2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Visual Odometry Revisited: What Should Be Learnt?

Huangying Zhan, Chamara Saroj Weerasekera, Jia-Wang Bian, Ian Reid

***Abstract*— In this work we present a monocular visual odometry (VO) algorithm which leverages geometry-based methods and deep learning. Most existing VO/SLAM systems with superior performance are based on geometry and have to be carefully designed for different application scenarios. Moreover, most monocular systems suffer from scale-drift issue. Some recent deep learning works learn VO in an end-to-end manner but the performance of these deep systems is still not comparable to geometry-based methods. In this work, we revisit the basics of VO and explore the right way for integrating deep learning with epipolar geometry and Perspective-n-Point (PnP) method. Specifically, we train two convolutional neural networks (CNNs) for estimating single-view depths and two-view optical flows as intermediate outputs. With the deep predictions, we design a simple but robust frame-to-frame VO algorithm (DF-VO) which outperforms pure deep learning-based and geometry-based methods. More importantly, our system does not suffer from the scale-drift issue being aided by a scale consistent single-view depth CNN. Extensive experiments on KITTI dataset shows the robustness of our system and a detailed ablation study shows the effect of different factors in our system. Code is available at here: DF-VO.**

I. Introduction

The ability for an autonomous robot to know its whereabouts and its surroundings is of utmost importance for tasks such as object manipulation and navigation. Vision-based localisation and mapping is often the preferred choice due to factors such as low cost and power requirements, and it can provide useful complementary information to other sensors such as IMU, GPS, laser scanners, etc. Two broad types of visual localisation methods are: Visual Odometry – the main focus area of this work – and Simulataneous Localisation and Mapping (SLAM). VO is useful when the (6DoF) motion of the robot relative to its previous state is of main interest, while visual SLAM is more suited when an accurate robot trajectory and map of the environment are required, and the latter also involves closing loops and re-localizing the robot when tracking is lost and it is re-visiting the same environment again. Moreover, VO is an integral part of SLAM for inter-frame tracking, and more accurate odometry reduces overall drift in localisation, and minimises the need for loop closure and global refinement of the camera trajectory.

Pure multi-view geometry-based VO is reliable and accurate under favourable conditions, such as when there is sufficient illumination and texture to establish correspondence [1]–[3], sufficient overlap between consecutive frames, and when majority of the visible scene is static (few moving objects). Having an accurate estimation of translation scale per-frame is crucial for any monocular VO/SLAM system. However, pure geometry-based VO/SLAM suffer from the scale drift issue. Resolving scale drift usually relies on keeping a scale consistent map for map-to-frame tracking, performing an expensive global bundle adjustment for scale optimization or additional prior assumptions like constant camera height from the known ground plane. Moreover, most monocular systems suffer from a single depth-translation scale ambiguity issue, which means the predictions (structure and motion) are up-to-scale.

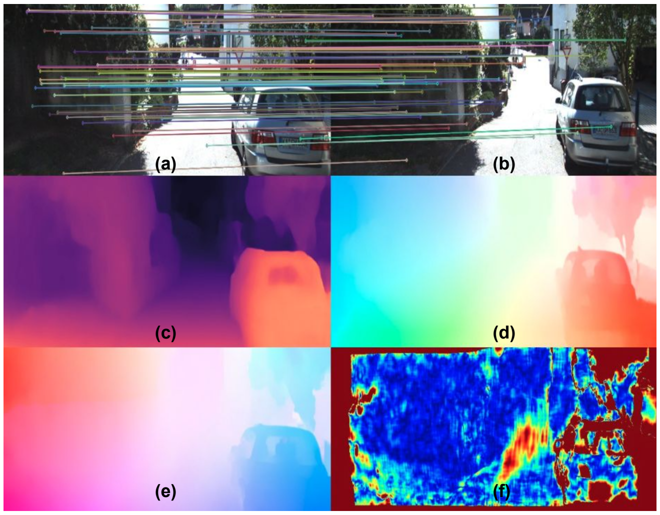

Fig. 1. Inputs and intermediate CNN outputs of the system. (a, b) Current and previous monocular input images with examples of auto-selected 2D-2D matches; (c) Single-view depth prediction; (d, e) Forward and backward optical flow prediction; (f) Forward-backward flow consistency using (d,e).

Recently deep learning based methods have made possible end-to-end learning of camera motion from videos [4]–[8], or prediction of pose with respect to a scene from a single view [9]–[11]. Also, learning from data allows the deep networks to give predictions associated with real world scale, which solves the scale ambiguity issue. While these learnt systems enable camera tracking/localisation in challenging conditions, these systems fail to provide the reliablility and accuracy of pure geometry based methods in those conditions that favour geometry-based algorithms.

In this work we revisit the basics of geometry-based VO and explore the right way of incorporating deep learning into it. A simple and robust frame-to-frame VO algorithm, named DF-VO, incorporating deep predictions (Fig. 1) is proposed. We extensively compare our system against both deep learning methods and geometry methods. Moreover, we conduct a detailed ablation study for evaluating the effect of different factors in our system.

All authors are with the School of Computer Science, at the University of Adelaide, and Australian Centre for Robotic Vision

978-1-7281-7395-5/20/$31.00 ©2020 IEEE

II. Related Work

Geometry based VO: Camera tracking is a fundamental and well studied problem in computer vision, with different pose estimation methods based on multiple-view geometry been established [12] [13]. Early work in VO dates back to the 1980s [14] [13], with a successful application of it in the Mars exploration rover in 2004 [15], albeit with a stereo camera. Two dominant methods for geometry based VO/SLAM are feature-based [16]–[18] and direct methods [19], [20]. The former involves explicit correspondence estimation, and the latter takes the form of an energy minimisation problem based on the image color/feature warp error, parameterized by pose and map parameters. There are also hybrid approaches which make use of the good properties of both [21]–[23]. One of the most successful and accurate full SLAM systems using a sparse (ORB) feature-based approach is ORB-SLAM2 [16], along with DSO [19], a direct keyframe-based sparse SLAM method. VISO2 [18] on the other hand is a feature-based VO system which only tracks against a local map created by the previous two frames. All of these methods suffer from the previously mentioned issues (including scale-drift) common to monocular geometry-based systems, which we show our proposed method addresses.

Deep learning for VO: For supervised learning, Agrawal *et al.* [24] propose to learn good visual features from a ego-motion estimation task, in which the model is capable of relative camera pose estimation. Wang *et al.* [25] propose a recurrent network for learning VO from videos. Ummenhofer *et al.* [26] and Zhou *et al.* [27] propose to learn monocular depth estimation and VO together in an end-to-end fashion by formulating structure from motion as a supervised learning problem. Dharmasiri *et al.* [28] train a depth network and extend the depth system for predicting optical flows and camera motion. Recent works suggest that both tasks can be jointly learnt in a self-supervised manner using a photometric warp loss to replace a supervised loss based on ground truth. SfM-Learner [4] is the first self-supervised method for jointly learning camera motion and depth estimation. SC-SfM-Learner [8] is a very recent work which solves the scale inconsistent issue in SfM-Learner by enforcing depth consistency. Some prior work like, UnDeepVO [29] and Depth-VO-Feat [5], solve the both scale ambiguity and inconsistency issue by using stereo sequences in training, which address the issue of metric scale. [6], [7] also incorporate optical flow in their joint training framework.

The issue with the above learning-based methods is that they don't explicitly account for the multi-view geometry constraints that are introduced due to camera motion *during inference*. In order to address this, recent works have been proposed to combine the best of learning and geometry to varying extent and degree of success. CNN-SLAM [30] fuse single-view CNN depths in a direct SLAM system, and CNN-SVO [31] initialize the depth at a feature location with CNN provided depth for reducing the uncertainty in the initial map. Yang *et al.* [32] feed depth predictions into DSO [19] as virtual stereo measurements. Li *et al.* [33] refine their pose predictions via pose-graph optimisation. In contrast to the above methods, we effectively utilize CNNs for *both* single-view depth prediction and correspondence estimation, on top of standard multi-view geometry to create a simple yet effective VO system.

Algorithm 1 DF-VO: Depth and Flow for Visual Odometry

```
Require: Depth-CNN: $M_d$; Flow-CNN: $M_f$
Input: Image sequence: $[I_1, I_2, ..., I_k]$
Output: Camera poses: $[\boldsymbol{T_1}, \boldsymbol{T_2}, ..., \boldsymbol{T_k}]$
Initialization $\boldsymbol{T_1} = \boldsymbol{I}$ ; $i = 2$
while $i \leq k$ do
    Get CNN predictions: $\boldsymbol{D}_i, \boldsymbol{F}^i_{i-1}$, and $\boldsymbol{F}^{i-1}_i$
    Compute forward-backward flow inconsistency.
    Form $N$-matches $(\boldsymbol{P}_i, \boldsymbol{P}_{i-1})$ from flows with the least flow inconsistency.
    if mean$(|\boldsymbol{F}'|) > \delta_f$ then
        Solve $\boldsymbol{E}$ from $(\boldsymbol{P}_i, \boldsymbol{P}_{i-1})$ and recover $[\boldsymbol{R}, \hat{\mathbf{t}}]$
        Triangulate $(\boldsymbol{P}_i, \boldsymbol{P}_{i-1})$ to get $\boldsymbol{D}'_i$
        Estimate scaling factor, $s$, by comparing $(\boldsymbol{D}_i, \boldsymbol{D}'_i)$
        $\boldsymbol{T}^{i-1}_i = [\boldsymbol{R}, s\hat{\mathbf{t}}]$
    else
        Form 3D-2D correspondences from $(\boldsymbol{D}_i, \boldsymbol{F}')$
        Estimate $[\boldsymbol{R}, \boldsymbol{t}]$ using PnP
        $\boldsymbol{T}^{i-1}_i = [\boldsymbol{R}, \boldsymbol{t}]$
    end if
    $\boldsymbol{T}_i \leftarrow \boldsymbol{T}_{i-1}\boldsymbol{T}^{i-1}_i$
end while
```

III. DF-VO: Depth and Flow for Visual Odometry

In this section, we revisit traditional pose estimation methods including epipolar geometry-based and PnP-based methods. We then follow up on the integration of these methods with our learned CNN outputs (Alg.1).

A. Traditional Visual Odometry

a) Epipolar Geometry: Given an image pair, (I_1, I_2), the basic method for estimating the relative camera pose is solving fundamental or essential matrix, $\boldsymbol{E}$. When 2D-2D pixel correspondences $(\boldsymbol{p_1}, \boldsymbol{p_2})$ between the image pair are formed, epipolar constraint is employed for solving the essential matrix. Thus, relative pose, $[\boldsymbol{R}, \boldsymbol{t}]$, can be recovered [34]–[37].

$$\boldsymbol{p}_2^T \boldsymbol{K}^{-\boldsymbol{T}} \boldsymbol{E} \boldsymbol{K}^{-1} \boldsymbol{p}_1 = 0, \text{where } \boldsymbol{E} = [\boldsymbol{t}]_\times \boldsymbol{R} \tag{1}$$

where $\boldsymbol{K}$ is the camera intrinsics. Typically, the 2D-2D pixel correspondences are formed either by extracting and matching salient feature points in the images, or by computing optical flow. However, solving essential matrix for camera pose has some well-known issues.

- Scale ambiguity: translation recovered from essential matrix is up-to-scale.
- Pure rotation issue: recovering $\boldsymbol{R}$ becomes unsolvable if the camera motion is pure rotation.
- Unstable solution: the solution is unstable if the camera translation is small.

b) PnP: Perspective-n-Point (PnP) is a classic method for solving camera pose given 3D-2D correspondences. Suppose the observed 3D points of view 1 and the observed projection in view 2 $(\boldsymbol{X_1}, \boldsymbol{p_2})$ are given, PnP can be deployed for solving the camera pose by minimizing the reprojection error,

$$e = \sum_i ||\boldsymbol{K}(\boldsymbol{R}\boldsymbol{X}_{(1,i)} + \boldsymbol{t}) - \boldsymbol{p}_{(2,i)}||_2 \qquad (2)$$

In order to establish the 3D-2D correspondences, we need to (1) estimate the 3D scene structure, (2) match 3D points to 2D pixels by matching features.

B. Deep Predictions for Correspondence

Suppose we have two deep models, M_d and M_f, for single-view depth estimation and two-view optical flow estimation respectively. Given an image pair, (I_i, I_{i-1}), M_f is employed to estimate the dense optical flow, which gives 2D-2D correspondences. However, the CNN optical flows are not accurate for all the pixels. The accuracy of the 2D-2D correspondences is significant for estimating accurate relative pose. In order to filter out the optical flow outliers, we estimate both forward and backward optical flows, $\boldsymbol{F}_{i-1}^{i}$ and $\boldsymbol{F}_{i}^{i-1}$, and use the forward-backward flow inconsistency $|-\boldsymbol{F}_{i-1}^{i} - \boldsymbol{F}_{i}^{i-1}|$ as a measure to choose good 2D-2D correspondences. Note that $\boldsymbol{F}_{i}^{i-1}$ is in fact the respective backward flow instead of the CNN backward flow prediction. We choose optical flows with the least flow inconsistency $\boldsymbol{F'}$ to form the best-*N* 2D-2D matches, $(\boldsymbol{P}_i, \boldsymbol{P}_{i-1})$, for estimating the relative camera pose. Comparing to traditional feature-based method, which only use salient feature points for matching and tracking, any pixel in the dense optical flow can be a candidate for tracking. Moreover, traditional features usually gather visual information from local regions while CNN gathers more visual information (large receptive field) and higher level contextual information, which gives more accurate and robust estimations. We can further estimate 3D structure w.r.t view-*i* by estimating single-view depths from M_d. Knowing the 2D-2D matches and the 3D positions of view-*i*, 3D-2D correspondences can be built.

C. Deep Predictions for Visual Odometry

Given a depth CNN prediction $(\boldsymbol{D}_i)$ and $(\boldsymbol{P}_i, \boldsymbol{P}_{i-1})$, 2D-2D and 3D-2D correspondences can be established. We can solve the relative camera pose either by solving PnP (3D-2D) or essential matrix (2D-2D). Unfortunately, the current state-of-the-art (SOTA) single-view depth estimation methods are still insufficient for recovering accurate 3D structure for accurate camera pose estimation, which is shown in Table II.

On the other hand, optical flow estimation is a more generic task and the current SOTA deep learning methods are accurate and with good generalization ability. Therefore, we can use the 2D-2D matches for solving essential matrix and recovering camera motion instead. Nevertheless, as we mentioned in Sec.III-A, solving essential matrix from two-view carries some well-known issues. We show that we can avoid/resolve the issues by leveraging deep predictions.

a) Scale ambiguity: Translation recovered from essential matrix is up-to-scale. However, we can use the CNN depths $\boldsymbol{D}_i$ as a reference for scale recovery. First, $[\boldsymbol{R}, \hat{\mathbf{t}}]$ are recovered from solving essential matrix. Then, triangulation is performed for $(\boldsymbol{P}_i, \boldsymbol{P}_{i-1})$ to recover up-to-scale depths $\boldsymbol{D}'_i$. A scaling factor can be estimated by comparing $(\boldsymbol{D}_i, \boldsymbol{D}'_i)$.

b) Unsolvable/Unstable solution: Pure rotation causes unsolvable essential matrix while small camera motion is an ill-conditioned issue. In order to avoid these two issues, we apply two condition checks. (1) *Average flow magnitude*: we solve essential matrix only when the average flow magnitude is larger than a certain threshold,$\delta_f = 5$, which avoids small camera motion which comes with small optical flows. (2) *Check for cheirality condition*: There are 4 possible solutions for $[\boldsymbol{R}, \hat{\mathbf{t}}]$ by decomposing $\boldsymbol{E}$. In order to find the correct unique solution, cheirality condition, i.e. the triangulated 3D points must be in front of both cameras, is checked to remove the other solutions. We further use the number of points satisfying cheirality condition as a reference to determine if the solution is stable.

If the above conditions are not satisfied, the 3D-2D correspondences are used to estimate the camera motion by solving PnP instead. To increase the robustness of the proposed pipeline, we wrap the following steps in RANSAC loops, including essential matrix estimation, scaling factor estimation, and PnP.

IV. CNN Training framework

In this section, we present the deep learning networks and the training frameworks for learning depths and optical flows.

A. Single-View Depth Network

Depending on the availability of data (monocular/stereo videos, ground truth depths), different depth models can be trained. Throughout the work, we use a standard fully convolutional encoder-decoder network with skip-connections [38] [39] to predict depths.

The most trivial way of training the depth model is using supervised learning [40]–[46] but ground truth depths are not always available for different application scenarios. Recent works have suggested that learning single-view depths (and camera motions) in a self-supervised manner is possible using monocular sequences [4], [8] or stereo sequences [47] [48] [5] [39]. Instead of using ground truth supervisions, the main supervision signal in the self-supervised frameworks is photometric consistency across multiple-views.

In this work, we follow [39] for training depth models using monocular sequences and stereo sequences. For self-supervised training, we jointly train the depth network and a pose network by minimizing the mean of the following *per-pixel* objective function over the whole image. The *per-pixel* loss is

$$\begin{aligned} L = &\min_j L_{pe}(I_i, I_j^i) + \lambda_{ds} L_{ds}(D_i, I_i) + \\ &\min_j \lambda_{dc} L_{dc}(D_i, D_j^i), \end{aligned} \qquad (3)$$

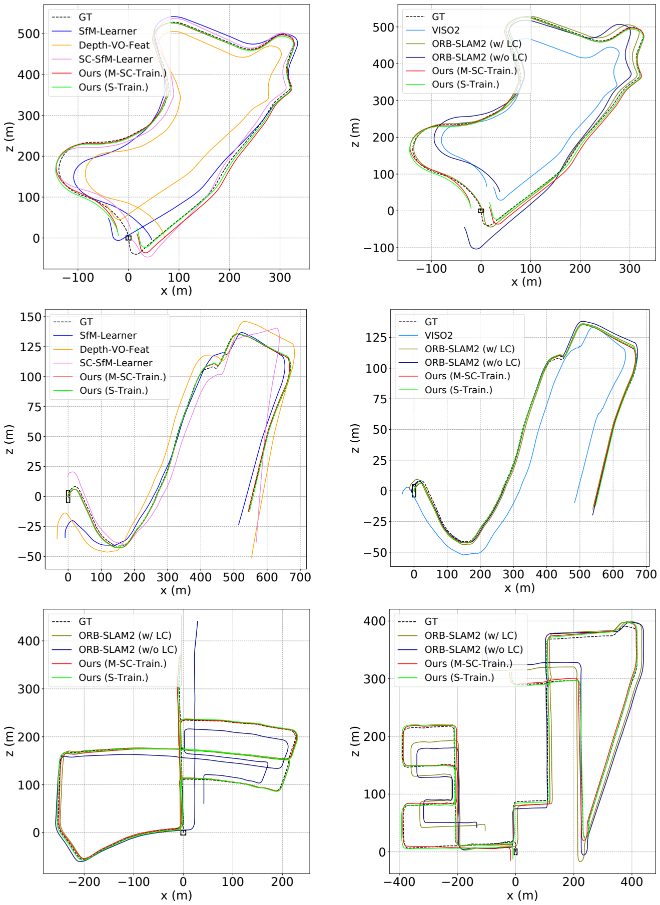

Fig. 2. Qualitative VO results on KITTI: (Top) Seq.09 and (Middle) Seq.10 against deep learning-based and geometry-based methods (shown separately) and (Bottom) Seq.05 and Seq.08 against ORB-SLAM2.

where $[\lambda_{ds}, \lambda_{dc}]$ are loss weightings. L_{pe} is the photometric error by computing the difference between the reference image I_i and the synthesized view I^i_j from the source image I_j, where $j \in [i-n, i+n, s]$. $[i-n, i+n]$ are neighbouring views of I_i while s is stereo pair if stereo sequences are used in training. As proposed in [39], instead of averaging the photometric error over all source images, [39] applies minimum to overcome the issues related to out-of-view pixels and occlusions.

$$L_{pe}(I_i, I^i_j) = \frac{\alpha}{2}\left(1 - \mathrm{SSIM}(I_i, I^i_j)\right) + (1-\alpha)|I_i - I^i_j| \quad (4)$$

$$I^i_j = f_w(I_j, h_1(K, D_i, T^j_i)), \quad (5)$$

where $\alpha = 0.85$, $f_w(.)$ is a warping function. K is the camera intrinsics. D_i and T^i_j are predicted from depth network and pose network respectively. $h_1(.)$ establish the reprojection from view-*i* to view-*j*.

L_{ds} is an edge-aware depth smoothness for regularization.

$$L_{ds}(D_i, I_i) = |\partial_x D_i| e^{-|\partial_x I_i|} + |\partial_y D_i| e^{-|\partial_y I_i|}, \quad (6)$$

Similar to traditional monocular 3D reconstruction, scale ambiguity and scale inconsistency issues exist when monocular videos are used for training. Since the monocular training usually uses image snippets for training, it does not guarantee a consistent scale is used across snippets which creates scale inconsistency issue. One solution is to use stereo sequences during training [29] [5] [39], the deep predictions are aligned with real-world scale and scale consistent because of the constraint introduced by the known stereo baseline. Even though stereo sequences are used during training, only monocular images are required during inference. Another possible solution to overcome the scale inconsistency issue is using temporal geometry consistency regularization proposed in [49] [8], which constrains the depth consistency across multiple views. As depth predictions are consistent across different views, the scale inconsistency issue is resolved. Using the rigid scene assumption as the cameras move in space over time we want the predicted depths at view-*i* to be consistent with the respective predictions at view-*j*. This is done by **correctly transforming** the scene geometry from frame-j to frame-i much like the image warping. Specifically, we adopt the inverse depth consistency proposed in [49].

$$L_{dc}(D_i, D^i_j) = |1/D_i - 1/D^i_j| \quad (7)$$

Inspired by [39], instead of averaging the depth consistency error over all source views, we use minimum to avoid occlusions and out-of-view scenes.

B. *Optical Flow Network*

Many state-of-the-art deep learning based methods for estimating optical flow have been proposed [50]–[54]. In this work, we choose LiteFlowNet [52] as our backbone for M_f as it is fast, lightweight, accurate and it has good generalization ability. LiteFlowNet consists of a two-stream network for feature extraction and a cascaded network for flow inference and regularization. We refer readers to [52] for more details. Even though LiteFlowNet [52] is trained with synthetic data (Scene Flow) [50], it shows good generalization ability on real data. In this work, we mainly use the LiteFlowNet model trained in Scene Flow. However, we also show that a self-supervised training/finetuning can be performed which helps the model better adapt to unseen environments. Similar to the self-supervised training of the depth network, the optical flow network is trained by minimizing the mean of the following *per-pixel* loss function over the whole image,

$$\begin{aligned} L = \min_j L_{pe}(I_i, I^i_j) + \lambda_{fs} L_{fs}(\|F^j_i\|_2, I_i) \\ + \lambda_{fc} L_{fc}(|-F^i_j - F^j_i|) \quad (8) \\ I^i_j = f_w(I_j, h_2(F^j_i)), \quad (9) \end{aligned}$$

Different from Eqn. 5, $h_2(.)$ takes optical flow and establish the correspondences between view-*i* and view-*j*. We also regularize the optical flow to be smooth using an edge-aware flow smoothness loss $L_{fs}(.)$. Similar to Meister *et al.* [54], we estimate both forward and backward optical flow and constrain the bidirectional predictions to be consistent with the loss L_{fc}.

TABLE I

QUANTITATIVE RESULT ON KITTI ODOMETRY SEQ. 00-10. THE BEST RESULT IS IN BOLD AND SECOND BEST IS UNDERLINED.

Method	Metric	00	01	02	03	04	05	06	07	08	09	10	Avg. Err.
SfM-Learner (updated ver.) [4]	t_{err}	21.32	**22.41**	24.10	12.56	4.32	12.99	15.55	12.61	10.66	11.32	15.25	14.068
	r_{err}	6.19	2.79	4.18	4.52	3.28	4.66	5.58	6.31	3.75	4.07	4.06	4.660
	ATE	104.87	$\underline{109.61}$	185.43	8.42	3.10	60.89	52.19	20.12	30.97	26.93	24.09	51.701
	RPE (m)	0.282	$\underline{0.660}$	0.365	0.077	0.125	0.158	0.151	0.081	0.122	0.103	0.118	0.158
	RPE (°)	0.227	0.133	0.172	0.158	0.108	0.153	0.119	0.181	0.152	0.159	0.171	0.160
Depth-VO-Feat [5]	t_{err}	6.23	$\underline{23.78}$	6.59	15.76	3.14	4.94	5.80	6.49	5.45	11.89	12.82	7.911
	r_{err}	2.44	1.75	2.26	10.62	2.02	2.34	2.06	3.56	2.39	3.60	3.41	3.470
	ATE	64.45	203.44	85.13	21.34	3.12	22.15	14.31	15.35	29.53	52.12	24.70	33.220
	RPE (m)	0.084	**0.547**	0.087	0.168	0.095	0.077	0.079	0.081	0.084	0.164	0.159	0.108
	RPE (°)	0.202	0.133	0.177	0.308	0.120	0.156	0.131	0.176	0.180	0.233	0.246	0.193
SC-SfMLearner [8]	t_{err}	11.01	27.09	6.74	9.22	4.22	6.70	5.36	8.29	8.11	7.64	10.74	7.803
	r_{err}	3.39	1.31	1.96	4.93	2.01	2.38	1.65	4.53	2.61	2.19	4.58	3.023
	ATE	93.04	**85.90**	70.37	10.21	2.97	40.56	12.56	21.01	56.15	15.02	20.19	34.208
	RPE (m)	0.139	0.888	0.092	0.059	0.073	0.070	0.069	0.075	0.085	0.095	0.105	0.086
	RPE (°)	0.129	**0.075**	0.087	0.068	$\underline{0.055}$	0.069	0.066	0.074	0.074	0.102	0.107	0.083
VISO2 [18]	t_{err}	10.53	61.36	18.71	30.21	34.05	13.16	17.69	10.80	13.85	18.06	26.10	19.316
	r_{err}	2.73	7.68	1.19	2.21	1.78	3.65	1.93	4.67	2.52	1.25	3.26	2.519
	ATE	79.24	494.60	70.13	52.36	38.33	66.75	40.72	18.32	61.49	52.62	57.25	53.721
	RPE (m)	0.221	1.413	0.318	0.226	0.496	0.213	0.343	0.191	0.234	0.284	0.442	0.297
	RPE (°)	0.141	0.432	0.108	0.157	0.103	0.131	0.118	0.176	0.128	0.125	0.154	0.134
DSO [19] (from [31])	ATE	113.18	/	116.81	1.39	**0.42**	47.46	55.62	16.72	111.08	52.23	11.09	52.600
ORB-SLAM2 (w/o LC) [16]	t_{err}	11.43	107.57	10.34	$\underline{0.97}$	$\underline{1.30}$	9.04	14.56	9.77	11.46	9.30	2.57	8.074
	r_{err}	$\underline{0.58}$	$\underline{0.89}$	**0.26**	**0.19**	$\underline{0.27}$	$\underline{0.26}$	$\underline{0.26}$	0.36	**0.28**	$\underline{0.26}$	0.32	$\underline{0.304}$
	ATE	40.65	502.20	47.82	**0.94**	1.30	29.95	40.82	16.04	43.09	38.77	5.42	26.480
	RPE (m)	0.169	2.970	0.172	0.031	0.078	0.140	0.237	0.105	0.192	0.128	**0.045**	0.130
	RPE (°)	0.079	0.098	$\underline{0.072}$	0.055	0.079	$\underline{0.058}$	0.055	$\underline{0.047}$	0.061	$\underline{0.061}$	0.065	$\underline{0.063}$
ORB-SLAM2 (w/ LC) [16]	t_{err}	2.35	109.10	$\underline{3.32}$	**0.91**	1.56	1.84	$\underline{4.99}$	1.91	9.41	2.88	3.30	3.247
	r_{err}	**0.35**	**0.45**	$\underline{0.31}$	**0.19**	$\underline{0.27}$	**0.20**	**0.23**	$\underline{0.28}$	$\underline{0.30}$	**0.25**	**0.30**	**0.268**
	ATE	**6.03**	508.34	**14.76**	$\underline{1.02}$	1.57	4.04	11.16	2.19	38.85	**8.39**	6.63	9.464
	RPE (m)	0.206	3.042	0.221	0.038	0.081	0.294	0.734	0.510	0.162	0.343	$\underline{0.047}$	0.264
	RPE (°)	0.090	$\underline{0.087}$	0.079	0.055	0.076	0.059	$\underline{0.053}$	0.050	0.065	0.063	0.066	0.066
CNN-SVO [31] (from [31])	ATE	17.53	/	50.52	3.46	2.44	8.15	11.51	6.51	10.98	$\underline{10.69}$	4.84	12.663
Ours (Mono-SC Train.)	t_{err}	$\underline{2.25}$	66.98	3.60	2.67	1.43	$\underline{1.15}$	**1.03**	**0.93**	$\underline{2.23}$	**2.47**	**1.96**	$\underline{1.972}$
	r_{err}	$\underline{0.58}$	17.04	0.52	0.50	0.29	0.30	$\underline{0.26}$	0.29	$\underline{0.30}$	0.30	$\underline{0.31}$	0.365
	ATE	12.64	695.75	23.11	1.23	1.36	$\underline{3.75}$	$\underline{2.63}$	$\underline{1.74}$	$\underline{7.87}$	11.02	**3.37**	$\underline{6.872}$
	RPE (m)	$\underline{0.040}$	1.281	$\underline{0.063}$	$\underline{0.030}$	$\underline{0.057}$	$\underline{0.025}$	$\underline{0.033}$	$\underline{0.023}$	$\underline{0.042}$	**0.055**	$\underline{0.047}$	$\underline{0.041}$
	RPE (°)	$\underline{0.056}$	0.725	**0.046**	$\underline{0.038}$	**0.030**	**0.035**	**0.029**	**0.030**	**0.036**	**0.037**	**0.042**	**0.038**
Ours (Stereo Train.)	t_{err}	**1.96**	56.76	**2.38**	2.49	**1.03**	**1.10**	**1.03**	$\underline{0.97}$	**1.60**	$\underline{2.61}$	$\underline{2.29}$	**1.746**
	r_{err}	0.60	13.93	0.55	$\underline{0.39}$	**0.25**	0.30	0.30	**0.27**	0.32	0.29	0.37	0.364
	ATE	$\underline{11.34}$	484.86	$\underline{21.16}$	2.04	$\underline{0.86}$	**3.63**	**2.53**	**1.72**	**5.66**	10.88	$\underline{3.72}$	**6.354**
	RPE (m)	**0.027**	1.203	**0.033**	**0.023**	**0.036**	**0.020**	**0.024**	**0.018**	**0.032**	$\underline{0.056}$	$\underline{0.047}$	**0.032**
	RPE (°)	**0.055**	0.773	**0.046**	**0.037**	**0.030**	**0.035**	**0.029**	**0.030**	$\underline{0.037}$	**0.037**	$\underline{0.043}$	**0.038**

V. EXPERIMENTS AND EVALUATIONS

In this section we describe the details of the experimental evaluation of our method. Due to space limitation, we mainly analyze our VO system on KITTI dataset [55], [56] and present indoor results in supplementary video. We compare our system with prior arts on VO estimation. Additionally, we perform a detailed analysis to show the effect of different factors affecting the VO performance.

KITTI Odometry dataset contains 11 driving sequences (stereo sequences) with publicly available ground truth camera poses. Following [4], we train our depth network and finetune our flow network on sequences 00-08. The dataset contains 36,671 training pairs, $[I_i, I_{i-1}, I_{i+1}, I_{i,s}]$.

A. Implementation Details

We train our network with the PyTorch [57] framework. All self-supervised experiments are trained using Adam optimizer [58] for 20 epochs. For KITTI, images with size 640x192 are used for training. Learning rate is set to 10^{-4} for the first 15 epochs and then is dropped to 10^{-5} for the remainder. The loss weightings are $[\lambda_{ds}, \lambda_{dc}] = [10^{-3}, 5]$ and $[\lambda_{fs}, \lambda_{fc}] = [10^{-1}, 5 \times 10^{-3}]$ for single-view depth and optical flow experiments respectively.

B. Visual Odometry Evaluation

We compare our VO system with pure deep learning methods [4] [5] [8], geometry-based methods including DSO [19], VISO2 [18], ORB-SLAM2 [59] (w/ and w/o loop-closure) and CNN-SVO [31]. The quantitative and qualitative results are shown in Table I and Fig. 2. ORB-SLAM2 occasionally suffers from tracking failure or unsuccessful initialization. We run ORB-SLAM2 three times and choose the run with the least trajectory error. We adopt common evaluation criteria for a more detailed analysis. KITTI Odometry criterion evaluates possible sub-sequences of length (100, 200, ..., 800) meters and report the average translational error $t_{err}(\%)$ and rotational errors $r_{err}(°/100m)$. Absolute trajectory error (ATE) measures the root-mean-square error between predicted camera poses $[x, y, z]$ and ground truth. Relative pose error (RPE) measures frame-to-frame relative pose error. Since most of the methods are monocular method, which lacks a scaling factor to match with real-world scale, we scale and align (7DoF optimization) the predictions to the ground truth associated poses during evaluation by minimizing ATE [60]. Except for methods using stereo depth models (*Ours (Stereo Train.)*, *Depth-VO-Feat*) and known scale prior (VISO2), which already align predictions to real-world scale,

TABLE II

Ablation study: Effect on VO performance by changing different components of the Reference Model.

Experiment	09				10			
	t_{err}	r_{err}	RPE(m)	RPE (°)	t_{err}	r_{err}	RPE(m)	RPE (°)
PnP only	7.12	2.43	0.082	0.081	6.83	3.88	0.058	0.093
Depth model (Mono.)	4.84	0.66	0.105	0.058	4.02	1.12	0.071	0.061
Depth model (Mono-SC)	4.09	0.68	0.087	0.057	3.02	0.97	0.063	0.062
Self-LiteFlowNet	2.97	0.64	0.048	0.047	2.87	0.77	0.047	0.053
Uniform sampling	5.86	1.26	0.084	0.055	5.81	1.99	0.054	0.061
Reference Model (Sec.V-C)	4.56	0.62	0.082	0.056	3.53	1.14	0.054	0.060
Full Image Res.	**2.61**	**0.29**	**0.056**	**0.037**	**2.29**	**0.37**	**0.047**	**0.043**

for fair comparison, we perform 6DoF optimization w.r.t ATE instead. We exclude Seq.01 while computing average error since a sub-sequence of Seq.01 does not contain trackable close features and most methods fail in this sequence. In Table I, *Ours (Mono-SC Train.)* uses a depth model trained with monocular videos and inverse depth consistency for ensuring scale-consistency. *Ours (Stereo Train.)* uses a depth model trained with stereo sequences. Note that even stereo sequences are used in training, monocular sequences are used in testing. Therefore, *Ours (Stereo Train.)* is still a monocular VO system. We show that our method outperforms pure deep learning methods by a large margin in all metrics. However, one interesting result for deep learning methods is that, all methods failed in Seq.01 except the deep models, which shows that deep models can be used as a complement to geometry methods with careful design in the future to overcome the failure mode for geometry methods. For long sequence evaluation, ORB-SLAM2 shows less rotation drift r_{err} but higher translation drift t_{err} due to scale drift issue which is showed in Fig. 2. Sometimes the issue can be resolved by loop closing. We use scale consistent depth predictions for scale recovery which mitigates the issue in most monocular VO/SLAM systems. As a result, our method shows less translation drift over long sequences. More importantly, our method shows consistently smaller RPE which allows our system to be a robust module for frame-to-frame tracking.

C. Ablation Study

Table II shows an ablation study on our VO system. We evaluate the effect of different components in our system by changing a variable in the *Reference Model* consisting of:

- Algorithm: Full algorithm shown in Alg. 1
- Depth model: Trained with stereo sequences
- Flow model: LiteFlowNet trained from synthetic dataset
- Matches: Top-N matches with least flow inconsistency
- Image resolution: downsampled size (640×192)

a) Algorithm: We compare the full algorithm with the algorithm using PnP only. Since the single-view depth predictions are not accurate enough to establish accurate 3D-2D matches, only using PnP performs worse than the full algorithm, which relies more on 2D-2D matches.

b) Depth model: We train three depth models with monocular or stereo sequences. For monocular experiments, depth models with and without depth consistency term, $\lambda_{dc} = [5, 0]$, are trained and used for depth estimation. The result shows that with scale-consistent depth predictions, *Mono-SC* model performs on par with the model trained with stereo sequences. However, both monocular models still have the scale ambiguity issue while *Stereo Train.* does not.

c) Flow model: Even LiteFlowNet is trained with synthetic data but it still shows good generalization ability from synthetic to real. With self-supervised finetuning, the model adapts to driving sequences better and gives better VO result.

d) Matches: Forward-backward flow consistency as a measure to choose good 2D-2D matches is proposed. To prove the effectiveness of the the flow consistency, we uniformly sample matches from the optical flows and compare the result against that using flow consistency.

e) Image resolution: At inference time, simply increasing the image size to full resolution, the optical flow network predicts more accurate correspondences which helps the relative pose estimation.

VI. Conclusion

In this paper, we have presented a robust monocular VO system leveraging deep learning and geometry methods. We explore the integration of deep predictions with classic geometry methods. Specifically, we use optical flow and single-view depth predictions from deep networks as an intermediate outputs to establish 2D-2D/3D-2D correspondences for camera pose estimation. We show that the deep models can be trained in a self-supervised manner and we explore the effect of different training schemes. Depth models with consistent scale can be used for scale recovery, which mitigates the scale drift issue in most monocular VO/SLAM systems. Instead of learning a complete VO system in an end-to-end manner, which does not perform competitively to geometry-based methods, we think that (1) integrating deep predictions with geometry-based methods can gain the best from both domains; (2) deep VO models can be used as a complement in standard VO/SLAM system when they fail. A future work will be turning the proposed system into a map-to-frame tracking system to further improve the tracking performance.

VII. Acknowledgement

This work was supported by the UoA Scholarship to HZ, the ARC Laureate Fellowship FL130100102 to IR and the Australian Centre of Excellence for Robotic Vision CE140100016.

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal on Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[2] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Computer Vision (ICCV), 2011 IEEE international conference on*. IEEE, 2011, pp. 2564–2571.

[3] J. Bian, W.-Y. Lin, Y. Liu, L. Zhang, S.-K. Yeung, M.-M. Cheng, and I. Reid, "GMS: Grid-based motion statistics for fast, ultra-robust feature correspondence," *International Journal on Computer Vision (IJCV)*, 2019.

[4] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] H. Zhan, R. Garg, C. S. Weerasekera, K. Li, H. Agarwal, and I. Reid, "Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018, pp. 340–349.

[6] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1983–1992.

[7] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, and M. J. Black, "Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 240–12 249.

[8] J.-W. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid, "Unsupervised scale-consistent depth and ego-motion learning from monocular video," in *Neural Information Processing Systems (NeurIPS)*, 2019.

[9] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 2938–2946.

[10] E. Brachmann, A. Krull, S. Nowozin, J. Shotton, F. Michel, S. Gumhold, and C. Rother, "Dsac-differentiable ransac for camera localization," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6684–6692.

[11] E. Brachmann and C. Rother, "Learning less is more-6d camera localization via 3d surface regression," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4654–4662.

[12] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge University Press, 2003.

[13] D. Scaramuzza and F. Fraundorfer, "Visual odometry: Part i: The first 30 years and fundamentals," *IEEE Robotics & Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[14] S. Ullman, "The interpretation of structure from motion," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 203, no. 1153, pp. 405–426, 1979.

[15] L. Matthies, M. Maimone, A. Johnson, Y. Cheng, R. Willson, C. Villalpando, S. Goldberg, A. Huertas, A. Stein, and A. Angelova, "Computer vision on mars," *International Journal on Computer Vision (IJCV)*, vol. 75, no. 1, pp. 67–92, 2007.

[16] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *CoRR*, vol. abs/1610.06475, 2016.

[17] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*. IEEE, 2007, pp. 225–234.

[18] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV)*, 2011.

[19] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)*, 2017.

[20] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2320–2327.

[21] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 15–22.

[22] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics (TRO)*, vol. 33, no. 2, pp. 249–265, 2016.

[23] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular slam," in *European Conference on Computer Vision (ECCV)*. Springer, 2014, pp. 834–849.

[24] P. Agrawal, J. Carreira, and J. Malik, "Learning to see by moving," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 37–45.

[25] S. Wang, R. Clark, H. Wen, and N. Trigoni, "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2043–2050.

[26] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[27] H. Zhou, B. Ummenhofer, and T. Brox, "Deeptam: Deep tracking and mapping," *arXiv preprint arXiv:1808.01900*, 2018.

[28] T. Dharmasiri, A. Spek, and T. Drummond, "Eng: End-to-end neural geometry for robust depth and pose estimation using cnns," *arXiv preprint arXiv:1807.05705*, 2018.

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

[30] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-time dense monocular slam with learned depth prediction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6243–6252.

[31] 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," *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[32] N. Yang, R. Wang, J. Stueckler, and D. Cremers, "Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry," in *European Conference on Computer Vision (ECCV)*, 2018.

[33] Y. Li, Y. Ushiku, and T. Harada, "Pose graph optimization for unsupervised monocular visual odometry," *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[34] D. Nister, "An efficient solution to the five-point relative pose problem," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003, pp. II–195.

[35] Z. Zhang, "Determining the epipolar geometry and its uncertainty: A review," *International Journal on Computer Vision (IJCV)*, vol. 27, no. 2, pp. 161–195, 1998.

[36] R. I. Hartley, "In defence of the 8-point algorithm," in *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 1995, pp. 1064–1070.

[37] J.-W. Bian, Y.-H. Wu, J. Zhao, Y. Liu, L. Zhang, M.-M. Cheng, and I. Reid, "An evaluation of feature matchers for fundamental matrix estimation," in *British Machine Vision Conference (BMVC)*, 2019.

[38] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[39] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth prediction," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[40] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Neural Information Processing Systems (NeurIPS)*, 2014, pp. 2366–2374.

[41] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5162–5170.

[42] F. Liu, C. Shen, G. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)*, vol. 38, no. 10, pp. 2024–2039, 2016.

[43] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 239–248.

[44] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5580–5590.

[45] V. Nekrasov, T. Dharmasiri, A. Spek, T. Drummond, C. Shen, and I. Reid, "Real-time joint semantic segmentation and depth estimation using asymmetric annotations," *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[46] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao, "Deep ordinal regression network for monocular depth estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 2002–2011.

[47] R. Garg, V. K. B G, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 740–756.

[48] C. Godard, O. Mac Aodha, and G. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 6602–6611.

[49] H. Zhan, C. S. Weerasekera, R. Garg, and I. D. Reid, "Self-supervised learning for single view depth and surface normal estimation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 4811–4817.

[50] 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 *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2758–2766.

[51] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2462–2470.

[52] T.-W. Hui, X. Tang, and C. C. Loy, "Liteflownet: A lightweight convolutional neural network for optical flow estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 8981–8989.

[53] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8934–8943.

[54] S. Meister, J. Hur, and S. Roth, "Unflow: Unsupervised learning of optical flow with a bidirectional census loss," in *Association for the Advancement of Artificial Intelligence (AAAI)*, 2018.

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

[56] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[57] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS Autodiff Workshop*, 2017.

[58] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[59] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics (TRO)*, vol. 31, no. 5, pp. 1147–1163, 2015.

[60] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Recognition and Machine Intelligence (TPAMI)*, no. 4, pp. 376–380, 1991.