the network for our task.

For depth prediction, any of the state-of-the-art methods [8], [7] and [25] can be adapted to our approach. In this paper, [25] is employed because of its good performance in outdoor scenes. [25] is an encoder-decoder architecture with appearance matching loss, disparity smoothness loss and left-right disparity consistency loss, which can be trained in unsupervised fashion. The training objective enables the network to perform the depth estimation from a monocular image. The network is also fine-tuned using the training KITTI data (as described in IV-C).

C. 3D flow association layer

We propose a 3D flow association layer which generates dense 3D flow from 2D flow and the corresponding depth maps. Assuming $F_{XY}^{k:k+1} \in \mathbb{R}^{h\times w\times 2}$ is the predicted dense 2D flow (on X-Y image plane) between frame k and $k+1$, and $D^k \in \mathbb{R}^{h\times w}$ is the predicted depth map of frame k, the 3D flow association layer can be defined as:

$$F_Z^{k:k+1}(x,y) = D^{k+1}\big((x,y) + F_{XY}^{k:k+1}(x,y)\big) - D^k(x,y) \tag{1}$$

$$F_{3D}^{k:k+1} = \mathcal{C}(F_{XY}^{k:k+1}, F_Z^{k:k+1}) \tag{2}$$

where $F_{3D}^{k:k+1}(x,y) \in \mathbb{R}^3$ refers to the 3D flow at pixel coordinate (x,y) and $\mathcal{C}$ is the concatenation operation. If the depth value in frame $k+1$ cannot be associated with the corresponding depth value in frame k, the missing flow pixels between two adjacent frames can be interpolated through bilinear filtering. It is worth noting that the inverse depth (i.e. disparity) is more sensitive to the motion of surroundings and objects close to the camera. Hence, the inverse depth is used instead of the depth value in our approach. We still use the term "depth" in order to make the following description more readable.

D. Learning odometry

As shown in Fig. 2, our learning odometry network is a dual stream architecture network, composed of two branches of convolution stacks followed by a squeeze layer [27] and two fully connected regressors. The convolution layers are composed of 3×3 filters and are of stride 2. The numbers of channels in the two branches are 64, 128, 256 and 512. In order to keep the spatial geometry information, the pooling layer is abandoned in these two CNN stacks. In the end, the feature maps of the two branches are concatenated together and squeezed using a 1×1 filter:

$$F_{3D} = \mathcal{S}(F_{XY}, F_Z) \tag{3}$$

where $\mathcal{S}$ is the squeeze operation, $F_{XY} \in \mathbb{R}^{h\times w\times n}$ and $F_Z \in \mathbb{R}^{h\times w\times n}$ are the feature maps of 2D flow and depth flow respectively, $F_{3D} \in \mathbb{R}^{h\times w\times n/4}$ is the squeezed feature, and n is the number of feature channels. The squeeze layer embeds the 3D feature map into a lower dimensional space, thereby reducing the input dimension of the regressors. A triple-layer fully-connected network is used for regression. We set the hidden layers of the regressors to size 128 with $relu$ activation function. The output of the translation regressor is 6 for bivariate Gaussian loss and that of the rotation regressor is 3, which is trained through a ℓ_2 loss. The details of the loss function are described as follows.

E. Bivariate Gaussian loss function

For most of the outdoor on-road driving data, e.g. KITTI dataset, there is a strong correlation between the translations along different axes in the horizontal plane. In contrast with the previous loss functions used in learning odometry, we aim to let our network learn the correlation along the forward and left/right translation directions. In this paper, this correlation is modeled as a multivariate Gaussian distribution.

The same camera configuration (axes definitions) as in the KITTI dataset is used, i.e. x : right (horizontal), y : down (vertical), z : forward (horizontal), then the translation variation along y coordinate is small compared to the other axes. Therefore, we only need to find the correlations between

translation x and translation z. In our approach, the Bivariate Gaussian Probabilistic-Density-Function (PDF) [28] is employed as the likelihood function for x (left/right) and z (forward) translation prediction. For the translation in y direction and orientations, ℓ_2 loss is used for optimization. Similar to [29], the Euler angles rather than quaternion are used to represent the orientation, as the quaternion representation opens up the possibility of over-fitting in the rotation regression. We further include a ℓ_2 regularization term for all weights to mitigate over-fitting. Our loss function is defined as:

$$\begin{aligned} loss = &\sum_{i}^{N} -\log\left(PDF((x_{gt}^i, z_{gt}^i),\ \mathcal{N}^i(\mu, \Sigma)\right) \\ &+\lambda_1 \sum_{i}^{N} \|y_p^i - y_{gt}^i\|_2 + \lambda_2 \sum_{i}^{N} \|r_p^i - r_{gt}^i\|_2 + \lambda_3 \|W\|_2 \end{aligned} \tag{4}$$

where N is the number of training pair images, $(x_{gt}^i, y_{gt}^i, z_{gt}^i)$ is the ground truth camera translation, and (x_p^i, y_p^i, z_p^i) is the predicted translation of the i^{th} image/camera. $r_p^i := (e_p^z, e_p^y, e_p^x)^i$ and $r_{gt}^i := (e_{gt}^z, e_{gt}^y, e_{gt}^x)^i$ are the predicted and ground-truth Euler angles, respectively. W are the trainable weights of the neural network. λ_1, λ_2 and λ_3 are the scale factors to balance the weights of translation and orientations. The Gaussian Density Function PDF is defined as:

$$\begin{aligned} &PDF\left((x_{gt}^i, z_{gt}^i),\ \mathcal{N}^i(\mu, \Sigma)\right) = \\ &\frac{exp(-\frac{1}{2}((x_{gt}^i, z_{gt}^i) - \mu)\Sigma^{-1}((x_{gt}^i, z_{gt}^i) - \mu)^T)}{((2\pi)^2|\Sigma|)^{-1/2}} \end{aligned} \tag{5}$$

where the bivariate Gaussian distribution $\mathcal{N}$ is:

$$\mu = (\mu_x, \mu_z)^i,\ \Sigma = \begin{pmatrix} \sigma_x^2 & \rho\sigma_x\sigma_z \\ \rho\sigma_x\sigma_z & \sigma_z^2, \end{pmatrix}^i. \tag{6}$$

where μ_x and μ_z are two mean variables in the left/right and forward direction, σ_x, σ_z are the corresponding standard deviations and ρ is the correlation coefficient of the translation between left/right and forward direction in the horizontal plane. Our neural network is expected to learn $(\mu_x, \mu_z, \sigma_x, \sigma_z, \rho, y_p)$, and (e_p^z, e_p^y, e_p^x), corresponding to the 6-dimensional and 3-dimensional outputs of two regression neural networks.

Once the network is trained, i.e. the translation $(\mu_x, \mu_z, \sigma_x, \sigma_z, \rho, y_p)$ and rotation (e_p^z, e_p^y, e_p^x) can be estimated from the network, the predicted translation in the horizontal plane is obtained through sampling within the bivariant Gaussian distribution using:

$$x, z = \frac{1}{N_s} \sum_{k}^{N_s} (x_s, z_s)^k \sim \mathcal{N}_p(\mu, \Sigma), \tag{7}$$

where $\mathcal{N}_p$ is obtained from $(\mu_x, \mu_z, \sigma_x, \sigma_z, \rho)$, $(x_s, z_s)^k$ is the kth sample, and N_s is the number of samples.

F. Octree depth fusion for mapping

We also proposed a dense 3D mapping method using the learned odometry and depth. Given the RGB image and the corresponding predicted depth image, the 3D point cloud (X, Y, Z) can be obtained through:

$$d_{u,v} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{8}$$

where f_x, f_y are the focal lengths, (c_x, c_y) is the principal point offset and s is the axis skew. (u, v) is the pixel position in the image plane.

Unfortunately, the depth prediction usually suffers from blur around the depth borders. The predicted depth is not accurate enough to be utilized directly for 3D mapping. In our approach, the OctoMap representation [30] is used to refine and maintain the 3D map. In order to build a robust, accurate dense 3D map, depth fusion using measurements from multiple views is employed. In OctoMap, each leaf node n stores the occupancy probability $P(n|o_{1:t})$. Given the 3D point measurements $o_{1:t}$, the probability $P(n|o_{1:t})$ can be updated as:

$$P(n|o_{1:t}) = \left[1 + \frac{1 - P(n|o_t)}{P(n|o_t)} \frac{1 - P(n|o_{1:t-1})}{P(n|o_{1:t-1})} \frac{P(n)}{1 - P(n)}\right]^{-1} \tag{9}$$

here, $P(n|o_t)$ can be obtained by a beam tracing sensor model. If the probability $P(n|o_{1:t})$ of the leaf node is beyond a threshold, this node will be marked as occupied in the dense 3D map. This probabilistic occupancy fusion can fuse the depth estimations from multiple views, and remove points arising from inaccurate depth predictions.

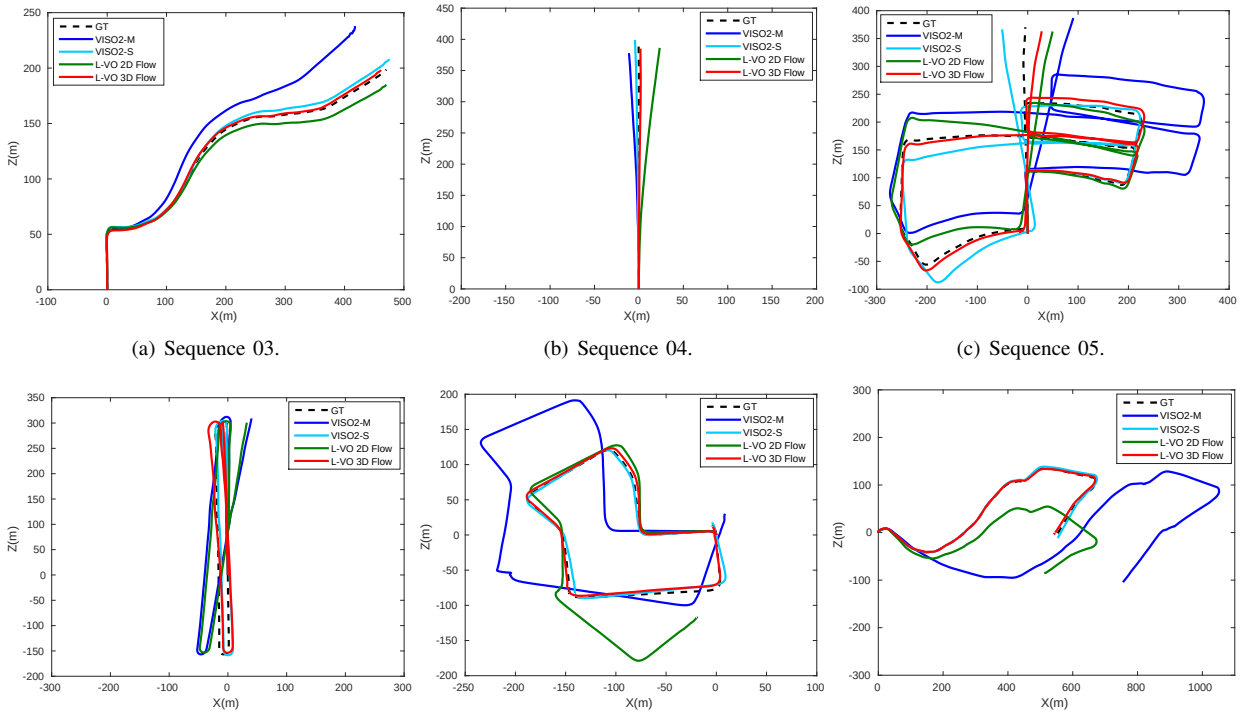
IV. Experiments

A. Dataset

The proposed L-VO Net is evaluated on the most popular KITTI VO/SLAM benchmark. The KITTI VO/SLAM benchmark consists of 22 sequences saved in PNG format. Sequences $00-10$ provide the sensor data with the accurate ground truth ($< 10cm$) from a GPS/IMU system, while sequences $11-21$ only provide the raw sensor data. The large number of dynamic objects such as cars means that visual odometry could easily fail on this challenging dataset.

B. Network training and testing

The network is trained with Adam optimization. The batch size is set to 100, the momentum is fixed to $(0.9, 0.999)$, and the starting learning rate is 0.0001. The step learning policy is adopted and the learning rate decay is set to 0.95. The network is trained by 100 epochs. In order to reduce the GPU memory requirement and training time, the raw images from the KITTI dataset are down-sampled 4 times to 320×96. But using this small image size for training can definitely degrade the performance. The whole network is end-to-end trainable. Considering the GPU limitation, training the network step-by-step is more practicable. The pre-trained model (without training on KITTI dataset) from [26] and [25] is adapted and then fine-tuned using the training KITTI data (as described in IV-C). In order to enhance the performance and avoid over-fitting, both geometric augmentation (translation, rotation, scaling) and image augmentation (color, brightness, gamma)



(a) Sequence 03. (b) Sequence 04. (c) Sequence 05.

(d) Sequence 06. (e) Sequence 07. (f) Sequence 10.

Fig. 3: The predicted trajectories of the proposed L-VO Net on Sequences 03, 04, 05, 06, 07 and 10. The network is trained on Sequence 00, 02, 08 and 09.

TABLE I: The comparison of the performance of L-VO against the baselines on the KITTI dataset according to the evaluation method [2]. Note that VISO-S is a stereo VO and the other methods are monocular VO. The L-VO model is trained on the sequences 00, 02, 08 and 09, and evaluated on the rest.

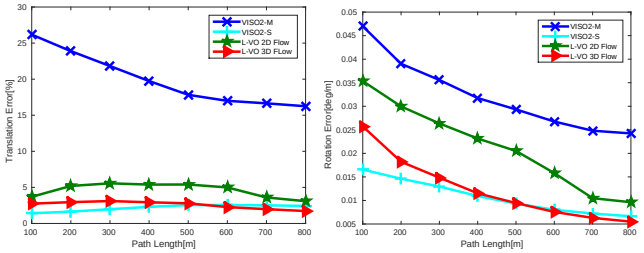
Seq.	VISO-S[31] (1242 × 376)		VISO-M[31] (1242 × 376)		ESP-VO[2] (1242 × 376)		L-VO(2D Flow) (320 × 92)		L-VO(3D Flow) (320 × 92)	
	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$	$t_{rel}(\%)$	$r_{rel}(^\circ)$
03	1.71	1.12	9.02	2.83	6.72	6.46	3.35	1.62	**3.18**	**1.31**
04	1.54	0.84	4.33	1.63	6.33	6.08	4.15	2.53	**2.04**	**0.81**
05	2.36	1.20	19.16	3.62	3.35	4.93	**2.49**	1.19	2.59	**0.99**
06	1.47	0.87	6.64	1.96	7.24	7.29	3.19	1.54	**1.39**	**0.95**
07	2.37	1.78	26.54	5.92	3.52	5.02	17.2	10.4	**2.81**	**2.54**
10	1.51	1.15	48.29	3.43	9.77	10.2	7.24	**3.06**	**4.38**	3.12
Mean	1.83	1.16	19.00	3.23	6.15	6.66	6.27	3.39	**2.73**	**1.62**

t_{rel} and r_{rel} are average translational RMSE(%) and rotational RMSE($^\circ$/100m) over $100m - 800m$ intervals.

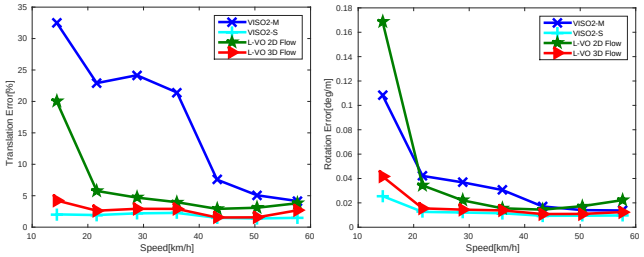
are employed. As mentioned in [22][25] and [2], we also observe that these data augmentation techniques are crucial to improve the 2D flow estimation, depth prediction, and especially VO prediction, because of the limited number of training examples. During testing, the number of Gaussian samples is set to 10000.

C. Visual odometry performance

We perform two kinds of evaluation for the proposed methods. The first evaluation is based on sequence $00 - 10$. Both the qualitative and quantitative results are reported for analysis. For fair comparison, we follow the same partition proposed by [23][2] and split the sequences 00-10 to 00, 02, 08, 09 for training and 03, 04, 05, 06, 07, 10 for testing. The second evaluation is based on sequence 00-21. The sequence 00-10 is employed for training and sequence 11-21 for testing. Only the qualitative results are provided because the ground truth of sequence 11-21 are not provided. The open-source visual odometry library VISO2 [31] is employed as the baseline method. It provides both monocular visual odometry and stereo odometry. For monocular VO, the fixed height (1.7) and pitch (-0.03) are employed in order to



(a) Translation vs path length. (b) Rotation vs path length.



(c) Translation vs speed. (d) Rotation vs speed.

Fig. 4: Average translational and rotational errors of the baselines against different path lengths and speeds. The L-VO model is trained on the sequences 00, 02, 08 and 09, and evaluated on the rest.

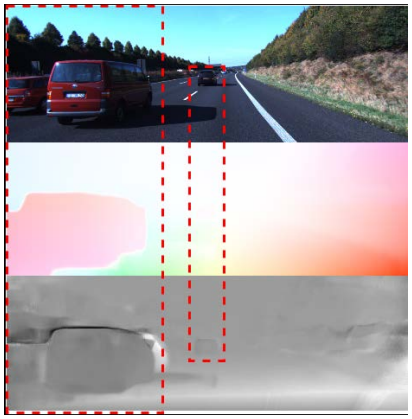


Fig. 5: Sample image, 2D flow and depth flow (from top to bottom) from sequence 21 of KITTI. The red box refers to fast moving objects. This sequence is a very long distance scenario with many dynamic objects.

recover the absolute scale.

We evaluate the learning odometry using the KITTI VO evaluation metrics, computing the average translational and rotational RMSE for all possible sub-sequences of length $(100, \ldots, 800)$ meters. Note that the same evaluation metric was employed in [2].

For the first evaluation, the overall performance of average translational and rotational errors of L-VO based on 2D flow and 3D flow can reach $4.71\%, 0.0241°/m$ and $2.68\%, 0.0143°/m$, respectively, using the standard KITTI evaluation metrics. The detailed comparison of performances (some entries are copied from [2]) is shown in Table I. It is clear that the performance of both L-VO (2D) and L-VO (3D) is much better than conventional monocular VO. L-VO (3D) performs slightly worse than conventional stereo VO. This can also be seen in the predicted trajectory Fig. 3. Most of the time, the average drift distances of the red line (L-VO 3D) and green one (L-VO 2D) are between that of the light blue line (stereo VO) and dark blue line (monocular VO). The red line is much closer to the light blue line.

The main limitation of monocular VO and SLAM is the absolute scale estimation. However, with a deep learning method, the scale can be estimated more accurately without any scene-based geometric constraints such as camera height. This is one of the main reasons why the proposed L-VO(2D) and L-VO(3D) outperform the conventional monocular VO.

As we formulate VO prediction as a regression problem, multi-modal features can enhance the prediction. That is the reason why the result of L-VO(3D) is better than L-VO(2D) and closer to the performance of stereo VO. Another reason why L-VO(3D) can be close to stereo VO is that the (x, z) constraint in the Bivariate Gaussian loss function can learn the translation correlation between the left/right and forward direction. This learned constraint can make the trajectory more accurate – see, for example, the straight line in Fig. 3(b) and corner in Fig. 3(c).

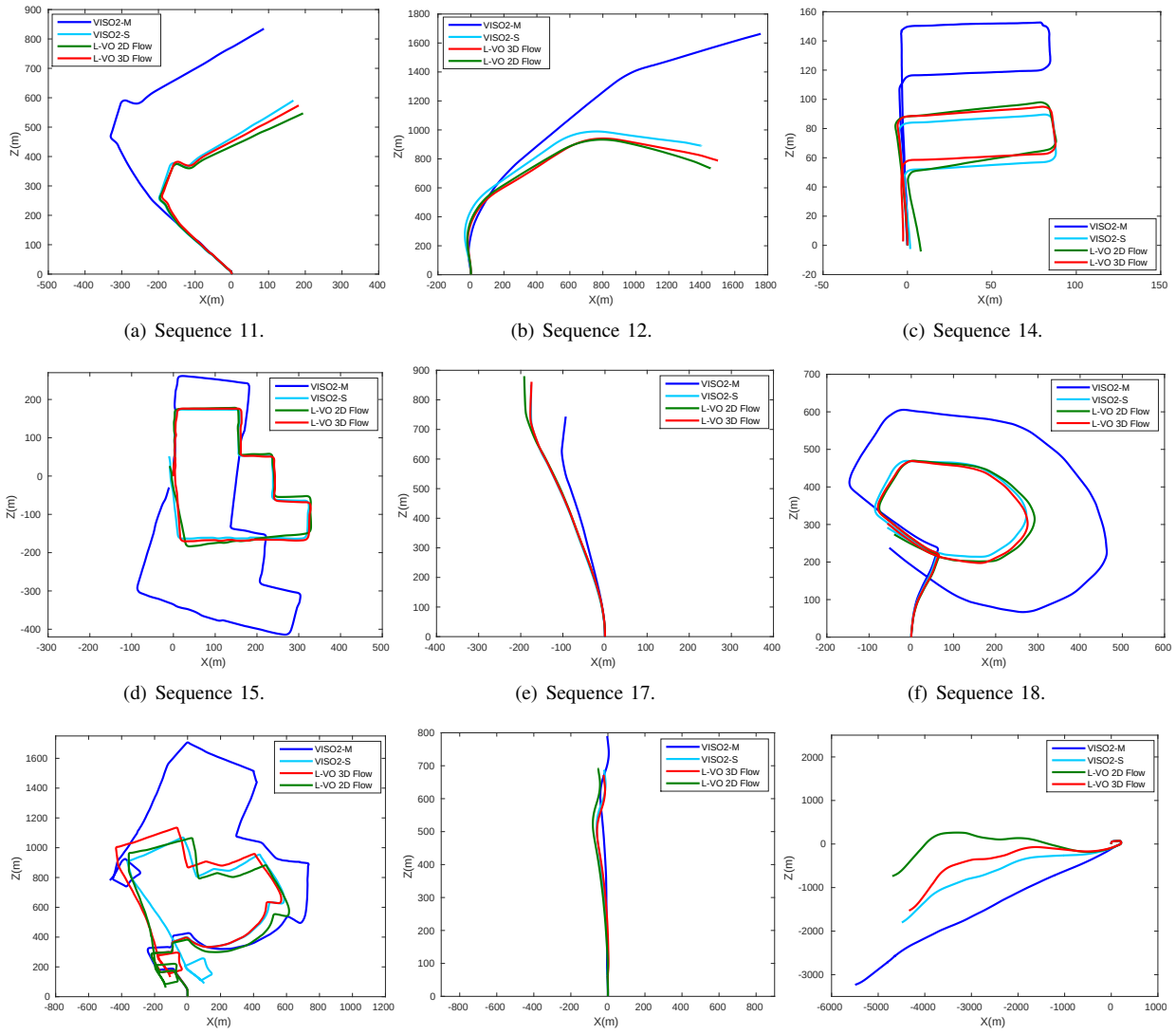
For low-speed scenarios, the magnitude of 2D flow is insignificant and thus provides a weak feature response to the network, while the magnitude of the depth flow is still quite strong even in a low-speed situation. Thus, the depth flow feature is a good complement to 2D flow in low-speed situations, which is further observed in Fig. 4(c) and Fig. 4(d). Moreover, because the training data is only provided by 4 sequences, multi-modal features, *i.e.*, 3D flow can enhance the robustness of 6DOF relative pose regression.

For the second evaluation, the L-VO network is trained using more data, i.e. sequence 00-10. Due to the lack of ground truth, only qualitative results are shown in Fig. 6. It can be seen that the L-VO network can also give a high-quality prediction in the new scenarios. Both L-VO(2D) and L-VO(3D) outperform monocular VO thanks to better scale estimation. The trajectory of L-VO(3D) is closer to stereo VO than L-VO(2D). However, the performance of L-VO(2D) is boosted more than L-VO(3D) by using more training data.

During testing, we observe that L-VO cannot give a similar prediction to stereo VO for sequence 21 (Fig. 6(i)). This sequence is very challenging as it is captured over a long distance in a high-speed scenario (up to 80km/h). The main difficulty L-VO encounters is the high number of moving objects such as fast-moving cars in this street. As displayed in Fig. 5, the main flow feature is extracted from the fast-moving cars. Therefore, the main challenge for flow-based learning VO is to remove the effects of dynamic objects.

D. Dense 3D mapping

A learning monocular SLAM system integrated with L-VO(3D) is deployed in this paper. The whole system is implemented under ROS and the neural network is implemented using Tensorflow trained on an NVIDIA Titan GPU. Compared to LSD and ORB monocular SLAM, our system



(a) Sequence 11. (b) Sequence 12. (c) Sequence 14.

(d) Sequence 15. (e) Sequence 17. (f) Sequence 18.

(g) Sequence 19. (h) Sequence 20. (i) Sequence 21.

Fig. 6: The predicted trajectories of proposed L-VO Net on Sequence 11, 12, 14, 15, 17, 18, 19, 20 and 21 (There are no ground truth available for these sequences). The L-VO model used is trained on Sequence from 00 to 10.

can generate a significantly denser 3D map. In order to alleviate the border blur and wrong prediction of depth, depth fusion from multiple frames is employed during mapping. In order to reduce the hardware resource requirement, OctoMap is used for the map representation instead of the point cloud. Given the dense refinement of depth information, a dense 3D map can be generated online. In Fig. 7, the center image is the dense 3D map of the sequence 07 in the KITTI dataset and the small images in the surrounding show enlarged local areas of the global map. It can be seen that after depth fusion, sharply defined shapes such as the car, trees and building are obtained. Moreover, a lot of outliers and noise are removed to make the map cleaner.

V. Conclusion

In this paper, a learning system for monocular SLAM is proposed, which can deploy simultaneous localization using a L-VO neural network and the dense 3D mapping. Its performance exceeds most of the monocular SLAM approaches and is even comparable with some stereo SLAM approaches. Compared with conventional SLAM, its main limitations are the high computational requirements and high dataset bias. A demo can be found on the first author's Youtube channel[2].

VI. Acknowledgement

The authors was funded by a DISTINCTIVE scholarship, EU H2020 projects: RoMaNS (645582) & ILIAD (732737).

References

[1] G. Costante and T. A. Ciarfuglia, "LS-VO: Learning dense optical subspace for robust visual odometry estimation," *arXiv preprint arXiv:1709.06019*, 2017.

[2] https://youtu.be/Ccj1O7yndIk

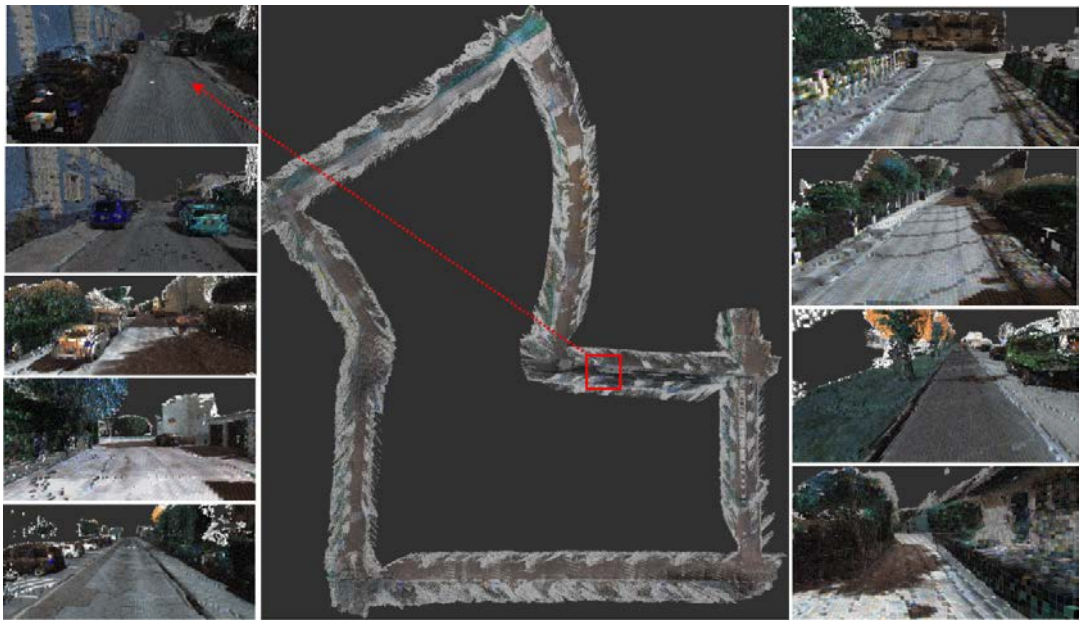


Fig. 7: The center image is the global dense 3D map of sequence 07 in the KITTI dataset. The small images in the surrounding show enlarged local areas of the global map.

[2] 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*, 2017.

[3] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, "VINet: Visual-inertial odometry as a sequence-to-sequence learning problem." in *AAAI*, 2017, pp. 3995–4001.

[4] T. Haarnoja, A. Ajay, S. Levine, and P. Abbeel, "Backprop KF: Learning discriminative deterministic state estimators," in *Advances in Neural Information Processing Systems*, 2016, pp. 4376–4384.

[5] P. Muller and A. Savakis, "Flowdometry: An optical flow and deep learning based approach to visual odometry," in *Proc. WACV*, 2017, pp. 624–631.

[6] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia, "Exploring representation learning with CNNs for frame-to-frame ego-motion estimation," *IEEE robotics and automation letters*, vol. 1, no. 1, pp. 18–25, 2016.

[7] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *CVPR*, vol. 2, no. 6, 2017, p. 7.

[8] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "DeMoN: Depth and motion network for learning monocular stereo," in *CVPR*, vol. 5, 2017.

[9] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "Sfm-Net: Learning of structure and motion from video," *arXiv preprint arXiv:1704.07804*, 2017.

[10] R. Li, S. Wang, Z. Long, and D. Gu, "UnDeepVO: Monocular visual odometry through unsupervised deep learning," *arXiv preprint arXiv:1709.06841*, 2017.

[11] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular slam with learned depth prediction," *arXiv preprint arXiv:1704.03489*, 2017.

[12] C. Zhao, L. Sun, and R. Stolkin, "A fully end-to-end deep learning approach for real-time simultaneous 3d reconstruction and material recognition," in *Advanced Robotics (ICAR), 2017 18th International Conference on*. IEEE, 2017, pp. 75–82.

[13] C. Zhao, L. Sun, B. Shuai, P. Purkait, and R. Stolkin, "Dense RGB-D semantic mapping with pixel-voxel neural network," *arXiv preprint arXiv:1710.00132*, 2017.

[14] L. Sun, C. Zhao, and R. Stolkin, "Weakly-supervised DCNN for RGB-D object recognition in real-world applications which lack large-scale annotated training data," *arXiv preprint arXiv:1703.06370*, 2017.

[15] L. Sun, Z. Yan, A. Zaganidis, C. Zhao, and T. Duckett, "Recurrent-OctoMap: Learning state-based map refinement for long-term semantic mapping with 3d-lidar data," *arXiv preprint arXiv:1807.00925*, 2018.

[16] R. Roberts, H. Nguyen, N. Krishnamurthi, and T. Balch, "Memory-based learning for visual odometry," in *ICRA*. IEEE, 2008, pp. 47–52.

[17] R. Roberts, C. Potthast, and F. Dellaert, "Learning general optical flow subspaces for egomotion estimation and detection of motion anomalies," in *CVPR*. IEEE, 2009, pp. 57–64.

[18] V. Guizilini and F. Ramos, "Visual odometry learning for unmanned aerial vehicles," in *ICRA*. IEEE, 2011, pp. 6213–6220.

[19] Guizilini and F. Ramos, "Semi-parametric models for visual odometry," in *ICRA*. IEEE, 2012, pp. 3482–3489.

[20] T. A. Ciarfuglia, G. Costante, P. Valigi, and E. Ricci, "Evaluation of non-geometric methods for visual odometry," *Robotics and Autonomous Systems*, vol. 62, no. 12, pp. 1717–1730, 2014.

[21] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *ECCV*. Springer, 2004, pp. 25–36.

[22] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *ICCV*, 2015, pp. 2758–2766.

[23] S. Wang, R. Clark, H. Wen, and N. Trigoni, "DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *ICRA*. IEEE, 2017, pp. 2043–2050.

[24] M. Turan, Y. Almalioglu, H. Araujo, E. Konukoglu, and M. Sitti, "Deep EndoVO: A recurrent convolutional neural network (RCNN) based visual odometry approach for endoscopic capsule robots," *Neurocomputing*, vol. 275, pp. 1861–1870, 2018.

[25] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *CVPR*, 2017.

[26] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *CVPR*, vol. 2, 2017.

[27] F. N. Iandola and others, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[28] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[29] A. Elgammal and C.-S. Lee, "Inferring 3d body pose from silhouettes using activity manifold learning," in *CVPR*, 2004.

[30] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[31] A. Geiger, J. Ziegler, and C. Stiller, "StereoScan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 963–968.