2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

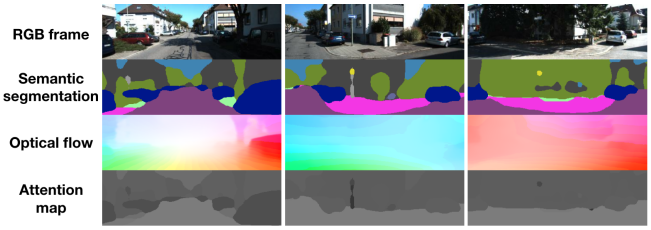
# Dynamic Attention-based Visual Odometry

Xin-Yu Kuo, Chien Liu, Kai-Chen Lin, Evan Luo, Yu-Wen Chen, and Chun-Yi Lee
Elsa Lab, Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
{bassykuo,liu_chien,kcheng1008105,luoevan6665,carrie,cylee}@gapp.nthu.edu.tw

***Abstract*— This paper proposes a dynamic attention-based visual odometry framework (DAVO), a learning-based VO method, for estimating the ego-motion of a monocular camera. DAVO dynamically adjusts the attention weights on different semantic categories for different motion scenarios based on optical flow maps. These weighted semantic categories can then be used to generate attention maps that highlight the relative importance of different semantic regions in input frames for pose estimation. In order to examine the proposed DAVO, we perform a number of experiments on the KITTI Visual Odometry and SLAM benchmark suite to quantitatively and qualitatively inspect the impacts of the dynamically adjusted weights on the accuracy of the evaluated trajectories. Moreover, we design a set of ablation analyses to justify each of our design choices, and validate the effectiveness as well as the advantages of DAVO. Our experiments on the KITTI dataset shows that the proposed DAVO framework does provide satisfactory performance in ego-motion estimation, and is able deliver competitive performance when compared to the contemporary VO methods.**

## I. Introduction

Learning based visual odometry (VO) has been a crucial research domain [1-14] in the past few years. The objective of it is to derive the ego-motion of a camera using learning based approaches such as deep convolutional neural networks (DCNNs). The techniques based on a single monocular camera is especially of interest to recent researches due to its wide availability and low cost. Conventional learning based VO works typically exploit the entire RGB input frames for determining the trajectory of the camera [15-17], where some of them may take additional inputs such as optical flow maps (or simply 'flow maps') [4], [10], [11], semantic segmentation [18-20], depth maps [1], [3], [21-25], or a fusion of them [20], [26]). These approaches usually treat every single frame the same way. Nevertheless, each semantic category in a frame may contribute different extents of information when they are used for estimating the trajectory of the camera in different motion scenarios (e.g., straight moves, making turns, etc.). For example, cars or pedestrians are usually considered as dynamic objects that may harm the performance of ego-motion estimation. This motivates recent researchers to propose techniques to deal with dynamic objects by directly removing them from input frames [27], [28] before estimating the trajectories of the camera. However, in some motion scenarios, objects belonging to these semantic categories are static, and can thus be reasonably used as references for performing ego-motion estimation. Simply eliminating certain semantic categories by heuristics or attention weights based on human priors in all scenarios may limit the performance of VO models. Moreover, attention is not required to be a binary decision limited to certain semantic categories. Motivated by the above insights, in this paper we assume that in different motion scenarios, dynamically adjusting the attention weights on different semantic categories for input frames might be beneficial for deriving the pose (and thus the ego-motion) of the camera.



(a) Moving straight. (b) Turning right. (c) Turning left.
Fig. 1: Changes in attention maps for three motion scenarios.

In order to validate the assumption, in this work we propose to employ flow maps for dynamically adjusting the attention weights of different semantic categories in different camera motion scenarios. Optical flow is an excellent type of feature representation derived from consecutive image frames that incorporates the information about the displacement of pixels [29]. As flow maps contain rich information relevant to the motion of the camera [4], [10-12] as well as those of the perceived semantic contents, it is reasonable to distinguish different motion scenarios by leveraging on them. In order to achieve this objective, in this work we propose to generate the attention weights of different semantic categories dynamically according to the changes in flow maps. These weighted semantic categories can then be used to generate the attention maps that highlight the relative importance of different regions in input frames for pose estimation. Fig. 1 shows an example for demonstrating the changes in attention maps of our work for three scenarios: (a) moving straight, (b) turning right, and (c) turning left. The four rows correspond to the RGB frames, their semantic segmentations, the flow maps generated from these frames and their next frames, and the derived attention maps. The brighter parts of the attention maps correspond to the regions with higher attention weights. It is observed that for scenario (a), the attention map focus more on the road and the sky regions while suppressing the regions of cars. For scenario (b) and (c), on the other hand, the regions of cars are less suppressed, while the sky region is less focused. These examples illustrate that the relative importance of the semantic regions in the attention maps may vary when estimating the ego motion of the camera. By leveraging flow maps, the concept discussed above enables derivation of the attention maps without human supervision.

978-1-7281-6211-9/20/$31.00 ©2020 IEEE

We additionally offer an ablation analysis of the sources for generating the attention maps in Section IV to justify our design decision.

As a result, in this paper we propose *DAVO*, a **D**ynamic **A**ttention-based **V**isual **O**dometry framework for estimating ego-motion of a monocular camera. DAVO is a learning based framework based on DCNNs, without using depth maps or recurrent memory cells [12], [14], [30]. Different from similar previous learning based works (which are described in Section II), DAVO feeds consecutive RGB input frames and flow maps adjusted with attention maps to its pose estimation DCNN. Each attention map is generated by an Attention Module revised from a squeeze-and-excitation network (SENet) [31], and is implemented as the weighted sum of the semantic segmentation channels, as depicted in Fig. 2 and later explained in Section III. The weights are dynamically adjusted according to the flow maps of consecutive RGB input frames, which is generated by FlowNet 2.0 [29] pre-trained on the Flying Chairs dataset [32]. These dynamically adjusted weights allow DAVO to alter its attention maps for different camera motion scenarios. To examine the advantage of DAVO, we perform experiments on the KITTI datasets [33], and compare the quantitative results on the evaluation trajectories with a set of baseline methods. We further illustrate the trajectories evaluated by DAVO as well as the baseline approaches to compare their differences to the reference ones. In order to validate the effectiveness of the proposed framework and justify our design choices, we perform a set of ablation analysis for the following cases: (1) with and without the Attention Module, (2) dynamic and static attention weights, (3) DAVO and feature-based attention design, and (4) different sources for generating the dynamic attention weights. The primary contributions of this paper is summarized as follows.

- A learning based DAVO framework that feeds RGB input frames and flow maps both weighted by the generated attention maps to the pose estimation DCNN.
- A concept of using flow maps for generating dynamic attention weights for semantic segmentation channels.
- An approach that enables derivation of attention weights without human supervision.

The rest of the paper is organized as follows. Section II briefly reviews the related works. Section III describes the proposed DAVO framework, its components, and the training cost functions. Section IV presents the experimental results as well as a set of ablation analyses. Section V concludes.

## II. Related Works

A number of learning-based monocular VO [8], [17], [34] works that embrace DCNNs have been proposed in the past few years. These monocular VO works exploit the advantages of DCNNs to enhance the performance of their ego-motion estimation accuracy as well as increase the robustness against noisy features perceived in real environments. We summarize the related works into two categories: (1) flow-based approaches, and (2) attention-based approaches. We do not consider the works that exploit depth maps as they are not relevant to the scope of this paper.

### A. Flow-based Approaches

A significant portion of research works [4], [10], [35] have introduced optical flow estimation into their VO models in recent years. Instead of directly feeding consecutive raw RGB frames into the VO models, flow maps can be used as inputs for the VO models, as displacements of pixels (and hence, the movements of the objects) between consecutive image frames can be better employed by these models in the process of ego-motion estimation. The authors in [4], [10] introduce the famous FlowNet [36] in their VO module, while the authors in [4] additionally introduce an auto-encoder (AE) in their network architecture to enhance the flow representation. Architectures based on [36] that employ recurrent memory cells to learn sequential dependencies and complex motion dynamics of an image sequence have also been investigated in [12], [13], [37]. Similar to [12], [13], a recent work that incorporates a cascade of multiple flow networks [29] followed by a number of recurrent neural network (RNN) cells and fully-connected (FC) layers are discussed in [11]. These works differ from DAVO in that their flow maps are directly fed into the pose estimation DCNNs, without considering any attention mechanisms.

### B. Attention-based Approaches

A few recent researchers attempt to introduce attention techniques to enhance the accuracy of their pose estimation DCNNs [14], [27], [28], [30], [38]. These techniques fall into two categories: (1) heuristic-based attention methods, and (2) feature-based attention methods. In the first category, attention masks to the input frames are defined based on human knowledge or heuristic experience. In Mask-SLAM [39], semantic masks are manually selected (e.g., sky, car, etc.) to filter out feature points extracted from input frames. On the other hand, in [27], [28], pre-defined semantic regions (e.g., walking people, moving vehicles, etc.) are directly removed from input frames before performing pose estimation. In [27], the authors further propose methods for enhancing moving objects detection according to the change in projection depth of same keypoints between two frames with respect to a custom defined threshold. These methods differ from ours in that DAVO does not require preliminary knowledge for determining the attention weights of different semantic categories. For feature-based attention methods, attention models are incorporated for adjusting the relative weights of feature channels in the pose estimation DCNNs. In [30], an attention model is employed to determine the global trajectory of the camera from meta features generated by a set of local pose estimators. In SRNN_channel [14], a guidance attention model is separately applied to the feature channels of the translation and rotation estimation networks. In [38], an attention mechanism is applied to both visual and inertial embedding before they are used for pose estimation. The above approaches differ from ours in that DAVO concentrates on generating attention maps for RGB input frames and

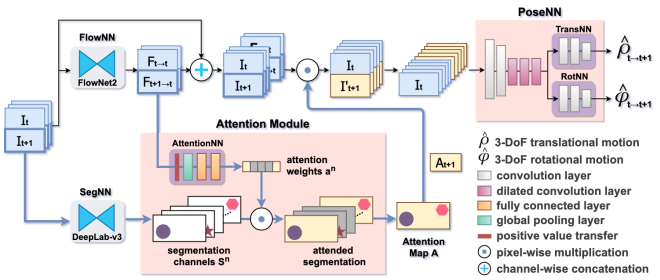


Fig. 2: Overview of the DAVO framework.

flow maps, rather than applying the attention model to the extracted feature embedding. A comparison of our method and SRNN_channel [14] is provided in Section IV. Please note that we do not compare ours with [30] and [38] because (1) the former does not release the source codes and does not benchmark their performance on the KITTI dataset [33], and (2) the latter requires additional inertial measurement units and is not an odometry method fully relies on visual inputs.

## III. Proposed Methodology

In this section, we introduce our DAVO framework. We first provide an overview of the framework and walk through the interplay of its components. We then dive into the implementation details of the two proposed modules in DAVO: the *Attention Module* and the *Dilated Pose Estimation Module* (abbreviated as *PoseNN* hereafter). The former is designed based on the idea that the attention weights should be adjusted dynamically, while the latter is developed with dilated convolutions [40] to enlarge the receptive fields of the feature maps for enhancing the pose estimation accuracy. Lastly, we explain the loss function used for training DAVO.

### A. Overview of the DAVO Framework

Fig. 2 illustrates an overview of our DAVO framework. The regions highlighted in red correspond to our Attention Module and PoseNN. The rest of the DCNNs, including the segmentation DCNN (*SegNN*) and the optical flow estimation DCNN (*FlowNN*), are commonly seen in contemporary VO models and are not within the scope of our contribution. In our DAVO framework, FlowNN and SegNN are implemented based on FlowNet 2.0 [29] and Deeplabv3+ [40] (using Xception65 [41] architecture as its model backbone), respectively.

For an RGB input frame $I$ received at time $t$ (denoted as $I_t$), FlowNN generates optical flow maps ($F_{t\to t}$, $F_{t+1\to t}$) by predicting the optical flow between consecutive input frames ($I_t \to I_t$, $I_{t+1} \to I_t$), respectively. In parallel, SegNN performs pixel-level classifications, which classifies each pixel of $I_{t+1}$ as one of a predefined set of $n$ categories and represents the classification results as $n$ segmentation channels $S^n_{t+1}$. Please note that in this work, $n$ is set to nineteen because SegNN is trained on the Cityscape dataset [42]. These segmentation channels are dynamically weighted by our Attention Module, which highlights their order of significance for each $I_{t+1}$ and produces an attention map $A_{t+1}$. This attention map is then applied to $I_{t+1}$ and $F_{t+1\to t}$ by pixel-wise multiplication to generate a weighted RGB frame $I'_{t+1}$ and a weighted flow map $F'_{t+1\to t}$, respectively. The procedure is formulated as:

$$I'_{t+1} = A_{t+1} \odot I_{t+1} \tag{1}$$

$$F'_{t+1\to t} = A_{t+1} \odot F_{t+1\to t}, \tag{2}$$

where $\odot$ denotes pixel-wise multiplication. PoseNN then takes $I_t$, $F_{t\to t}$, $I'_{t+1}$, and $F'_{t+1\to t}$ as its inputs, encodes them by an eight-layer DCNN, and generates a three degree of freedom (3-DoF) translational motion estimation $\hat{\rho}_{t\to t+1}$ and another 3-DoF rotational motion estimation $\hat{\varphi}_{t\to t+1}$ by two separate DCNN branches named *TransNN* and *RotNN*, respectively. The predicted relative pose $\hat{\chi}_{t\to t+1}$ is given by:

$$\hat{\chi}_{t\to t+1} = (\hat{\rho}_{t\to t+1}, \hat{\varphi}_{t\to t+1}) \tag{3}$$

$$= \mathscr{P}(I_t \oplus F_{t\to t} \oplus I'_{t+1} \oplus F'_{t+1\to t}), \tag{4}$$

where $\mathscr{P}(\cdot)$ denotes PoseNN, $\oplus$ represents the channel-wise concatenation operator, and $\hat{\rho}$ and $\hat{\varphi}$ correspond to the predicted translational and rotational motions, respectively. Note that $F_{t\to t}$ is a zero map consisting of two channels.

Finally, the entire trajectory is generated according to $\hat{\chi}_{t\to t+1} = (\hat{\rho}_{t\to t+1}, \hat{\varphi}_{t\to t+1})$ collected at different frame timestamps. Please note that our framework leverages two pairs of frames $(I_t, I_{t+1})$ and $(I_t, I_{t-1})$ as its inputs to predict $\hat{\chi}_{t\to t+1}$ and $\hat{\chi}_{t\to t-1}$ during the training phase for improving the representation learning of $\hat{\rho}$ and $\hat{\varphi}$. During the evaluation phase, only a single pair $(I_t, I_{t+1})$ is used by our framework.

### B. Attention Module

The Attention Module is designed to validate our assumption that under different motion scenarios, dynamically adjusting the attention weights on different semantic categories for input frames might be beneficial for deriving the pose of the camera. This module takes the flow map $F_{t\to t'}$ produced by FlowNN and the $n$ segmentation channels $S^n_t$ produced by SegNN as its inputs, and employs an attention network *AttentionNN* to generate $n$ attention weights for the segmentation channels. The architecture of our AttentionNN is inspired by SENet [31], which is composed of one global pooling layer and two FC layers. The attention weights reflect the relative importance of the segmentation channels, and are collectively referred to as $a^n_t$. The Attention Module next generates the attention map $A_t$ by multiplying $a^n_t$ with the segmentation channels $S^n_t$ and then performing channel-wise addition of them. The attention map $A_t$ not only dynamically preserves the semantic categories of the segmentation according to $F_t$, but also highlights the relative importance of the regions in $I_t$ and $F_t$ that should be taken into consideration by PoseNN. We therefore formulate above derivation as the following:

$$A_t = \sum_{i=1}^{n} a^i_t \odot S^i_t \text{ where } a^i_t = \mathscr{A}(F_{t\to t'}) \in [0,1] \tag{5}$$

$\mathscr{A}(\cdot)$ represents the AttentionNN which is composed of one positive value transfer, one global pooling layer, and two FC layers, followed by a Tanh layer and a Sigmoid layer, respectively. We validate the design choice of taking the flow map $F_{t\to t'}$ as the input of the Attention Module by an ablation study presented and discussed in Section IV-D.

TABLE I: The parameter setups of *PoseNN*. In our work, the shared layers 1 to 5 of PoseNN extract features for *TransNN* and *RotNN*, while the layers 6 to 8 are used for predicting the translational and rotational motions. The symbols $B$, $H$, $W$, and $C$ denote the batch size, the height and width of the input feature maps, and the number of channels, respectively.

| | PoseNN | | |
|---|---|---|---|
| | TransNN | RotNN | No. of channels |
| Input | Concatenation of two consecutive (B,H,W,C) frames | | C*2 |
| layer 1 | conv, 3x3, stride=2 | | 16 |
| layer 2 | conv, 3x3, stride=2 | | 32 |
| layer 3 | dilated conv, 3x3, rate=2 | | 64 |
| layer 4 | dilated conv, 3x3, rate=4 | | 128 |
| layer 5 | dilated conv, 3x3, rate=8 | | 256 |
| layer 6 | dilated conv, 3x3, rate=2 | dilated conv, 3x3, rate=2 | 128 |
| layer 7 | conv, 3x3, stride=2 | conv, 3x3, stride=2 | 256 |
| layer 8 | conv, 1x1, stride=1 | conv, 1x1, stride=1 | 3 |
| Output | Reduce mean to (B,3)-vector | Reduce mean to (B,3)-vector | |

### C. Dilated Pose Estimation Module

Our PoseNN is developed based on a decoupling architecture, which is inspired by [43]. PoseNN consists of two convolutional layers and three atrous convolutional layers, followed by two decoupled branches *TransNN* and *RotNN*, as described in Section III-A and Table I. PoseNN employs atrous convolutions [40] to enlarge the receptive fields of the convolutional filters so as to provide it with a wider perspective to extract necessary features for performing pose estimation. Atrous convolutions are also commonly referred to as dilated convolutions, which enable DCNN layers to capture features at desired resolutions, resulting in expanded window sizes without increasing the number of filter parameters. This is accomplished by sampling pixels according to the strides given, and inserting zeros into convolutional kernels. The dilation rates of the three atrous convolutional layers in our PoseNN are set to 2, 4, 8 [44], respectively. The five convolutional layers first extract features from consecutive frames and flow maps $I$, $F$, $I'$ and $F'$, and then forward these features to TransNN and RotNN to predict translational motion $\hat{\rho}$ and rotational motion $\hat{\varphi}$ accordingly. Each layer in PoseNN is appended with a layer of ReLU activation function except for the final layers. TransNN and RotNN are made of one atrous convolutional layer with the dilation rate set to 2, followed by two standard convolutional layers. These two branches separately extract their required feature maps, which are further analyzed in Section IV-D.

### D. Training Loss Function

The loss function used for training DAVO is a supervised L2-norm loss for comparing the 6-DoF ground truth pose $\chi$ and the predicted pose $\hat{\chi}$, where $\chi = (\rho, \varphi)$, as defined in Section III-A. It consists of a translational loss term $L_{trans}$ and a rotational loss term $L_{rot}$, and can be represented as:

$$L_{pose} = L_{trans} + \lambda L_{rot}, \tag{6}$$

where $\lambda$ is a scaling factor. Please note that in this work, the value of $\lambda$ is set to ten. $L_{trans}$ and $L_{rot}$ are represented as:

$$L_{trans} = \|\langle\rho\rangle - \langle\hat{\rho}\rangle\|_2 + (\|\rho\|_2 - \|\hat{\rho}\|_2)^2 \tag{7}$$

$$L_{rot} = \|\varphi - \hat{\varphi}\|_2, \tag{8}$$

where $\langle\cdot\rangle$ denotes the Euclidean normalization vector, and $\|\cdot\|_2$ represents the L2-norm operator.

## IV. Experimental Results

In this section, we present our experimental results both qualitatively and quantitatively, and discuss the implications.

### A. Experimental Setup

**General setups.** The proposed framework is implemented on top of TensorFlow [46], and is trained on a server equipped with an Intel i9-7990X CPU and three NVIDIA GeForce GTX 2080 Ti GPUs. The Adam Optimizer [47] is used for training with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The learning rate is initialized to 0.001, and is multiplied by 0.9 every 100K iterations. The input image size is scaled down to $416 \times 128$ in order to fit in the memory space of the GPUs. In order to enhance the performance and avoid overfitting, our framework is trained using geometric augmentation [48]. Moreover, our proposed DAVO framework and its variants developed for our ablation analyses are trained for 1,500k iterations to ensure convergence of the models.

**Baselines.** We compare the results of DAVO in terms of pose estimation accuracy against a number of baseline methods. These baseline methods include non-DCNN based visual SLAM framework ORB-SLAM2 [16] (without enabling the loop closing for a fair comparison with the baseline VO algorithms), the open-source visual odometry library VISO2 [45], as well as the related learning-based VO works discussed in Section II, such as CL-VO [11], DeepVO [12], ESP-VO [13], SRNN_channel [14], and the method proposed by Fei Xue *et al.* [37]. Please note that we do not consider the methods that exploit depth maps, because these methods are not fairly comparable to the scope of this paper.

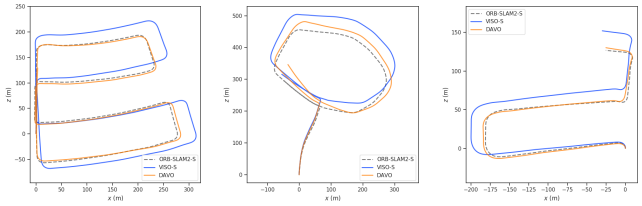
**Evaluation benchmark.** We evaluated DAVO and the baselines on the famous KITTI Visual Odometry and SLAM benchmark suite [33], which contains eleven annotated video sequences. Similar to the baselines [11-14], DAVO and its variants are trained on sequences 00, 02, 08, 09, and evaluated on sequences 03, 04, 05, 06, 07, 10. The performance of the evaluated trajectories for the sequences is measured and reported using a metric called *relative trajectory error (RTE)*, which is the benchmark metric adopted in KITTI [33] for measuring the relative translational and rotational errors $t_{rel}$ and $r_{rel}$, respectively.

In Section IV-C, we additionally plot trajectories of our framework trained on sequences 00, 02, 08, 09 and evaluated on the testing sequences 16, 18 of the KITTI Visual Odometry Dataset [33], and sequence `2011_09_26_drive_0022_sync` (denoted as 22) of the KITTI Raw Dataset [33]. We compare the evaluated trajectories with those obtained from ORB-SLAM2 stereo version (denoted as ORB-SLAM2-S) [16] and VISO2 stereo version (denoted as VISO2-S) [45], since these testing sequences do not offer the ground truth data. We adopt the trajectories generated by ORB-SLAM2-S as our reference ones, because

TABLE II: Comparison of the evaluated $t_{rel}$ and $r_{rel}$ for different evaluation sequences selected from [33].

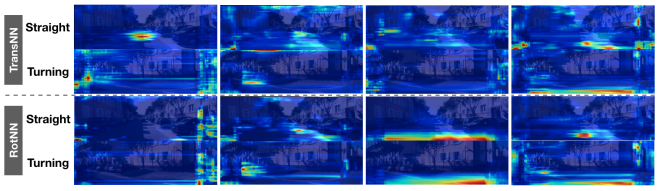
| Method | Seq. 03 | | Seq. 04 | | Seq. 05 | | Seq. 06 | | Seq. 07 | | Seq. 10 | | Ave. | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $t_{rel}$ | $r_{rel}$ | $t_{rel}$ | $r_{rel}$ | $t_{rel}$ | $r_{rel}$ | $t_{rel}$ | $r_{rel}$ | $t_{rel}$ | $r_{rel}$ | $t_{rel}$ | $r_{rel}$ | $t_{rel}$ | $r_{rel}$ |
| **Non-Learning-based Monocular** VO | | | | | | | | | | | | | | |
| VISO2-M [45] | 17.55 | 0.0626 | 7.71 | 0.0180 | 18.87 | 0.0516 | 11.03 | 0.0326 | 16.03 | 0.1026 | 30.17 | 0.0577 | 16.89 | 0.0542 |
| ORB-SLAM2-M [16] w/o LoopClosing | 1.37 | 0.0022 | 1.23 | 0.0019 | 17.46 | 0.0063 | 21.02 | 0.0026 | 12.74 | 0.0143 | 4.44 | 0.0044 | 9.71 | 0.0053 |
| **Learning-based Monocular** VO | | | | | | | | | | | | | | |
| CL-VO [11] | 8.12 | 0.0347 | 7.57 | 0.0261 | 5.77 | 0.0200 | 7.66 | 0.0166 | 6.79 | 0.0300 | 8.29 | 0.0294 | 7.37 | 0.0261 |
| ESP-VO [13] | 6.72 | 0.0646 | 6.33 | 0.0608 | 3.35 | 0.0493 | 7.24 | 0.0729 | 3.52 | 0.0502 | 9.77 | 0.1020 | 6.16 | 0.0666 |
| DeepVO [12] | 8.49 | 0.0689 | 7.19 | 0.0697 | 2.62 | 0.0361 | 5.42 | 0.0582 | 3.91 | 0.0460 | 8.11 | 0.0883 | 5.96 | 0.0612 |
| SRNN_channel [14] | 5.44 | 0.0322 | **2.91** | **0.0130** | 3.27 | 0.0162 | 8.50 | 0.0274 | 3.37 | 0.0225 | 6.32 | 0.0233 | 4.97 | 0.0224 |
| Fei Xue *et al.* [37] | 3.32 | 0.0210 | 2.96 | 0.0176 | 2.59 | 0.0125 | 4.93 | 0.0190 | 3.07 | **0.0176** | **3.94** | 0.0172 | **3.47** | 0.0175 |
| **Our Proposed Methods** | | | | | | | | | | | | | | |
| Our framework w/o Attention Modele | 4.08 | 0.0205 | 10.07 | 0.0149 | 2.48 | **0.0108** | 3.34 | 0.0084 | 6.53 | 0.0462 | 5.53 | 0.0189 | 5.34 | 0.0199 |
| Our framework w/ Feature Attention | **2.54** | **0.0183** | 7.23 | 0.0189 | 3.45 | 0.0129 | 3.97 | 0.0139 | 4.81 | 0.0329 | 6.85 | 0.0252 | 4.81 | 0.0204 |
| Our framework w/ Static Attention | 5.67 | 0.0246 | 6.89 | 0.0270 | 2.69 | 0.0134 | 2.48 | 0.0086 | 5.64 | 0.0313 | 5.26 | 0.0189 | 4.77 | 0.0207 |
| DAVO (segementation source) | 5.77 | 0.0248 | 8.51 | 0.0243 | 2.95 | 0.0137 | 3.49 | 0.0164 | 2.90 | 0.0217 | 4.94 | 0.0233 | 4.76 | 0.0207 |
| DAVO (depth source) | 3.60 | 0.0231 | 5.42 | 0.0295 | 3.25 | 0.0134 | 3.04 | 0.0107 | 6.20 | 0.0319 | 6.91 | 0.0228 | 4.74 | 0.0219 |
| DAVO (rgb source) | 5.50 | 0.0271 | 6.03 | 0.0237 | **2.28** | 0.0114 | 4.19 | 0.0169 | 4.11 | 0.0261 | 4.26 | 0.0170 | 4.40 | 0.0204 |
| DAVO | 3.39 | 0.0194 | 7.07 | **0.0130** | 2.54 | 0.0109 | **2.31** | **0.0083** | **2.78** | 0.0198 | 5.37 | **0.0164** | 3.91 | **0.0146** |

[1] $t_{rel}$ : Average translational RMSE drift (%) on length from 100, 200 to 800 m.
[2] $r_{rel}$ : Average rotational RMSE drift (°/100m) on length from 100, 200 to 800 m.




(a) Sequence 16. (b) Sequence 18. (c) Sequence 22.

Fig. 3: Comparison of the trajectories evaluated on the test sequences (described in Section IV-A) selected from [33].




(a) DAVO (b) Ours w/o attention (c) Ours w/ static attention (d) Ours w/ feature attention

Fig. 4: Visualization of the feature maps extracted from TransNN and RotNN for DAVO and its three variants.

ORB-SLAM2-S involves global optimization steps such as loop closure detection and bundle adjustment.

### B. Comparison of the Quantitative Results

Table II compares the evaluation results of DAVO against the baselines. The first column corresponds to the names of the VO methods. The rest of the columns correspond to the results measured from the evaluation sequences. We report the results of the baselines at the upper rows and place the results of DAVO and its variants at the bottom rows. It is observed that DAVO outperforms most of the baseline methods, and delivers comparable performance to [37], which additionally employs a spatial-temporal latent feature-based attention unit for weighting the hidden states of its recurrent memory cells as well as a refining module for aggregating them in its architecture. The averaged $t_{rel}$ of DAVO (the bottom row) is slightly (12.70%) higher than that of [37]. However, DAVO delivers a lower (19.86%) averaged $r_{rel}$ than [37] without using any recurrent memory cell. When compared with SRNN_channel, which is the best one among the rest of the learning-based VO baselines, the averaged $t_{rel}$ and $r_{rel}$ of DAVO are 21.33% and 34.82% lower than those of SRNN_channel, respectively. The results presented in Table II therefore validate the effectiveness and advantage of DAVO. We further investigate a variant of DAVO that adopts the feature-based attention mechanism similar to [14] in our ablation analyses. Please note that we do not implement a latent feature-based attention module as in [37], since we do not use any recurrent memory cell in our DAVO framework.

### C. Comparison of the Generated Trajectories

Fig. 3 plots the trajectories of DAVO, VISO2-S, and ORB-SLAM2-S evaluated on the test sequences 16, 18, and 22. The reference trajectories (i.e., ORB-SLAM2-S) are plotted in dashed curves. These test sequences cover a wide variety of paths with various lengths and different number of turns. As pose estimation errors accumulate along each path and may dramatically influence the resultant shape of predicted trajectories, these experiments thus suffice to examine the effectiveness of different VO models. It is observed that DAVO outperforms VISO2-S in all of the above test sequences. Compared with VISO-S, the trajectories generated by DAVO are more closely matched to the reference ones. For sequences with evaluation length longer than 300 meters, DAVO is able to generate trajectories sufficiently aligned with the reference ones. This observation justifies that DAVO is less susceptible to the evaluation length than VISO-S, whose trajectories gradually deviate from the reference trajectories during the course of evaluation. For sequences with several turns (e.g., sequences 16 and 18), DAVO still performs better than VISO-S and demonstrates satisfactory results. This suggests that DAVO is capable of handling both straight moves and turns, rather than being only applicable to limited scenarios. These qualitative results as well as the insights discussed above thus justify the effectiveness of the proposed DAVO framework.

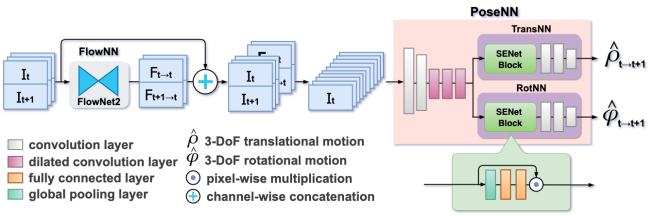



Fig. 5: Illustration of our feature-based attention variant.

### D. Ablation Analysis

In this section, we provide a set of ablation analyses with an aim to validate our design choices employed in DAVO.

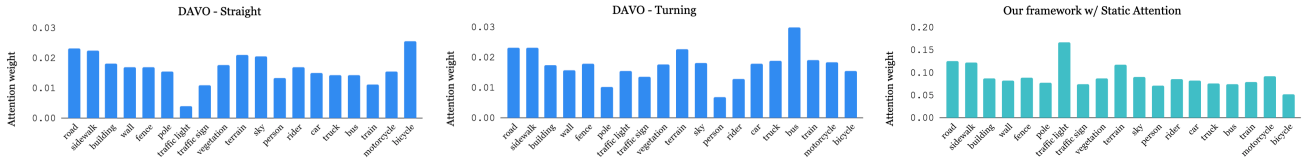
*1) DAVO w/ and w/o the Attention Module:* We first inspect the impact of the dynamic attention weights by evaluating our framework with and without the Attention Module, where the former corresponds to DAVO. This experiment intends to validate if the attention module does offer a positive impact on PoseNN. The results of the two cases are summarized in Table II. It is observed that for most sequences presented in Table II, the Attention Module does help DAVO to bring down the averaged $t_{rel}$ and $r_{rel}$, resulting in a reduction of them by 26.78% and 26.63%, respectively. To further elaborate on the above observation, we take sequence 07 as an example and visualize the focused regions of the feature maps [49] within PoseNN for the two cases in Fig. 4. We select this evaluation sequence because it contains both straight roads and clear turns, and is thus suitable for demonstrating the difference between the two cases. For DAVO, it is observed that the proposed Attention Module enables the focused regions of TransNN to become steady and concentrated on straight roads (i.e., Fig. 4 (a)). During turning, the feature maps of RotNN concentrate on the sides of the road, enabling DAVO to leverage the changes from the sides of the frames to infer the turning angle. In contrast, when the Attention Module of DAVO is removed (i.e., Fig. 4 (b)), the focused regions become unsteady and are scattered to multiple uncorrelated parts of the input frames for both TransNN and RotNN, leading to a degradation in pose estimation accuracy. Please refer to our supplementary video for a more detailed demonstration and visualization of the above cases.

*2) Comparison of dynamic and static attention weights:* In order to examine the benefits of the proposed dynamic attention weights over the static ones, we next compare the performance of DAVO with a variant which replaces the dynamically adjusted attention weights in DAVO by a set of static weights. These static weights are treated as trainable parameters for optimization during the training phase, and are then fixed during the evaluation phase. The sequences used for training and evaluation are the same as those in Section IV-B. Fig. 6 compares the dynamic attention weights of DAVO with the derived static weights for different motion scenarios. These attention weights reveal that when the static attention mechanism is adopted, semantics such as road, sidewalk, terrain, and traffic light are more emphasized than person, car, and rider, as the objects belonging to the latter group might move and is less favorable for pose estimation. On the other hand, the emphasis of the dynamic attention weights changes under different motion scenarios. We further plot the ground truth camera motion (the top part) and the attention weights of the nineteen semantic categories (the bottom part) versus frame id for the evaluation sequence 07 in Fig. 7. It is observed that the peaks in yaw of the ground truth motion aligns with the periods of changes in the dynamic attention weights. In other words, the dynamic weights change values during turning motions.

Table II compares the results for the above cases. These results suggest that when the static attention weights are used, the values of $t_{rel}$ and $r_{rel}$ for each sequence improve significantly when compared to the case without the Attention Module. The averaged values of $t_{rel}$ and $r_{rel}$ are decreased by 10.67% and increased by 3.86%, respectively. When the dynamic attention mechanism is employed (i.e., DAVO), the averaged $t_{rel}$ and $r_{rel}$ are further decreased by 18.03% and 29.47% with respect to the static attention case.

In order to explain this improvement, we similarly plot the focused regions of the feature maps for the static attention case in Fig. 4 (c). It is observed that the focused regions of TransNN are widely scattered within the road areas as well as several other regions of the input frame, rather than concentrating on one or more specific regions or semantic categories as Fig. 4 (a). On the other hand, the focused regions of RotNN consistently fall on the bottom part of roads, no matter the camera is moving forward or making turns. Merely concentrating on the bottom regions of roads may restrict RotNN from acquiring sufficient information for inferring the correct turning angles, which in turn results in a degradation in the averaged value of $r_{rel}$. The above observation therefore provides the rationale of employing the dynamic attention weights in the proposed DAVO framework, rather than adopting and optimizing a set of static ones.

*3) Comparison of DAVO with a variant architecture that employs feature-based attention mechanism:* In order to validate the effectiveness of the proposed dynamic attention mechanism, in this section, we investigate a feature-based attention variant of DAVO and compare its performance against the proposed DAVO framework illustrated in Fig. 2. The architecture of this variant is illustrated in Fig. 5. For this variant, our Attention Module described in Section III-B is removed, and two additional SENet blocks [31] are separately inserted into the TransNN and RotNN branches to serve as the feature-based attention modules. Such a feature-based attention mechanism is recently introduced by SRNN_channel [14]. The sequences used for training and evaluation are the same as those described in Section IV-B. The values of $t_{rel}$ and $r_{rel}$ of this variant are also presented in Table II. For this variant, it is observed that the averaged values of $t_{rel}$ and $r_{rel}$ are slightly better than those of the case without using any attention module (i.e., the case described in Section IV-D.1). However, they are still worse than those obtained by DAVO significantly. The visualization of the feature maps presented in Fig. 4 (d) shows that for TransNN, the focused regions of this feature-based variant are also not concentrated and scattered to multiple parts of the input frames. For RotNN, the focused regions merely




(a) DAVO attentions when moving straight. (b) DAVO attentions when making a turn. (c) Attention weights of our static variant.

Fig. 6: Attention weights of semantic categories learned by DAVO and our framework with static attention mechanism.

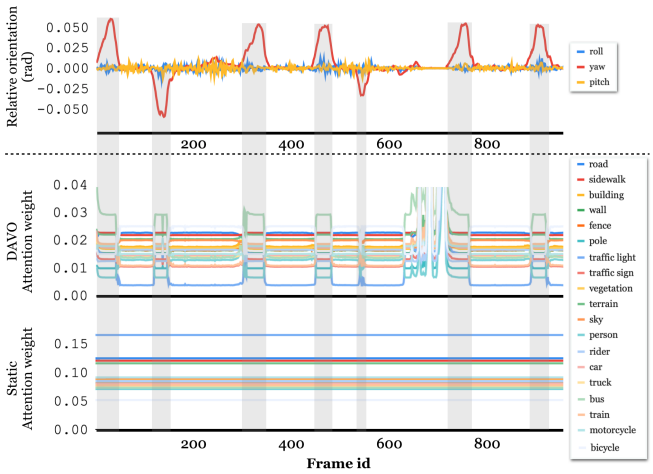



Fig. 7: The relation between the camera motion and the attention weights in sequence 07. The horizontal axis corresponds to the frame id. The chart on the top shows the ground truth motion of the camera, while the chart on the bottom displays the changes in attention weights for the 19 semantic categories. Please note that the noisy region between the $650^{th}$ and $730^{th}$ frames is caused by the traffic flow in front the camera when it is waiting to make a turn. We plot the static weights discussed in Section IV-D.2 for a reference.

fall on an even narrower range of the bottom part of roads than Fig. 4 (c). It leads to a degradation in performance, as focusing on a narrow region may constraint the features available to RotNN.

*4) Comparison of different input sources for generating the dynamic attention weights:* In this section, we present an ablation study of using different types of input sources to generate the dynamic attention weights for validating our design of the Attention Module in DAVO. Under the same framework, we additionally attempt other types of inputs for AttentionNN, including raw RGB frame $I$, depth map, and semantic segmentation channels $S^n$, to generate the attention weights for the segmentation channels. We train these variant frameworks with the same hyperparameter setup and training procedure described in Section IV-A. For the depth map variant, depth maps are produced by Monodepth2 [21]. The results are also summarized in Table II. It is observed that when segmentation channels $S^n$ are used as the input of AttentionNN for generating the dynamic attention weights, the averaged values of $t_{rel}$ and $r_{rel}$ are 4.76% and $0.207^\circ/km$, respectively, significantly higher than those of DAVO. When depth maps are used instead, the averaged value of $t_{rel}$ is further reduced by 0.42%, while the averaged value of $r_{rel}$ is increased by 5.48%. The use of RGB frames enables the averaged values of $t_{rel}$ and $r_{rel}$ to be decreased to 4.40% and $0.204^\circ/km$, respectively. However, these variant frameworks are still not comparable to DAVO. One potential explanation is that compared with the other types of input sources, flow maps directly reflects the changes between consecutive frames, while the other types of sources embed such changes implicitly. This ablation study thus justifies the use of flow maps for generating our dynamic attention weights in DAVO.

## V. Conclusions

In this paper, we proposed DAVO, a learning-based framework for estimating the ego-motion of a monocular camera. In order to validate the proposed framework, we examined DAVO as well as its variants, and compared them with the other contemporary VO approaches on the KITTI Visual Odometry and SLAM benchmark suite. In our experiments, DAVO demonstrated superior performance to the baseline methods both quantitatively and qualitatively. In addition, we provided a set of ablation analyses, validating each of our design choices adopted in the proposed framework. As the proposed mechanism that leverages dynamic attention weights on different semantic categories has been validated effective and beneficial in this work, DAVO thus offers a promising direction for future attention-based VO researches.

## Acknowledgments

This work was supported by the Ministry of Science and Technology (MOST) in Taiwan under grant nos. MOST 109-2636-E-007-018 (Young Scholar Fellowship Program) and MOST 109-2634-F-007-017. The authors acknowledge the financial support from MediaTek Inc., Taiwan. The authors would also like to acknowledge the donation of the GPUs from NVIDIA Corporation and NVIDIA AI Technology Center (NVAITC) used in this research work.

## References

[1] Y. Almalioglu, M. R. U. Saputra, P. P. B. de Gusmão, A. Markham, and A. Trigoni. GANVO: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 5474–5480. IEEE, May 2019. 1

[2] V. M. Babu, K. Das, A. Majumdar, and S. Kumar. UnDEMoN: Unsupervised deep network for depth and ego-motion estimation. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pages 1082–1088, Sep. 2018. 1

[3] V. M. Babu, S. Kumar, A. Majumder, and K. Das. UnDEMoN 2.0: Improved depth and ego motion estimation through deep image sampling, Nov. 2018. 1

[4] G. Costante and T. A. Ciarfuglia. LS-VO: Learning dense optical subspace for robust visual odometry estimation. In *Proc. IEEE Robotics and Automation Letters (RA-L)*, volume 3, pages 1735–1742, Sep. 2017. 1, 2

[5] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. SVO: Semidirect visual odometry for monocular and multicamera systems. *IEEE Trans. Robotics*, 33:249–265, 2017. 1

[6] K. R. Konda and R. Memisevic. Learning visual odometry with a convolutional network. In *Proc. Int. Conf. Computer Vision Theory and Applications (VISAPP)*, Mar. 2015. 1

[7] R. Li, S. Wang, Z. Long, and D. Gu. UnDeepVO: Monocular visual odometry through unsupervised deep learning. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 7286–7291, May 2017. 1

[8] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3D geometric constraints. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 5667–5675, Jun. 2018. 1, 2

[9] V. Mohanty, S. Agrawal, S. Datta, A. Ghosh, V. D. Sharma, and D. Chakravarty. DeepVO: A deep learning approach for monocular visual odometry, Nov. 2016. 1

[10] P. Muller and A. E. Savakis. Flowdometry: An optical flow and deep learning based approach to visual odometry. In *Proc. IEEE Winter Conf. Applications of Computer Vision (WACV)*, pages 624–631, May 2016. 1, 2

[11] M. R. U. Saputra, P. P. B. de Gusmao, S. Wang, A. Markham, and A. Trigoni. Learning monocular visual odometry through geometry-aware curriculum learning. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 3549–3555, May 2019. 1, 2, 4, 5

[12] S. Wang, R. Clark, H. Wen, and A. Trigoni. DeepVO: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 2043–2050, May 2017. 1, 2, 4, 5

[13] S. Wang, R. Clark, H. Wen, and A. Trigoni. End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. *Int. J. Robotics Res.*, 37:513–542, 2018. 1, 2, 4, 5

[14] F. Xue, Q. Wang, X. Wang, W. Dong, J. Wang, and H. Zha. Guided feature selection for deep visual odometry. *Proc. Asian Conf. Computer Vision (ACCV)*, Dec. 2018. 1, 2, 3, 4, 5, 6

[15] A. J. Davison, I. D. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29:1052–1067, 2007. 1

[16] R. Mur-Artal and J. D. Tardós. ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Trans. Robotics*, 33(5):1255–1262, Jun. 2017. 1, 4, 5

[17] X. Wang, H. B. Zhang, X. Yin, M. Du, and Q. Chen. Monocular visual odometry scale recovery using geometrical constraint. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 988–995, May 2018. 1, 2

[18] P. Ganti and S. Waslander. Network uncertainty informed semantic feature selection for visual SLAM. In *Proc. Conf. Computer and Robot Vision (CRV)*, pages 121–128. IEEE, May 2019. 1

[19] K.-N. Lianos, J. L. Schönberger, M. Pollefeys, and T. Sattler. VSO: Visual semantic odometry. In *Proc. European Conf. Computer Vision (ECCV)*, Sep. 2018. 1

[20] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia. Every pixel counts: Unsupervised geometry learning with holistic 3D motion understanding. In *Proc. European Conf. Computer Vision (ECCV) Workshop*, Mar. 2018. 1

[21] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. Digging into self-supervised monocular depth prediction. In *The International Conference on Computer Vision (ICCV)*, Oct. 2019. 1, 7

[22] S. Y. Loo, A. J. Amiri, S. Mashohor, S. H. Tang, and H. Zhang. CNN-SVO: Improving the mapping in semi-direct visual odometry using single-image depth prediction. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 5218–5223, 2018. 1

[23] T. Shen, Z. Luo, L. Zhou, H. Deng, R. Zhang, T. Fang, and L. Quan. Beyond photometric loss for self-supervised ego-motion estimation. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 6359–6365, May 2019. 1

[24] Z. Yin and J. Shi. GeoNet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1983–1992, Jun. 2018. 1

[25] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 6612–6619, Jul. 2017. 1

[26] C. Zhao, L. Sun, P. Purkait, T. Duckett, and R. Stolkin. Learning monocular visual odometry with dense 3D mapping from sense 3D flow. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pages 6864–6871, Oct. 2018. 1

[27] B. Bescos, J. M Fácil, J. Civera, and J. Neira. Dynaslam: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters*, 3(4):4076–4083, 2018. 1, 2

[28] T. Sun, Y. Sun, M. Liu, and D.-Y. Yeung. Movable-object-aware visual slam via weakly supervised semantic segmentation. *ArXiv*, 2019. 1, 2

[29] 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 *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 1647–1655, Jul. 2017. 1, 2, 3

[30] E. Parisotto, D. Singh Chaplot, J. Zhang, and R. Salakhutdinov. Global pose estimation with an attention-based recurrent network. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR) Workshop*, pages 237–246, Jun. 2018. 2, 3

[31] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 7132–7141, Jun. 2018. 2, 3, 6

[32] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic. Seeing 3D chairs: Exemplar part-based 2D-3D alignment using a large dataset of CAD models. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 3762–3769, Jun. 2014. 2

[33] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, Jun. 2012. 2, 3, 4, 5

[34] G. Iyer, J. K. Murthy, G. Gupta, K. M. Krishna, and L. Paull. Geometric consistency for self-supervised end-to-end visual odometry. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshop*, pages 380–388, Jun. 2018. 2

[35] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia. Exploring representation learning with CNNs for frame-to-frame ego-motion estimation. In *Proc. IEEE Robotics and Automation Letters (RA-L)*, volume 1, pages 18–25, 2016. 2

[36] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. FlowNet: Learning optical flow with convolutional networks. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pages 2758–2766, Dec. 2015. 2

[37] F. Xue, d S. Li X. Wang, Q. Wang, J. Wang, and H. Zha. Beyond tracking: Selecting memory and refining poses for deep visual odometry. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019. 2, 4, 5

[38] C. Chen, S. Rosa, Y. Miao, C. X. Lu, W. Wu, A. Markham, and N. Trigoni. Selective sensor fusion for neural visual-inertial odometry. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pages 10542–10551, Jun. 2019. 2, 3

[39] M. Kaneko, K. Iwami, T. Ogawa, T. Yamasaki, and K. Aizawa. Mask-slam: Robust feature-based monocular slam by masking using semantic segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR) Workshop*, pages 258–266, Jun. 2018. 2

[40] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. European Conf. Computer Vision (ECCV)*, Sep. 2018. 3, 4

[41] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017. 3

[42] S. Ramos T. Rehfeld M. Enzweiler R. Benenson U. Franke S. Roth M. Cordts, M. Omran and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[43] P. Kim, B. Coltin, and H. J. Kim. Low-drift visual odometry in structured environments by decoupling rotational and translational motion. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 7247–7253, May 2018. 4

[44] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI)*, Apr. 2017. 4

[45] A. Geiger, J. Ziegler, and C. Stiller. StereoScan: Dense 3D reconstruction in real-time. In *Proc. IEEE Intelligent Vehicles Symposium (IV)*, pages 963–968, Jun. 2011. 4, 5

[46] M. Abadi *et al.* TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 4

[47] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. Int. Conf. Learning Representations (ICLR)*, May 2015. 4

[48] C. Zhao, L. Sun, P. Purkait, T. Duckett, and R. Stolkin. Learning monocular visual odometry with dense 3d mapping from dense 3d flow. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pages 6864–6871. IEEE, Oct. 2018. 4

[49] J. Brownlee. How to visualize filters and feature maps in convolutional neural networks. https://reurl.cc/kdDy1x, May 2019. 6