Feature-Level Collaboration: Joint Unsupervised Learning of Optical Flow, Stereo Depth and Camera Motion

Cheng Chi[1], Qingjie Wang[1], Tianyu Hao[1], Peng Guo[1], Xin Yang[1*]
[1]Huazhong University of Science and Technology
{chengchi2019, wqj, hty, guopeng, xinyang2014} @hust.edu.cn

Abstract

Precise estimation of optical flow, stereo depth and camera motion are important for the real-world 3D scene understanding and visual perception. Since the three tasks are tightly coupled with the inherent 3D geometric constraints, current studies have demonstrated that the three tasks can be improved through jointly optimizing geometric loss functions of several individual networks. In this paper, we show that effective feature-level collaboration of the networks for the three respective tasks could achieve much greater performance improvement for all three tasks than only loss-level joint optimization. Specifically, we propose a single network to combine and improve the three tasks. The network extracts the features of two consecutive stereo images, and simultaneously estimates optical flow, stereo depth and camera motion. The whole network mainly contains four parts: (I) a feature-sharing encoder to extract features of input images, which can enhance features' representation ability; (II) a pooled decoder to estimate both optical flow and stereo depth; (III) a camera pose estimation module which fuses optical flow and stereo depth information; (IV) a cost volume complement module to improve the performance of optical flow in static and occluded regions. Our method achieves state-of-the-art performance among the joint unsupervised methods, including optical flow and stereo depth estimation on KITTI 2012 and 2015 benchmarks, and camera motion estimation on KITTI VO dataset.

1. Introduction

Optical flow, depth and camera motion estimation are three fundamental tasks in the field of computer vision. Deep learning methods have greatly advanced the state-of-the-art in optical flow and stereo depth estimation [32, 13, 7]. Meanwhile, learning-based camera ego-motion prediction [53, 52] has also made significant progress recently. Jointly estimating optical flow, stereo depth and camera motion can be applied in a wide range of applications, such as autonomous navigation [9, 42], 3D scene reconstruction [38] and robot control [1]. Many unified unsupervised framework [6, 44, 45, 27, 2] have been proposed to jointly optimize two or three tasks concurrently. These joint methods demonstrate that jointly tackling these tasks has a positive effect on each of them.

There is a tight geometric relationship among optical flow, stereo depth and camera pose, due to that each one of the three tasks can be calculated by the other two. Therefore, previous joint methods [4, 30, 21, 45] usually estimate them by several individual networks, and construct various geometric consistency constraint losses to mutually guide each other. Recently, some works [24, 27] have tried to share the same network for stereo depth and optical flow estimation. However, based on the epipolar constraint, the stereo depth estimation network only needs to search pixel correspondences in horizontal lines, while the optical flow estimation network demands a more comprehensive search in both horizontal and vertical directions. So these methods only treat the stereo depth and optical flow estimation as exactly the same task, but fail to allow full play the advantage of sharing features between stereo depth and optical flow estimation. That is, most if not all, existing joint methods [4, 30, 45, 21, 24, 27] do not take full advantage of the feature-level collaboration to constrain each other in the learning process of the three tasks.

In this paper, we demonstrate that effective feature-level collaboration for the three respective tasks could achieve much greater performance improvement for all three tasks than loss-level joint optimization. The intuition behind this idea is that both stereo depth and optical flow estimation networks find pixel correspondences between two images, sharing features between the two tasks is reasonable and meanwhile constrain the training process of the two tasks. In addition, camera motion can be directly calculated by the optical flow and stereo depth, so utilizing the feature-level information of optical flow and stereo depth could add further geometric constraint for feature training. To this

* represents the corresponding author

end, we design a single network to integrate all the three tasks. We obtain the features of stereo images and consecutive frames by a feature-sharing encoder, and then take full advantage of the features to predict both optical flow and stereo depth by a pooled decoder. After that, the extracted image features are used to predict the camera motion. In this way, we achieve the feature-level mutual leaning of the three tasks. And to our best knowledge, our method is the first feature-level collaboration of the three tasks. And our experiments verify that the collaboration at feature level can significantly improve the performance of the three tasks.

Occlusion is also a primary problem in unsupervised optical flow estimation, because the occluded pixels in the former frame can not find the corresponding matching pixels in the next frame. In terms of image, the occluded pixels do not obey photometric consistency hypothesis. Several methods [27, 44, 40, 23] utilize various geometric loss terms instead of photometric loss to guide the optical flow estimation of the occluded pixels. In terms of features, the cost volume stores the matching costs of corresponding pixels in different images, so it is obvious that the part of the cost volume corresponding to the occlusion pixels is inaccurate, which will degrade the performance of the network. However, existing joint unsupervised methods hardly handle the inaccurate cost volume. In this study, we find that in the real-world scenes, there are fewer occluded pixels between the left and right image captured by the stereo camera. Based on the fact, we propose to leverage the cost volume of stereo images to complement the cost volume of two consecutive frames, which can achieve a better performance on optical flow estimation in occluded regions.

In summary, we propose a single unsupervised network to jointly estimate optical flow, stereo depth and camera motion, and achieve feature-level collaboration of the three tasks. Our main contributions include: (I) We use a feature-sharing encoder for optical flow, stereo depth and camera motion estimation; (II) We design a pooled decoder to estimate both the optical flow and stereo depth; (III) We propose a novel method to estimate camera motion by using image features which are shared with optical flow and stereo depth. Furthermore, we also design a pose refinement module to further improve the camera motion accuracy; (IV) We explore the cost volumes complementary method to solve the occlusion problem at feature level. (V) Our method outperforms existing unsupervised joint methods. Remarkably, our method in optical flow estimation even outperforms some classic supervised methods [43, 18].

2. Related work

In this section, we introduce some deep learning methods which are closely related to our work.

Depth estimation. DispNetC [32] is the first deep-learning method to estimate stereo depth, which utilizes the cost volume to assist the estimation. After that, many supervised stereo depth estimation methods [3, 13, 5] make a 4D feature cost volume and incorporate 3D convolution for further regularization. These supervised methods directly predict depth from CNNs framework by minimizing the difference between the predicted depth and the ground truth. Due to the scarcity of data with ground truth, [11, 51] propose unsupervised depth estimation methods by minimizing photometric error.

Optical flow estimation. Starting from FlowNet [7], various supervised [39, 43, 15] learning architectures of optical flow estimation have been proposed. These supervised methods significantly improve the optical flow prediction by extracting feature pyramid and constructing cost volumes at different scales. However, it is difficult to obtain the ground truth of optical flow in the real-world scenes. So recent works [20, 33, 47] propose unsupervised optical flow estimation methods based on the photometric consistency assumption and spatial smoothness assumption. However, the pixels in occluded regions do not obey the brightness constancy and greatly decrease the whole performance. Some works address this problem by detecting occlusion and then excluding occluded pixels when computing photometric [46, 19] constancy loss, or using the teacher-student framework to provide more accurate constraint for occluded pixels [26, 28]. These methods have definitely obtained improvement by constructing more effective losses, but ignore the degraded cost volumes.

Joint unsupervised depth and camera pose estimation. Zhou et al. [53] shows that it is possible to simultaneously predict monocular depth and camera motion between two consecutive video frames. After that, some works follow the idea by constructing a 3D-based geometric consistency loss [31] or utilizing semantic segmentation to estimate the individual object motion to handle dynamic scenes [6]. [12] propose a per-pixel minimum reprojection loss to handle occlusions. These unsupervised methods use geometric consistency to jointly predict monocular depth and camera pose or object motions. However, due to the limitation of monocular cameras, their performance have a large gap with stereo depth estimation methods.

Joint unsupervised stereo matching and optical flow estimation. Due to both stereo matching and optical flow estimation can be modeled as the same problem to find matching correspondence between two images, [24] firstly use the same network to predict optical flow or stereo matching, and introduces geometric constraints to guide the joint estimation of the two tasks. Based on [24], [27] further employs a Teacher-Student network to guide the occluded regions. However, in order to use a single network for optical flow and depth estimation, [24, 27] estimate a two-dimensional disparity and then set the vertical dimension to be zero, which may add extra error for the disparity

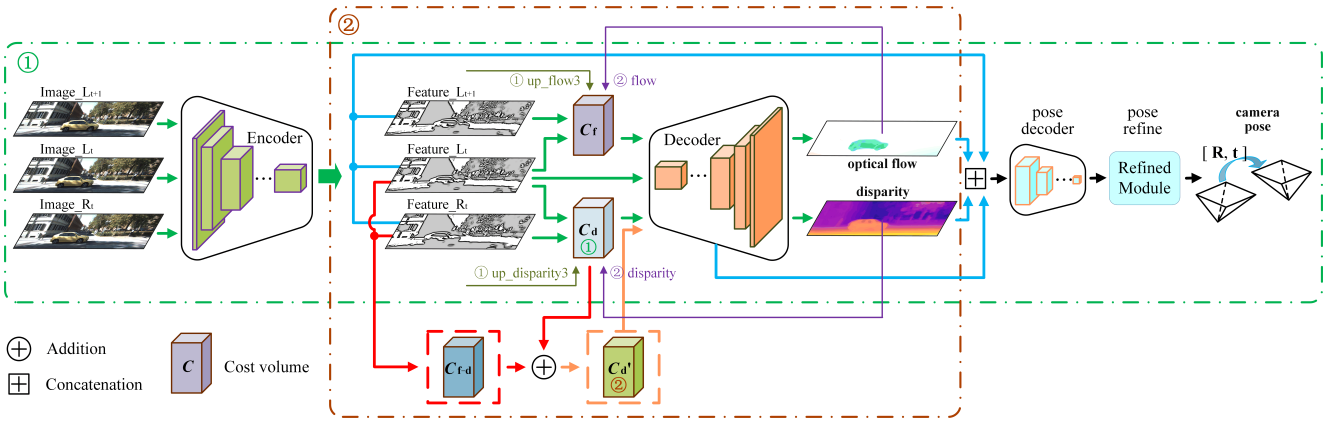

Figure 1. Overview of our united unsupervised learning framework which estimates optical flow, stereo depth and camera pose. We construct our network based on PWC-Net [43] which estimates five-scale flows. The green box ① presents our *Feature-sharing encoder* (Sec. 3.1), *Pooled decoder module*(Sec. 3.2) and *Camera pose estimation module*(Sec. 3.3). The brown box ② illustrates *Cost volume complement*(Sec. 3.4), which uses the $\boldsymbol{C}_d'$ instead of $\boldsymbol{C}_d$ as the input of the decoder, where $\boldsymbol{C}_d'$ is the combination of original $\boldsymbol{C}_d$ and $\boldsymbol{C}_{f-d}$ whose construction process is shown in Fig. 2 in detail.

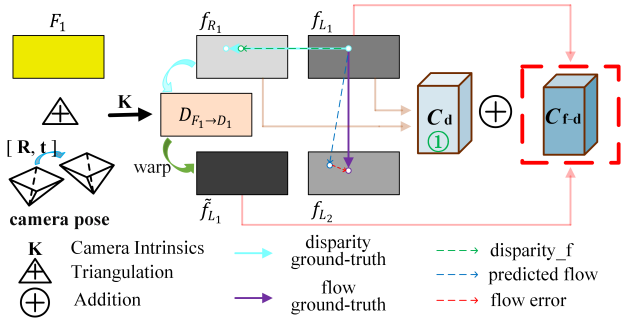

Figure 2. Overview of *Cost volume complement*. Given the refined camera pose and camera intrinsics, the estimated optical flow $\boldsymbol{F_1}$ can be converted into the disparity $\boldsymbol{D}_{\boldsymbol{F_1}\to\boldsymbol{D_1}}$ by Triangulation Algorithm [14]. Then the disparity $\boldsymbol{D}_{\boldsymbol{F_1}\to\boldsymbol{D_1}}$ is used to warp the feature $\boldsymbol{f}_{R_1}$ to the coordinate system of L_1, and then the warpped $\widetilde{\boldsymbol{f}}_{L_1}$ and the feature $\boldsymbol{f}_{L_1}$ are used to construct the cost volume $\boldsymbol{C}_{f-d}$. Finally, the $\boldsymbol{C}_{f-d}$ is added to the $\boldsymbol{C}_d$ which is constructed by the estimated disparity $\boldsymbol{D_1}$, feature $\boldsymbol{f}_{L_1}$ and $\boldsymbol{f}_{R_1}$.

compared with our method which directly estimates one-dimensional disparity. That is because that disparity can be modeled as a task to search pixel correspondence only in the horizontal lines.

Joint unsupervised optical flow, depth and camera pose estimation. GeoNet [48] is the first to jointly estimate depth, pose and optical flow. Following that, some methods [54, 4, 30, 45] use three individual networks to predict the three tasks respectively, and encourage geometric consistency between the rigid flow (computed from estimated disparity and camera motion) and the predicted optical flow in static regions. To better exploit geometric relationship in the real-world scenes with multiple moving objects, [44, 40, 23] use semantic information to segment the scene into static and moving regions. Then some works obtain moving objects motion by proposing a residual flow module [25] or estimate a 6-DOF rigid-body transformation for each dynamic object [6]. All these joint methods use several individual networks, which is computationally expensive and difficult to train simultaneously due to the inherent conflict. Though some of them exploit semantics to exclude or address dynamic objects, the predicted camera motion and dynamic objects motion are still far from ideal.

Different from these joint methods, in this paper, we propose a single feature-level collaboration network jointly estimate these three tasks which can greatly improve the performance of each task.

3. Method

The inputs of our network are two pairs of consecutive stereo images (L_1, L_2, R_1, R_2), where (L_1, R_1) and (L_2, R_2) indicate the left and right image at time t and $t+1$ respectively. Then the network can estimate the optical flow $\boldsymbol{F}_{t\to s}$ (from target to source image), the disparity $\boldsymbol{D}_{t\to s}$ and the camera motion $\boldsymbol{\xi}_{t\to s} \in \boldsymbol{se(3)}$. For concision, in this section, we denote $\boldsymbol{F_1}$ as the optical flow from L_1 to L_2, while $\boldsymbol{D_1}$ as the disparity from L_1 to R_1. We introduce our whole architecture in Sec. 3.1 and several losses for the unified learning in Sec. 3.2.

The whole network architecture shown in Fig. 1 includes four components: 1) feature-sharing encoder, 2) pooled optical flow and disparity decoder, 3) camera pose estimation, 4) cost volume complement.

3.1. Feature-sharing encoder

We adopt the feature extractor of PWC-Net [43] to extract the 5-scale feature pyramid of L_1, L_2, R_1 and R_2. And we denote $\boldsymbol{f}_t^l$ as the lth-level feature of image t. These extracted features are integrated together for the following optical flow and stereo depth estimation. Meanwhile, the shared features can be further used to predict camera motion. In addition, this weight-sharing feature extractor module reduce the number of network parameters.

3.2. Pooled optical flow and disparity decoder

In order to fully leverage the features of the four images and simultaneously decode both optical flow and disparity, we make two modifications to the original PWC-Net decoder: 1) inputting two cost volumes instead of original one cost volume. The whole estimation of optical flow and disparity is a coarse-to-fine process. Taking the lth-level decoder for example, one of the two cost volumes is constructed by the lth-level image features $\boldsymbol{f}_{L_1}^l$, $\boldsymbol{f}_{L_2}^l$ and the upsampled optical flow $\text{up}_2(\boldsymbol{F}_{\boldsymbol{1}}^{l+1})$, the other is constructed by the lth-level image features $\boldsymbol{f}_{L_1}^l$, $\boldsymbol{f}_{R_1}^l$ and upsampled disparity $\text{up}_2(\boldsymbol{D}_{\boldsymbol{1}}^{l+1})$. 2) increasing the number of output channels of the decoder from 2 to 3, to simultaneously decode both optical flow and disparity (i.e., the first 2 channels are for the optical flow and last 1 channel is for disparity). The whole process is shown in Fig. 1.

The pooled decoder module can integrate the features of optical flow and stereo depth, which can benefit both tasks. Compared with constructing only one cost volume as the input of the decoder in PWC-Net, the combination of the flow and disparity cost volumes can significantly enhance the features' representation ability of the feature-sharing encoder. In addition, this method also helps the network learn the inherent relationships between the two tasks, which can improve the performance of the decoder.

3.3. Camera pose estimation

This module includes two sub-modules: camera pose prediction module and refinement module.

Camera pose prediction module: This module estimates the relative camera pose $\boldsymbol{\xi}_{t\to s}$ between two frames. When predicting the relative camera motion $\boldsymbol{\xi}_{L_1\to L_2}$, the module inputs are the 2nd-level flow and disparity, the 2nd-level image features, and the fused features of the pooled decoder in 2nd-level, and the output is a 6-DOF camera pose. Compared with previous learning-based methods [54, 4, 30, 45] which directly regress the camera pose from original images, our method utilizes the feature-level information of optical flow and depth, which can achieve better performance on camera motion estimation.

Camera pose refinement module: To further improve the camera pose estimated by CNNs, we propose a pose refinement module. The initial estimated $\boldsymbol{\xi}_{t\to s}$ may have a slight deviation, which can be seen as a slight camera pose perturbation $\Delta\boldsymbol{\xi}_{t\to s}$. So like RDVO proposed in [45], our pose refinement module aims at finding the $\Delta\boldsymbol{\xi}_{t\to s}$ to obtain a more accurate camera pose. First, we show several conversion formula among flow, depth and disparity.

The disparity $\boldsymbol{D}$ and the depth $\boldsymbol{Z}$ can converted to each other by,

$$\boldsymbol{Z} = -f_x * b/\boldsymbol{D} \tag{1}$$

where f_x and b is the focal length and the baseline of stereo cameras, respectively.

Given the camera pose $\boldsymbol{\xi}_{t\to s}$ and the depth $\boldsymbol{Z}_t$, the rigid flow from t to s can be calculated as,

$$\boldsymbol{F}_{t\to s}^{rig} = \pi(\boldsymbol{K}\exp(\boldsymbol{\xi})\boldsymbol{Z}_t\boldsymbol{K}^{-1}p_t) - \pi(p_t) \tag{2}$$

where $\boldsymbol{K}$ is the given camera intrinsic and p_t is homogeneous coordinates in pixel plane. $\pi([x,y,z]) = [x/z, y/z]^T$ returns 2D non-homogeneous coordinates.

Like [45], we obtain a mask $\boldsymbol{M}_s$ of static regions by the flow consistency check, and the non-occluded region mask $\boldsymbol{M}_{noc}$ by the forward-backward consistency prior [33]. Then a static and non-occluded region mask $\boldsymbol{M}_{t\to s}^{s-noc}$ can be computed by

$$\boldsymbol{M}_{t\to s}^{s-noc} = \boldsymbol{M}_s \cdot \boldsymbol{M}_{noc} \tag{3}$$

Our basic assumption is that the estimated optical flow and stereo depth in non-occluded regions are accurate enough. So the $\Delta\boldsymbol{\xi}_{t\to s}$ can be computed by minimizing the reprojection error of N pixels which are randomly chosen in static and non-occluded regions,

$$\Delta\boldsymbol{\xi}^* = \arg\min_{\Delta\boldsymbol{\xi}^*} \frac{1}{\text{N}} \sum ||\pi(\boldsymbol{K}\boldsymbol{P}_s) - \pi(p_s)||^2 \tag{4}$$

$$\boldsymbol{P}_s = \exp(\Delta\boldsymbol{\xi}_{t\to s})\exp(\boldsymbol{\xi}_{t\to s})\boldsymbol{Z}_t\boldsymbol{K}^{-1}p_t \tag{5}$$

where $p_t \in \boldsymbol{M}_{t\to s}^{s-noc}$, and p_s is the corresponding pixel coordinate in the image s which is computed by p_t and the estimated optical flow. Now Eq(4) can be seen as a simple Bundle Adjustment(BA) problem which can be solved by Levenberg-Marquardt(LM) algorithm [35, 29] to get the optimal $\Delta\boldsymbol{\xi}_{t\to s}^*$. Then, finally, we can obtain a more accurate camera pose $\exp(\boldsymbol{\xi}_{t\to s}^{'}) = \exp(\Delta\boldsymbol{\xi}_{t\to s}^*)\exp(\boldsymbol{\xi}_{t\to s})$.

3.4. Cost volume complement

As analyzed in Sec. 1, existing joint methods usually focus on the loss-level jointly optimization, but not handle the degraded cost volume. In this section, based on the shared features, we propose a cost volume complement module to further utilize the image features to reinforce the flow and disparity cost volumes. This module mainly contains four parts: cost volume enhancement, cost volume interaction, iterative optimization and moving objects handling.

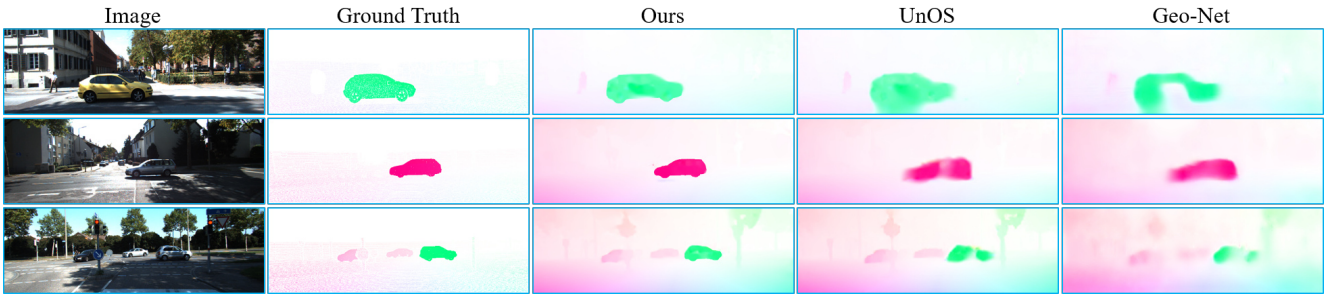

Figure 3. Qualitative results of our method. We compare each of our optical flow to previous methods, UnOS [45] and Geo-Net [48].

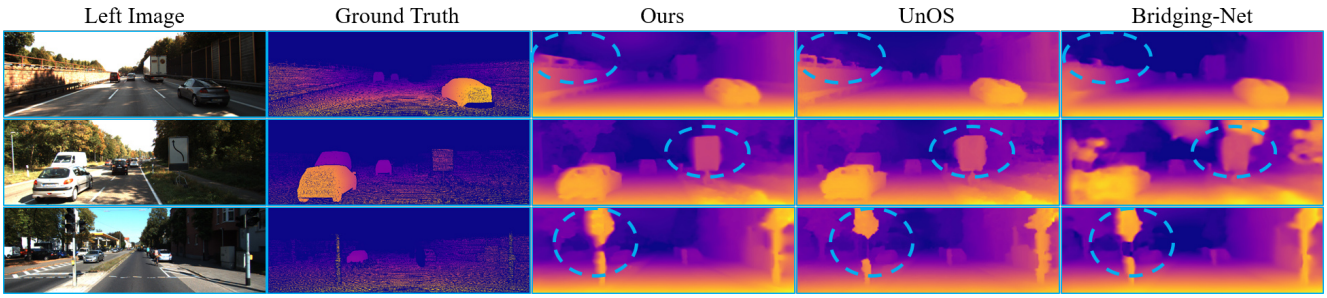

Figure 4. Qualitative results of our method. We compare each of our depth to previous methods, UnOS [45] and Bridging-Net [24]. Significantly improved regions are highlighted with dash circles. In the first and third rows, UnOS [45] estimate a disconnected handrail and telegraph pole, but our estimation is better.

Cost volume enhancement: As shown in Fig. 2, given the camera pose $\boldsymbol{\xi}_{L_1 \to L_2}$ and the optical flow $\boldsymbol{F_1}$, we build the cost volume $\boldsymbol{C}_{f-d}$. In most real-world datasets, most of the occluded pixels are invisible in the L_2, but visible in the R_1. So the cost volume $\boldsymbol{C}_{f-d}$ is a powerful enhancement for the cost volume $\boldsymbol{C}_f$.

Cost volume interaction: As shown in Fig. 1, we combine the $\boldsymbol{C}_d$ and $\boldsymbol{C}_{f-d}$ to get $\boldsymbol{C}_d^{'}$. Because the estimated disparity is more accurate than optical flow, especially in occluded regions, the combination of $\boldsymbol{C}_{f-d}$ and $\boldsymbol{C}_d$ would further improve the optical flow estimation performance.

Iterative optimization: For achieving the cost volume complement, The 2nd-level pooled decoder has been iterated three times with the same weights but different inputs and losses. Firstly, input original $\boldsymbol{C}_d$ and $\boldsymbol{C}_f$ to estimate optical flow and disparity. And the camera pose estimation model is only trained once at the first iteration. After the 1st iteration, we get the initial optical flow, depth and camera pose. Secondly, input $\boldsymbol{C}_d^{'}$ and $\boldsymbol{C}_f$, and only constrain the static regions with the static mask. After the 2nd iteration, we get more accurate optical flow in static regions. Thirdly, input $\boldsymbol{C}_d^{'}$ and $\boldsymbol{C}_f$, and constrain non-occluded regions by photometric loss and static occluded regions by the flow consistency loss. In the 3rd iteration, we further refine the optical flow and disparity. As shown in Fig. 1.

Moving objects handling: In theory, our cost volume complement module can also be applied in moving objects by using the different moving objects' poses instead of the camera pose. But due to that it is extremely difficult to estimate the accurate poses of moving objects, here, we introduce a novel method to handle moving objects. We re-extract the features from original input images, and copy the 2nd-level pooled decoder to predict the optical flow and disparity of dynamic objects. To this end, only the photometric constraints are provided for the training of this mini-module. The process is similar with the 1st iteration.

3.5. Training losses

In this section, we introduce several losses we designed for our joint unsupervised architecture.

Photometric losses. [30, 45, 48] use the linear combination of SSIM and L1 norm as their photometric loss. To better eliminate the negative influence of illumination variation, we add an extra Census loss [33]. Formally, our photometric loss can be written as,

$$\mathcal{L}_{*p}(\mathrm{O}) = \sum_{p_t} \mathrm{V}_*(p_t, \mathrm{O}) \cdot s(I_t(p_t), \widetilde{I}_t(p_t)), \tag{6}$$

$$\begin{aligned} \text{where, } s(I_t(p_t), \widetilde{I}_t(p_t)) = (1-\alpha)|I_t(p_t) - \widetilde{I}_t(p_t)| + \\ \alpha \cdot (1 - \frac{1}{2}\mathrm{SSIM}(I_t(p_t) - \widetilde{I}_t(p_t))) + \\ \beta \cdot \mathrm{Census}(I_t(p_t) - \widetilde{I}_t(p_t)) \end{aligned} \tag{7}$$

Method	United	Stereo	Supervised	KITTI 2012 train EPE-noc	KITTI 2012 train EPE-all	KITTI 2012 train EPE-occ	KITTI 2012 test EPE-all	KITTI 2015 train EPE-noc	KITTI 2015 train EPE-all	KITTI 2015 train EPE-occ	KITTI 2015 train Fl-all	KITTI 2015 test Fl-all
SpyNet-ft [39]			✓	–	(4.13)	–	–	–	–	–	–	–
FlowNet2-ft [18]			✓	–	(1.28)	–	1.8	–	(2.30)	–	–	11.48%
PWC-Net-ft [43]			✓	–	(1.47)	–	1.7	–	(2.16)	–	–	9.60%
UnFlow [33]				1.26	3.29	–	–	–	8.10	–	23.27%	–
DDFlow [26]				1.02	2.35	11.31	3.0	2.73	5.72	24.68	–	14.29%
SelFlow [28]				0.91	1.69	6.95	2.2	2.40	4.84	19.68	–	14.19%
UFlow [22]				–	1.68	–	1.9	[1.88]	[2.71]	–	–	11.13%
Geonet [48]	✓			–	–	–	–	–	10.81	–	–	–
Jiang *et al*. [21]	✓			0.94	1.56	–	1.9	–	–	–	–	–
DF-net-ft [54]	✓			–	3.54	–	4.4	–	[8.98]	–	[26.01%]	[22.82%]
Bridging-Net [24]	✓	✓		1.39	2.56	–	–	4.30	7.13	17.79	27.13%	–
EPC++ [30]	✓	✓		–	1.91	–	2.2	3.83	5.43	–	–	20.52%
UnOS [45]	✓	✓		1.04	1.64	5.30	1.8	3.79	5.58	22.01	–	18.00%
Flow2Stereo [27]	✓	✓		**0.82**	1.45	5.52	1.7	2.12	3.54	12.58	10.04%	11.10%
Matteo *et al*. [44]	✓			–	–	–	–	3.29	5.39	–	20.00%	19.47%
Junhwa *et al*. [17]	✓			–	–	–	–	–	7.51	–	23.49%	23.54%
Our(full)	✓	✓		**0.82**	**1.25**	**3.88**	**1.5**	**1.57**	**2.35**	**6.68**	**9.09%**	**9.70%**

Table 1. Quantitative evaluation on the optical flow task. EPE means average end-point-error where the postfix '-noc' and '-all' only accounts for non-occlusion regions and all regions, respectively. Fl is the percentage of erroneous pixels. A pixel is considered to be correctly estimated if the EPE is <3px or <5%. '()': trained on the labeled evaluation set, '[]' trained on the unlabeled evaluation set.

Here, α and β are the balancing hyper-parameter which are set to be 0.85 and 0.5 repectively. O represents the type of the inputs for obtaining the matching pixel, which can be optical flow, rigid flow or disparity. $\widetilde{I}_t$ is the warped image from I_s to target image plane by O. V_* indicates the visible regions mask relying on the matching images. For the optical flow, V_{OF} is computed by forward-backward consistency check [16].

Edge-aware smoothness. We use similar image gradient based edge-aware smooth loss $\mathcal{L}_{*s}(\mathrm{O})$ like [30].

Camera pose loss. Based on [45], we add the extra pixel loss between L_1 and R_2 by leveraging the fixed camera pose between the left and right camera $\boldsymbol{\xi}_{L\to R}$.

$$\exp(\boldsymbol{\xi}^{'}_{L_1\to R_2}) = \exp(\boldsymbol{\xi}_{L\to R}) \cdot \exp(\boldsymbol{\xi}^{'}_{L_1\to L_2}) \tag{8}$$

Given the refined camera pose $\boldsymbol{\xi}^{'}_{L_1\to L_2}$ and the estimated disparity, we can obtain the $\boldsymbol{\xi}^{'}_{L_1\to R_2}$ by Eq(8), and they can be used to compute $\boldsymbol{F}^{rig}_{L_1\to L_2}$ and $\boldsymbol{F}^{rig}_{L_1\to R_2}$ by Eq(2). And we can obtain the static non-occluded masks $\boldsymbol{M}^{s-noc}_{L_1\to L_2}$ and $\boldsymbol{M}^{s-noc}_{L_1\to R_2}$ using Eq(3). Then we can use the two rigid flows and the two masks to compute two photometric losses, which can be added together to get the final camera pose loss $\mathcal{L}_{pose}$.

Flow consistency loss. In static occluded regions, the rigid flow is more accurate than predicted optical flow. So we construct a flow consistency loss between the optical flow and rigid flow in the regions, which can be written as,

$$\mathcal{L}_{ro} = \sum_{p_t} \boldsymbol{M}^{s-occ}_{t\to s}\, \rho(\boldsymbol{F}^{rig^{'}}_{t\to s}(p_t) - \boldsymbol{F}_{t\to s}(p_t)) \tag{9}$$

where $\boldsymbol{F}^{rig^{'}}_{t\to s}(p_t)$ is computed by refined camera pose $\boldsymbol{\xi}^{'}_{L_1\to L_2}$ and depth $\boldsymbol{Z}_t$ according Eq(2), the $\boldsymbol{M}^{s-occ}_{t\to s}$ is the static and occluded mask, which can be computed as,

$$\boldsymbol{M}^{s-occ}_{t\to s} = \boldsymbol{M}_s \dot{(}1 - \boldsymbol{M}_{noc}) \tag{10}$$

and ρ is the generalized Charbonnier loss [49], $\rho(\boldsymbol{x}) = (\boldsymbol{x}+\epsilon)^{\alpha_1}$. In our experiment, the ϵ and α_1 are set to be 0.001 and 0.5 respectively.

Total loss. In summary, our total loss function is,

$$\mathcal{L}_{total} = (\mathcal{L}_{fp}+\mathcal{L}_{dp})+\lambda_s(\mathcal{L}_{fs}+\mathcal{L}_{ds})+\lambda_{pose}\mathcal{L}_{pose}+\lambda_{ro}\mathcal{L}_{ro} \tag{11}$$

where λ_s, λ_{pose} and λ_{ro} are the balanced weights.

4. Experiment

In this section, we first introduce the datasets, and then describe our training details, and compare our results with other SOTA methods on the tasks of optical flow, stereo depth and visual odometry.

4.1. Datasets

For the optical flow and stereo depth tasks, we use the first 11 sequences of KITTI Visual Odometry (VO) dataset and raw multi-view extension of KITTI 2012[10] and KITTI 2015 [34] excluding neighboring frames(frames 9-12) as [41, 46, 27, 26, 28] as the training set. And as most previous methods, we evaluate the optical flow and stereo depth on KITTI 2012 benchmark(194 training image pairs and 195 test image pairs) and KITTI 2015 benchmark(200 training image pairs and 200 test image pairs). For the odometry task, we use the official odometry data

Method	Train Stereo	Test Stereo	Lower the better						
			Abs Rel	Sq Rel	RMSE	RMSE log	EPE-noc	EPE-all	EPE-occ
Geonet [48]			0.153	1.328	5.737	0.232	–	–	–
Ranjan *et al.* [40]			0.140	1.070	5.326	0.217	–	–	–
Matteo *et al.* [44]			0.118	0.748	4.608	0.186	–	–	–
EPC++ [30]	✓		0.109	1.004	6.232	0.203	–	–	–
Godard *et al.* [11]	✓		0.097	0.896	5.093	0.176	–	–	–
Zhou *et al.* [51]	✓	✓	–	-	-	-	8.35	9.41	–
Godard *et al.* [11]	✓	✓	0.068	0.835	4.392	0.146	–	–	–
Bridging-Net [24]	✓	✓	0.087	0.765	4.380	0.184	1.47	1.55	1.98
UnOS [45]	✓	✓	0.049	0.515	3.404	0.121	1.22	1.28	1.62
Flow2Stereo [27]	✓	✓	–	-	-	-	1.31	1.34	2.56
Our(Depth-only)	✓	✓	0.063	0.662	4.312	0.140	1.30	1.42	1.85
Our(full)	✓	✓	**0.047**	**0.394**	**3.358**	**0.119**	**1.16**	**1.22**	**1.50**

Table 2. Quantitative evaluation of the stereo depth task on the KITTI 2015 training set. Abs Rel, Sq Rel, RMSE, RMSE log, EPE are standard metrics for depth evaluation [24, 45]. We capped the depth to be between 0-80 meters to compare with existing methods. Using stereo pairs during training/testing is also indicated in the table.

Method	KITTI 2012			KITTI 2015		
	EPE noc	EPE all	EPE occ	EPE noc	EPE all	EPE occ
Flow Only	1.04	2.40	9.99	3.38	5.96	16.53
Feature Fusion	0.94	1.93	7.62	2.95	5.51	14.98
Feature Fusion + Pose	0.88	1.35	4.42	2.02	3.08	8.76
Cost Volume Fusion	0.90	1.87	7.40	2.64	5.06	12.69
Cost Volume Fusion + Pose	**0.82**	**1.25**	**3.88**	**1.57**	**2.35**	**6.68**

Table 3. Ablation study of optical flow estimation on KITTI 2012 and KITTI 2015 training split.

Method	KITTI 2015	
	EPE static	EPE moving
Feature Fusion + Pose	2.56	4.55
Cost Volume Fusion + Pose(static)	**1.84**	10.05
Cost Volume Fusion + Pose(moving)	3.10	**3.30**
Cost Volume Fusion + Pose(full)	**1.84**	**3.30**

Table 4. Ablation study of optical flow estimation on KITTI 2015 training split.

split, i.e. using 9 sequences(Seq. 00-08) for training and 2 sequences(Seq. 09 and Seq. 10) for testing.

4.2. Implementation details

The whole training process consists of four stages. In the first stage, we train the feature-sharing encoder and pooled decoder on KITTI VO dataset, using our photometric losses and edge-aware smooth losses, which λ_s is set to 10. During the training, the batch size is 2, and the initial learning rate is 0.0001 and decreases to 0.00001 after 10 epochs. The original images with simple color agumentation are randomly cropped or resized to 320 $\times$ 896 as the inputs. In the second stage, we use our camera pose loss $\mathcal{L}_{pose}$ to train the camera pose prediction module on Seq. 00-08 of KITTI VO dataset. Then in the third stage, according to the procedure metioned in Sec. 3.4, we train the the cost volume complement module on the multi-view extension of KITTI 2012 and KITTI 2015. After the 3rd stage, we can obtain the best static flow. In the 4th stage, we train the moving objects handling module to get the dynamic objects' flow. At last, we obtain the final flow by compositing the static flow and moving objects' flow.

4.3. Main results

Optical flow. We evaluate our model on both KITTI 2015, KITTI 2012 training and test splits. The quantitative results are shown in Tab. 1 and the visualization results are shown in Fig. 3. The results show that our method outperforms all current joint unsupervised estimation methods on all evaluation metrics(EPE and Fl). Specially, on KITTI 2012, we achieve EPE-all = 1.25, which even outperforms some SOTA supervised methods, like PWC-Net [43] (1.47).

In addition, there is also much improvement on KITTI 2015. We achieve EPE-all = 2.35, resulting in 33.6% relative improvement than previous best joint unsupervised method Flow2Stereo [27]. And our method also performs better than the Uflow [22], which is the current best unsupervised optical flow estimation method.

• **Ablation Study.** We choose the training split for our ablation study because the ground truth of test split is withheld. We present 5 different variants in Tab. 3, including:

- Flow Only. We train original PWC-Net to estimate optical flow, using our photometric and smoothness losses.

- Feature Fusion. Only the first two modules(feature-sharing encoder and pooled decoder) are trained with the same losses with Flow Only.

- Cost Volume Fusion. We use the cost volume $C_d^{'}$ instead of original C_d to train the network by the same losses with Feature Fusion.

- Feature Fusion + Pose. The model is trained basing on Feature Fusion model by adding the flow consistency loss.

- Cost Volume Fusion + Pose. The Cost Volume Fu-

Method	Stereo	Seq. 09	Seq. 10
ORB-SLAM(full) [36]		(0.014 ± 0.008)	(0.012 ± 0.011)
Zhou *et al.* [53]		(0.021 ± 0.017)	(0.020 ± 0.015)
DF-Net [54]		(0.017 ± 0.007)	(0.015 ± 0.009)
Mahjourian *et al.* [31]		(0.013 ± 0.010)	(0.012 ± 0.011)
EPC++(mono) [30]		(0.013 ± 0.007)	(0.012 ± 0.008)
GeoNet *et al.* [48]		(0.012 ± 0.007)	(0.012 ± 0.009)
Ranjan *et al.* [40]		(0.012 ± 0.007)	(0.012 ± 0.008)
EPC++(stereo) [30]	✓	(0.012 ± 0.006)	(0.012 ± 0.008)
UnOS(MotionNet) [45]	✓	(0.023 ± 0.010)	(0.022 ± 0.016)
UnOS(Full) [45]	✓	(0.012 ± 0.006)	(0.013 ± 0.008)
Our(MotionModule)	✓	(0.011 ± 0.005)	$(\mathbf{0.006 \pm 0.003})$
Our(Full)	✓	$(\mathbf{0.009 \pm 0.005})$	$(\mathbf{0.006 \pm 0.003})$

Table 5. Odometry evaluation on two testing sequences of KITTI dataset using the metric of the absolute trajectory error.

sion continues to be trained basing on Cost Volume Fusion model by adding the flow consistency loss.

The results of Flow Only and Feature Fusion validate our insight that feature-sharing encoder and pooled decoder can improve the optical flow estimation in non-occluded region (from 1.04 to 0.94 on KITTI 2012, from 3.38 to 2.95 on KITTI 2015), while the results of Flow Only and Cost Volume Fusion verify that the interactive cost volume for the decoder can significantly improve the optical flow estimation in occluded region (from 9.99 to 7.40 on KITTI 2012, from 16.53 to 12.69 on KITTI 2015). The results of last two rows in Tab. 3 show that the rigid flow can provide a more accurate constraint for optical flow of occluded pixels.

We also present 4 different variants in Tab. 4 with the metrics of the flow estimation in static and moving regions. The results of Feature Fusion + pose and Cost Volume Fusion + Pose(static) validate our assumption that the interactive cost volume can significantly improve the optical flow estimation in static regions (from 2.56 to 1.84). In addition, the results of Feature Fusion + pose and Cost Volume Fusion + Pose(moving) validate our moving objects handling can improve the performance in moving regions.

Stereo Depth. We evaluate the depth task on the KITTI 2015 dataset, and use Abs Rel, Sq Rel, RMSE, RMSE log and EPE as our evaluation metrics as [24, 45]. The results are shown in Tab. 2 and Fig. 4. The Depth-only is obtained by modifying the PWC-Net to estimate only one dimension as the disparity. Our(full) shows that our method performs better than existing unsupervised depth estimation methods, especially in SqRel(0.394).

Visual Odometry. We compare our pose estimation method with SOTA methods. ORB-SLAM [36], ORB-SLAM2 [37] and LSD-SLAM [8] are traditional visual SLAM system and others are unsupervised deep-learning methods. We use two commonly adopted metrics proposed in [53, 50], and use the same evaluation method as UnOS [45]. The quantitative results are shown in Tab. 5 and Tab. 6, and the trajectory are shown in Fig. 5.

Method	Seq. 09		Seq. 10	
	$t_{err}\%$	$r_{err}(^o/100)$	$t_{err}\%$	$r_{err}(^o/100)$
ORB-SLAM(full) [36]	2.51	0.26	2.10	0.48
ORB-SLAM2(stereo) [37]	0.82	–	0.58	–
LSD-SLAM(stereo) [8]	1.22	–	0.75	–
Zhou *et al.* [53]	30.75	11.41	44.22	12.42
GeoNet *et al.* [48]	39.43	14.30	28.99	8.85
Zhan *et al.* [50]	11.92	3.60	12.62	3.43
EPC++(mono) [30]	8.84	3.34	8.86	3.18
Jiang *et al.* [21]	4.36	0.69	4.04	1.37
UnOS(MotionNet) [45]	13.98	5.36	19.67	9.13
UnOS(Full) [45]	5.21	1.80	5.20	2.18
Our(MotionModule)	5.77	1.92	5.34	2.86
Our(Full)	**2.02**	**0.54**	**1.81**	**1.03**

Table 6. Odometry evaluation on two testing sequences of KITTI dataset using the metric of average translation and rotational errors.

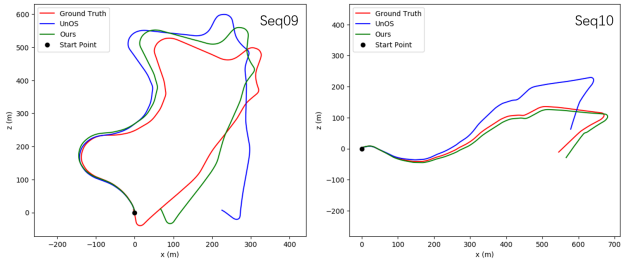

Figure 5. Trajectories of our method, UnOS [45] and Ground Truth in KITTI sequnences 09 and 10.

Our(MotionModule) is the results directly predicted by the camera pose prediction module. It is even better than current unsupervised learning-based camera motion estimation methods. Notably, the performance on the metric of Average Translation and Rotation Errors is much more impressive, which shows that our methods can obtain better odometry performance in the long test sequences. In addition, compared with the results of UnOS(MotionNet) which uses original images as the input of camera motion module, Our(MotionModule) performs much better, i.e. (5.77 and 1.92) vs (13.98 and 5.36) in Seq. 09, (5.34 and 2.86) vs (19.67 and 9.13) in Seq. 10. This validates that our idea that the feature-level information of optical flow and depth can help further improve the camera pose estimation.

5. Conclusion

We present a single network to jointly estimate optical flow, stereo depth and camera pose in unsupervised manner. We explore the feature-level collaboration of the three tasks. And our method achieves superior performance among all the three sub-tasks joint unsupervised methods. In the future, we will explore how to further improve the camera motion estimation by using multiple frames.

References

[1] Arunkumar Byravan and Dieter Fox. Se3-nets: Learning rigid body motion using deep neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 173–180. IEEE, 2017. 1

[2] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Unsupervised monocular depth and ego-motion learning with structure and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1

[3] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. *CoRR*, abs/1803.08669, 2018. 2

[4] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *Proceedings of the IEEE international conference on computer vision*, pages 7063–7072, 2019. 1, 3, 4

[5] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *arXiv preprint arXiv:1810.02695*, 2018. 2

[6] Qi Dai, Vaishakh Patil, Simon Hecker, Dengxin Dai, Luc Van Gool, and Konrad Schindler. Self-supervised object motion and depth estimation from video, 2019. 1, 2, 3

[7] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 1, 2

[8] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014. 8

[9] Andreas Geiger, Martin Lauer, Christian Wojek, Christoph Stiller, and Raquel Urtasun. 3d traffic scene understanding from movable platforms. *IEEE transactions on pattern analysis and machine intelligence*, 36(5):1012–1025, 2013. 1

[10] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 6

[11] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 270–279, 2017. 2, 7

[12] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 3828–3838, 2019. 2

[13] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. Group-wise correlation stereo network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3273–3282, 2019. 1, 2

[14] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2 edition, 2004. 3

[15] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8981–8989, 2018. 2

[16] Junhwa Hur and Stefan Roth. Mirrorflow: Exploiting symmetries in joint optical flow and occlusion estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 312–321, 2017. 6

[17] Junhwa Hur and Stefan Roth. Self-supervised monocular scene flow estimation, 2020. 6

[18] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks, 2016. 2, 6

[19] Joel Janai, Fatma Guney, Anurag Ranjan, Michael Black, and Andreas Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 690–706, 2018. 2

[20] J Yu Jason, Adam W Harley, and Konstantinos G Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *European Conference on Computer Vision*, pages 3–10. Springer, 2016. 2

[21] Shihao Jiang, Dylan Campbell, Miaomiao Liu, Stephen Gould, and Richard Hartley. Joint unsupervised learning of optical flow and egomotion with bi-level optimization. *arXiv preprint arXiv:2002.11826*, 2020. 1, 6, 8

[22] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. *arXiv preprint arXiv:2006.04902*, 2020. 6, 7

[23] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem by semantic guidance. *arXiv preprint arXiv:2007.06936*, 2020. 2, 3

[24] Hsueh-Ying Lai, Yi-Hsuan Tsai, and Wei-Chen Chiu. Bridging stereo matching and optical flow via spatiotemporal correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1890–1899, 2019. 1, 2, 5, 6, 7, 8

[25] Seokju Lee, Sunghoon Im, Stephen Lin, and In So Kweon. Learning residual flow as dynamic motion from stereo videos. *arXiv preprint arXiv:1909.06999*, 2019. 3

[26] Pengpeng Liu, Irwin King, Michael R Lyu, and Jia Xu. Ddflow: Learning optical flow with unlabeled data distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8770–8777, 2019. 2, 6

[27] Pengpeng Liu, Irwin King, Michael R Lyu, and Jia Xu. Flow2stereo: Effective self-supervised learning of optical flow and stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6648–6657, 2020. 1, 2, 6, 7

[28] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4571–4580, 2019. 2, 6

[29] M. L. A. Lourakis and A. A. Argyros. Is levenberg-marquardt the most efficient optimization algorithm for implementing bundle adjustment? In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1526–1531 Vol. 2, 2005. 4

[30] Chenxu Luo, Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts++: Joint learning of geometry and motion with 3d holistic understanding. *arXiv preprint arXiv:1810.06125*, 2018. 1, 3, 4, 5, 6, 7, 8

[31] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5667–5675, 2018. 2, 8

[32] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016. 1, 2

[33] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. *arXiv preprint arXiv:1711.07837*, 2017. 2, 4, 5, 6

[34] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3061–3070, 2015. 6

[35] Jorge J. Moré. The levenberg-marquardt algorithm: Implementation and theory. In G. A. Watson, editor, *Numerical Analysis*, pages 105–116, Berlin, Heidelberg, 1978. Springer Berlin Heidelberg. 4

[36] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 8

[37] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017. 8

[38] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 343–352, 2015. 1

[39] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4161–4170, 2017. 2, 6

[40] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 12240–12249, 2019. 2, 3, 7, 8

[41] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 6

[42] Amnon Shashua, Yoram Gdalyahu, and Gaby Hayun. Pedestrian detection for driving assistance systems: Single-frame classification and system level performance. In *IEEE Intelligent Vehicles Symposium, 2004*, pages 1–6. IEEE, 2004. 1

[43] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8934–8943, 2018. 2, 3, 4, 6, 7

[44] Fabio Tosi, Filippo Aleotti, Pierluigi Zama Ramirez, Matteo Poggi, Samuele Salti, Luigi Di Stefano, and Stefano Mattoccia. Distilled semantics for comprehensive scene understanding from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4654–4665, 2020. 1, 2, 3, 6, 7

[45] Yang Wang, Peng Wang, Zhenheng Yang, Chenxu Luo, Yi Yang, and Wei Xu. Unos: Unified unsupervised optical-flow and stereo-depth estimation by watching videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8071–8081, 2019. 1, 3, 4, 5, 6, 7, 8

[46] Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, Peng Wang, and Wei Xu. Occlusion aware unsupervised learning of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4884–4893, 2018. 2, 6

[47] Jonas Wulff and Michael J Black. Temporal interpolation as an unsupervised pretraining task for optical flow estimation. In *German Conference on Pattern Recognition*, pages 567–582. Springer, 2018. 2

[48] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1983–1992, 2018. 3, 5, 6, 7, 8

[49] Jason J. Yu, Adam W. Harley, and Konstantinos G. Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness, 2016. 6

[50] Huangying Zhan, Ravi Garg, Chamara Saroj Weerasekera, Kejie Li, Harsh Agarwal, and Ian Reid. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 340–349, 2018. 8

[51] Chao Zhou, Hong Zhang, Xiaoyong Shen, and Jiaya Jia. Unsupervised learning of stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 2, 7

[52] Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In *Proceedings of the European conference on computer vision (ECCV)*, pages 822–838, 2018. 1

[53] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017. 1, 2, 8

[54] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 36–53, 2018. 3, 4, 6, 8