GyroFlow: Gyroscope-Guided Unsupervised Optical Flow Learning

Haipeng Li[1] Kunming Luo[1] Shuaicheng Liu[2,1*]

[1]Megvii Technology
[2]University of Electronic Science and Technology of China
{lihaipeng, luokunming}@megvii.com, liushuaicheng@uestc.edu.com

Abstract

Existing optical flow methods are erroneous in challenging scenes, such as fog, rain, and night because the basic optical flow assumptions such as brightness and gradient constancy are broken. To address this problem, we present an unsupervised learning approach that fuses gyroscope into optical flow learning. Specifically, we first convert gyroscope readings into motion fields named gyro field. Second, we design a self-guided fusion module to fuse the background motion extracted from the gyro field with the optical flow and guide the network to focus on motion details. To the best of our knowledge, this is the first deep learning-based framework that fuses gyroscope data and image content for optical flow learning. To validate our method, we propose a new dataset that covers regular and challenging scenes. Experiments show that our method outperforms the state-of-art methods in both regular and challenging scenes. Code and dataset are available at https://github.com/megvii-research/GyroFlow.

1. Introduction

Optical flow estimation is a fundamental yet essential computer vision task that has been widely applied in various applications such as object tracking [1], visual odometry [4], and image alignments [23]. The original formulation of the optical flow was proposed by Horn and Schunck [10], after which the accuracy of optical flow estimation algorithms has been improved steadily. Early traditional methods minimize pre-defined energy functions with various assumptions and constraints [35]. Deep learning-based methods directly learn the per-pixel regression through convolutional neural networks, which can be divided into supervised [6, 40, 43] and unsupervised methods [41, 36]. The former methods are primarily trained on synthetic data [6, 3] due to the lack of ground-truth labels.

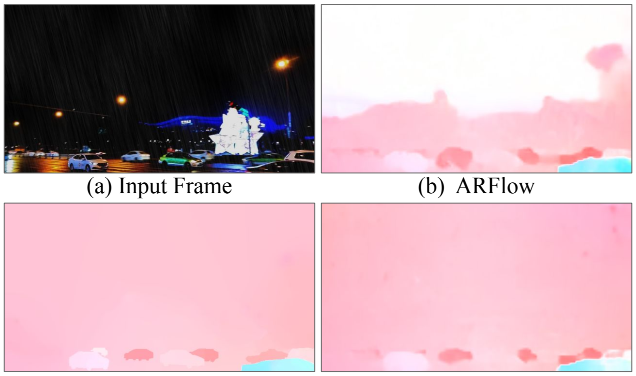

Figure 1. (a) Input low-light frame. (b) Optical flow result from existing baseline method ARFlow [30]. (c) Ground-Truth. (d) Result from our GyroFlow.

In contrast, the later ones can be trained on abundant and diverse unlabeled data by minimizing the photometric loss between two images. Although existing methods achieve good results, they rely on image contents, requiring images to contain rich texture and similar illumination conditions.

On the other hand, gyroscopes do not rely on image contents, which provide angular velocities in terms of roll, pitch, and yaw that can be converted into 3D motion, widely used for system control [26] and the HCI of mobiles [8]. Among all potential possibilities [2, 27, 14], one is to fuse the gyro data for the motion estimation. Hwangbo *et al.* proposed to fuse gyroscope to improve the robustness of KLT feature tracking [14]. Bloesch *et al.* fused gyroscope for the ego-motion estimation [2]. These attempts demonstrate that if the gyroscope is integrated correctly, the performance and the robustness of the method can be largely improved.

Given camera intrinsic parameters, gyro readings can be converted into motion fields to describe background motion instead of dynamic object motion because it is confined to camera motion. It is engaging that gyroscopes do not require the image contents but still produce reliable background camera motion under conditions of poor texture or

[*]Corresponding author

dynamic scenes. Therefore, gyroscopes can be used to improve the performance of optical flow estimation in challenging scenes, such as poor texture or inconsistent illumination conditions.

In this paper, we propose GyroFlow, a gyroscope-guided unsupervised optical flow estimation method. We combine the advantages of image-based optical flow that recovers motion details based on the image content with those of a gyroscope that provides reliable background camera motion independent of image contents. Specifically, we first convert gyroscope readings into gyro fields that describe background motion given the image coordinates and the camera intrinsic. Second, we estimate optical flow with an unsupervised learning framework and insert a proposed **S**elf-**G**uided **F**usion (SGF) module that supports the fusion of the gyro field during the image-based flow calculation. Fig. 1 shows an example, where Fig. 1 (a) represents the input of a night scene with poor image texture, and Fig. 1 (c) is the ground-truth optical flow between two frames. Image-based methods such as ARFlow [30] (Fig. 1 (b)) can produce the dynamic object motion but fail to compute the background motion in the sky, where no texture is available. Fig. 1 (d) shows our GyroFlow fusion result. As seen, both global motion and motion details can be retained. From experiments, we notice that motion details can be better recovered if global motion is provided.

To validate our method, we propose a dataset GOF (**G**yroscope **O**ptical **F**low) containing scenes under 4 different categories with synchronized gyro readings, including one regular scene (RE) and three challenging cases as low light scenes (Dark), foggy scenes (Fog), and rainy scenes (Rain). For quantitative evaluations, we further propose a test set, which includes accurate optical flow labels by the method [29], through extensive efforts. Note that existing flow datasets, such as Sintel [3], KITTI [7, 38] cannot be used for the evaluation due to the absence of the gyroscope readings. To sum up, our main contributions are:

- We propose the first DNN-based framework that fuses gyroscope data into optical flow learning.
- We propose a self-guided fusion module to effectively realize the fusion of gyroscope and optical flow.
- We propose a dataset for the evaluation. Experiments show that our method outperforms existing methods.

2. Related Work

2.1. Gyro-based Vision Applications

Gyroscopes reflect the camera rotation. Many applications equipped with the gyroscope have been widely applied, including but not limited to video stabilization [21], image deblur [39], optical image stabilizer (OIS) [25], simultaneous localization and mapping (SLAM) [11], ego-motion estimation [2], gesture-based user authentication on mobile devices [9], image alignment with OIS calibration [33] and human gait recognition [49]. The gyroscopes are important in mobile phones. The synchronization between the gyro readings and the video frames is important. Jia *et al.* [18] proposed gyroscope calibration to improve the synchronization. Bloesch *et al.* [2] fused optical flow and inertial measurements to deal with the drifting issue. In this work, we acquire gyroscope data from the bottom layer of the Android layout, i.e., Hardware Abstraction Layer (HAL), to achieve accurate synchronizations.

2.2. Optical Flow

Our method is related to optical flow estimation. Traditional methods minimize the energy function between image pairs to compute an optical flow [35]. Recent deep approaches can be divided into supervised [6, 40, 43] and unsupervised methods [41, 36].

Supervised methods require labeled ground-truth to train the network. FlowNet [6] first proposed to train a fully convolutional network on synthetic dataset FlyingChairs. To deal with the large displacement scenes, SpyNet [40] introduced a coarse-to-fine pyramid network. PWC-Net [42], LiteFlowNet [12], IRR-PWC [13] designed lightweight and efficient networks by warping features, computing cost volumes, and introducing residual learning for iterative refinement with shared weights. Recently, RAFT [43] achieved state-of-the-art performance by constructing a pixel-level correlation volume and using a recurrent network to estimate optical flow.

Unsupervised methods do not require ground-truth annotations. DSTFlow [41] and Back2Basic [17] are pioneers for unsupervised optical flow estimation. Several works [37, 31, 44, 30] focus on dealing with the occlusion problem by forward-backward occlusion checking, range-map occlusion checking, data distillation, and augmentation regularization loss. Other methods concentrate on optical flow learning by improving image alignment, including the census loss [37], formulation of multi-frames [16], epipolar constraints [51], depth constraints [47], feature similarity constraints [15], and occlusion inpainting [34]. UFlow [19] proposed a unified framework to systematically analyze and integrate different unsupervised components. Recently, UPFlow [36] proposed a neural upsampling module and pyramid distillation loss to improve the upsampling and learning of pyramid network, achieving state-of-art performance.

However above methods may not work well under challenging scenes, such as dark, rain, and fog environments. Zheng *et al.* proposed a data-driven method that establishes a noise model to learn optical flow from low-light images [50]. Li *et al.* proposed a RainFlow, which includes 2 modules to handle the rain veiling effect and rain streak effect respectively, to produce optical flow in the heavy rain [28]. Yan *et al.* proposed a semi-supervised network that converts foggy images into clean images to deal with dense foggy scenes [46]. In this paper, we build our GyroFlow upon unsupervised components with the fusion of gyroscope to cover both regular and the challenging scenes.

2.3. Gyro-based Motion Estimation

Hwangbo *et al.* proposed an inertial-aided KLT feature tracking method to handle the camera rolling and illumination change [14]. Bloesch *et al.* presented a method for fusing optical flow and inertial measurements for robust ego-motion estimation [2]. Li *et al.* proposed a gyro-aided optical flow estimation method to improve the performance under fast rotations [27]. Specifically, they produce a sparse optical flow that ignores foreground motion. However, none of them took challenging scenes into account nor used neural networks to fuse gyroscope data for optical flow improvement. In this work, including producing dense optical flow and taking rolling-shutter effects into account, we propose a DNN-based solution that fuses gyroscope data to image-based flow to improve optical flow estimations.

3. Algorithm

Our method is built upon convolutional neural networks that inputs a gyro field G_{ab} and two frames I_a, I_b to estimate a forward optical flow V_{ab} that describes the motion for every pixel from I_a towards I_b as:

$$V_{ab} = \mathcal{F}_\theta \left(G_{ab}, I_a, I_b\right), \tag{1}$$

where $\mathcal{F}$ is our network with parameter θ.

Fig. 2 illustrates our pipeline. Firstly, the gyro field G_{ab} is produced by the gyroscope readings between the relative frames I_a and I_b (Sec. 3.1), then it is concatenated with the two frames to be fed into the network to produce an optical flow V_{ab} between I_a and I_b. Our network consists of two stages. For the first stage, we extract feature pairs at different scales. For the second stage, we use the decoder $\mathbf{D}$ and the self-guided fusion module $\mathbf{SGF}$ (Sec. 3.2) to produce optical flow in a coarse-to-fine manner.

Our decoder $\mathbf{D}$ is same as UPFlow [36] which consists of the feature warping [42], the cost volume construction [42], the cost volume normalization [19], the self-guided upsampling [36], and the parameter sharing [13]. In summary, the second pyramid decoding stage can be formulated as:

$$\begin{aligned} \hat{V}_{ab}^{i-1} &= \mathbf{SGF}\left(F_a^i, F_b^i, V_{ab}^{i-1}, G_{ab}^{i-1}\right), \\ V_{ab}^i &= \mathbf{D}\left(F_a^i, F_b^i, \hat{V}_{ab}^{i-1}\right), \end{aligned} \tag{2}$$

where i represents the number of pyramid levels, F_a^i, F_b^i are extracted features from I_a and I_b at the i-th pyramid level. In the i-th layer, $\mathbf{SGF}$ takes image features F_a^i, F_b^i from the feature pyramid, the output V_{ab}^{i-1} of decoder $\mathbf{D}$ from the last layer and the downscale gyro field G_{ab}^{i-1} as inputs, then it produces a fusion result $\hat{V}_{ab}^{i-1}$ which is fed into $\mathbf{D}$. The D takes image features F_a^i, F_b^i and the fusion result $\hat{V}_{ab}^{i-1}$ as inputs and outputs a flow V_{ab}^i. Specifically, the output flow is directly upsampled at the last layer. Next, we first describe how to convert the gyro readings into a gyro field in Sec. 3.1 and then introduce our $\mathbf{SGF}$ module in Sec. 3.2.

3.1. Gyro Field

We obtain gyroscope readings from mobile phones that are widely available and easy to access. For mobile phones, gyroscopes reflect camera rotations. We compute rotations by compounding gyroscope readings that include 3-axis angular velocities and timestamps. In particular, compared to previous work [21, 24, 39] that read gyro readings from the API, we directly read them from HAL of Android architecture to avoid the non-trivial synchronization problem that is critical for the gyro accuracy. Between frames I_a and I_b, the rotation vector $n = (\omega_x, \omega_y, \omega_z) \in \mathbb{R}^3$ is computed according to method [21], then the rotation matrix $R(t) \in SO(3)$ can be produced by Rodrigues Formula [5].

In the case of a global shutter camera, e.g., the pinhole camera, a rotation-only homography can be computed as:

$$\mathbf{H(t)} = \mathbf{KR}(t)\mathbf{K}^{-1}, \tag{3}$$

where K is the camera intrinsic matrix, t represents the time from the first frame I_a to the second frame I_b, and $R(t)$ denotes the camera rotation from I_a to I_b.

For a rolling shutter camera that most mobile phones adopt, each scanline of the image is exposed at a slightly different time, as illustrated in Fig 3. Therefore, Eq. (3) is not applicable anymore, since every row of the image should have a different orientation. In practice, it is not necessary to assign each row with a rotation matrix. We group several consecutive rows into a row patch and assign each patch with a rotation matrix. The number of row patches depends on the number of gyroscope readings per frame.

Here, the homography between the n-th row at frame I_a and I_b can be modeled as:

$$\mathbf{H}_n(t) = \mathbf{KR}\left(t_b\right)\mathbf{R}^\top\left(t_a\right)\mathbf{K}^{-1}, \tag{4}$$

where the n is the index of row patches, $\mathbf{H}_n(t)$ denotes the homography of the n-th row patch from I_a to I_b, and $\mathbf{R}\left(t_b\right)\mathbf{R}^\top\left(t_a\right)$ can be computed by accumulating rotation matrices from t_a to t_b.

In our implementation, we regroup the image into 14 patches that compute a homography array containing 14 horizontal homography between two consecutive frames. Furthermore, to avoid the discontinuities across row patches, we convert the array of homography into an array of 4D quaternions [48] and then apply the spherical linear interpolation (SLERP) to interpolate the camera orientation smoothly, yielding a smooth homography array. As shown in Fig 3, we use the homography array to transform every

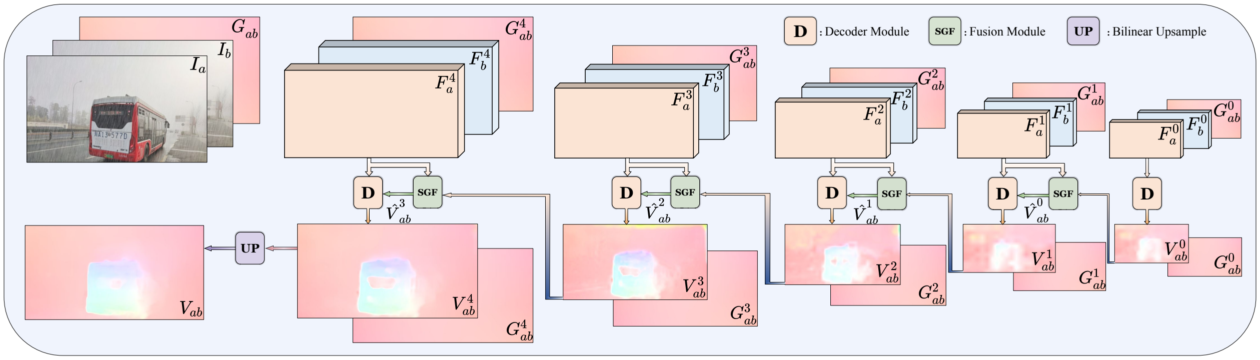

Figure 2. The overview of our algorithm. It consists of a pyramid encoder and a pyramid decoder. For each pair of frames I_a to I_b, our encoder extracts features at different scales. The decoder includes two modules, at each layer l, **SGF** functions to fuse a gyro field G^l_{ab} and an optical flow V^l_{ab} to produce a fused flow $\hat{V^l_{ab}}$ as input to D, which estimates an optical flow to the next layer.

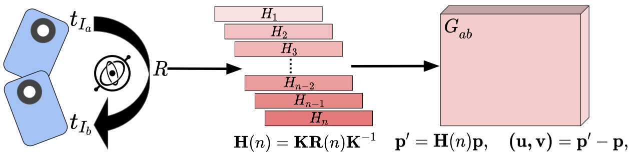
t_{I_a}
t_{I_b}
R
H_1
H_2
H_3
H_{n-2}
H_{n-1}
H_n
G_{ab}
$\mathbf{H}(n) = \mathbf{KR}(n)\mathbf{K}^{-1}$
$\mathbf{p}' = \mathbf{H}(n)\mathbf{p}, \quad (\mathbf{u}, \mathbf{v}) = \mathbf{p}' - \mathbf{p},$

Figure 3. The pipeline of generating gyro field. Given timestamps t_{I_a} and t_{I_b}, gyroscope readings can be read out to compute an array of rotation matrices $R = (R_1 \dots R_n)^\mathsf{T}$. We then convert the rotation array into the homography array that projects pixels p of the first image into p', yielding a gyro field G_{ab}.

pixel p to p', and subtract p' from p as:

$$\mathbf{p}' = \mathbf{H}(\mathbf{n})\mathbf{p}, \quad (\mathbf{u}, \mathbf{v}) = \mathbf{p}' - \mathbf{p}, \tag{5}$$

computing offsets for every pixel produces a gyro field G_{ab}.

3.2. Self-guided Fusion Module

As Fig. 1 illustrates: Fig. 1 (a) denotes the input images. Fig. 1 (b) is the output of the ARFlow [30], an unsupervised optical flow approach, where only the motion of moving objects is roughly produced. As image-based optical flow methods count on image contents for the registration, they are prone to be erroneous in challenging scenes, such as textureless scenarios, dense foggy environments [46], dark [50] and rainy scenes [28]. Fig. 1 (c) represents the ground-truth. To combine the advantages of the gyro field and the image-based optical flow, we propose a self-guided fusion module (SGF). In Fig. 1 (d), with the gyro field, our result is much better compared with the ARFlow [30].

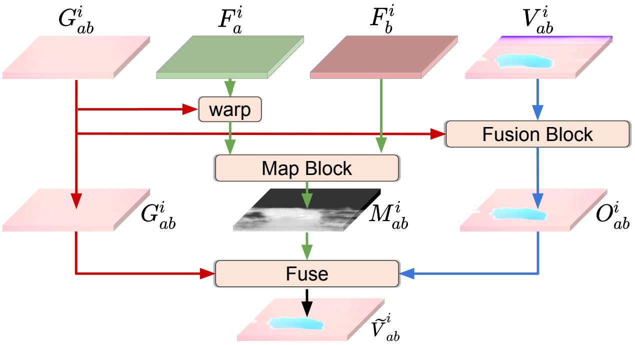

Figure 4. Illustration of our self-guided fusion module (SGF). For a specific layer i, we use 2 blocks to independently produce the fusion map M^i_{ab} and the fusion flow O^i_{ab}, then we generate the output $\widetilde{V}^i_{ab}$ by Eq. 6.

The architecture of our SGF is shown in Fig. 4. Given the input features of image I_a and I_b at the i-th layer as F^i_a and F^i_b. F^i_a is warped by the gyro field G^i_{ab}, which is the forward flow from feature F^i_a to F^i_b. Then the warped feature is concatenated with F^i_b as inputs to the map block, yielding a fusion map M^i_{ab} that ranges from 0 to 1. Note that, in M^i_{ab}, those background regions which can be aligned with the gyro field are close to zeros, while the rest areas are distributed with different weights. Next, we input the gyro field G^i_{ab} and optical flow V^i_{ab} to the fusion block that computes a fusion flow O^i_{ab}. Finally, we fuse the G^i_{ab} and O^i_{ab} with M^i_{ab} to guide the network to focus on the moving foreground regions. The process can be described as:

$$\widetilde{V}^i_{ab} = M^i_{ab} \odot O^i_{ab} + \left(1 - M^i_{ab}\right) \odot G^i_{ab}, \tag{6}$$

where $\widetilde{V}^i_{ab}$ is the output of our SGF module and $\odot$ denotes the element-wise multiplier.

4. Experimental Results

4.1. Dataset

The representative datasets for optical flow estimation and evaluation include FlyingChairs [6], MPI-Sintel [3], KITTI 2012 [7], and KITTI 2015 [38]. On the gyro side, a dedicated dataset embedded with gyroscopes, named GF4 [33], is proposed for the homography estimation. However, none of them combine accurate gyroscope readings with image contents to evaluate the optical flow. Therefore, we propose a new dataset and benchmark: GOF.

Training Set Similar to GF4 [33], a set of videos with gyroscope readings are recorded using a cellphone. Compared

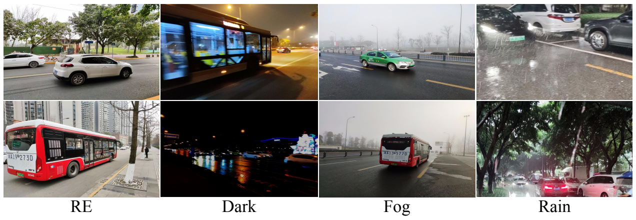

Figure 5. A glance at our evaluation set. It can be divided into 4 categories, including regular scenes(RE), low light scenes(Dark), foggy scenes(Fog), and rainy scenes(Rain). Each category contains 70 pairs, and a total of 280 pairs evaluation dataset is proposed with synchronized gyroscope readings.

to GF4, which uses a phone with an OIS camera. We carefully choose a non-OIS camera phone to eliminate the effect of the OIS module. We collect videos in 4 different environments, including regular scenes (RE), low light scenes (Dark), foggy scenes (Fog), and rainy scenes (Rain). For each scenario, we record a set of videos lasting for 60 seconds, yielding 1, 800 frames under every environment. In total, we collect 5, 000 frames for the training set.

Evaluation Set For evaluation, similar to the train set, we capture videos in 4 scenes to compare to image-based registration methods. Each category contains 70 pairs, yielding a 280 pairs evaluation set. Fig. 5 shows some examples.

For quantitative evaluation, a ground-truth optical flow is required for each pair. However, labeling ground-truth flow is non-trivial. As far as we know, no powerful tool is available for this task. Following [28, 46], we adopt the most related approach [29] to label the ground-truth flow with many efforts. It costs approximately 20 $\sim$ 30 minutes per image, especially for challenging scenes. We firstly label an amount of 500 examples containing rigid objects, then we select those with good visual performance, i.e., the performance of image alignment, and discard the others. Furthermore, we refine the selected samples with detailed modifications around the motion boundaries.

To verify the effectiveness of our labeled optical flow, we choose to label several samples from KITTI 2012 [7]. Given the ground-truth, we compare our labeled optical flow with results produced by the state-of-the-art supervised method, i.e., RAFT [43] pre-trained on FlyingChairs. Our labeled flow computes an endpoint error (EPE) of 0.7, where RAFT computes an EPE of 2.4, which is more than 3 times larger than ours. Fig. 6 shows one example. As the error map illustrates, our labeled flow is much more accurate than the current SOTA method. We leverage this approach to generate ground-truth for evaluations.

4.2. Implementation Details

We conduct experiments on GOF dataset. Our method is built upon the PWC-Net [42]. For the first stage, we train our model for 100k steps without the occlusion mask. For the second stage, we enable the bidirectional occlusion mask [37], the census loss [37], and the spatial transform [30] to fine-tune the model for about 300k steps.

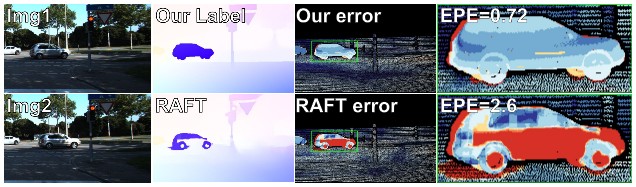

Figure 6. One label example on KITTI 2012 [7], compared to RAFT[43](the second line) that computes an EPE equals 2.6, our label flow(the first line) produces a 0.72 EPE. From the error map, we notice that our labeled optical flow is much more accurate.

We collect videos with gyroscope readings using Qualcomm QRD equipped with Snapdragon 7150, which records videos in 600 × 800 resolution. We add random crop, random horizontal flip, and random weather modification (add fog and rain [20]) during the training. We report endpoint error (EPE) in the evaluation set. The implementation is in PyTorch, and one NVIDIA RTX 2080 Ti is used to train our network. We use Adam optimizer [22] with parameters setting as $LR = 1.0 \times 10^{-4}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1.0 \times 10^{-7}$. The batch size is 4. It takes 3 days to finish the entire training process. On one single 1080ti, the time to generate an optical flow is 58ms per frame. Same to previous work [19, 36], we use the photometric loss and smooth term to train the network.

4.3. Comparisons with Image-based Methods

In this section, we compare our method with traditional, supervised, and unsupervised methods on GOF evaluation set with quantitative (Sec. 4.3.1) and qualitative comparisons (Sec. 4.3.2). To validate the effectiveness of key components, we conduct an ablation study in Sec.4.4.

4.3.1 Quantitative Comparisons

In Table 1, the best results are marked in red, and the second-best results are in blue. The percentage in the bracket indicates the improvements over the second-best results. Therefore, the percentage of the best results is negative. The second best is all zeros while the others are positive. '$I_{3\times3}$' refers to no alignment, and 'Gyro Field' refers to alignment with pure gyro data.

For traditional methods, we compare our GyroFlow with DIS [23] and DeepFlow [45] pre-trained on Sintel [3] (Table 1, 3∼4). As seen, their average EPEs are 4 times larger than ours. In particular, DIS fails in foggy scenarios, and DeepFlow crashes in rainy scenes. Moreover, we try to implement the traditional gyroscope-based optical flow method [27] because no replies are received from the authors. Due to the lack of implementation details, we do not get reasonable results, so they are not reported.

Next, we compare with deep supervised optical flow

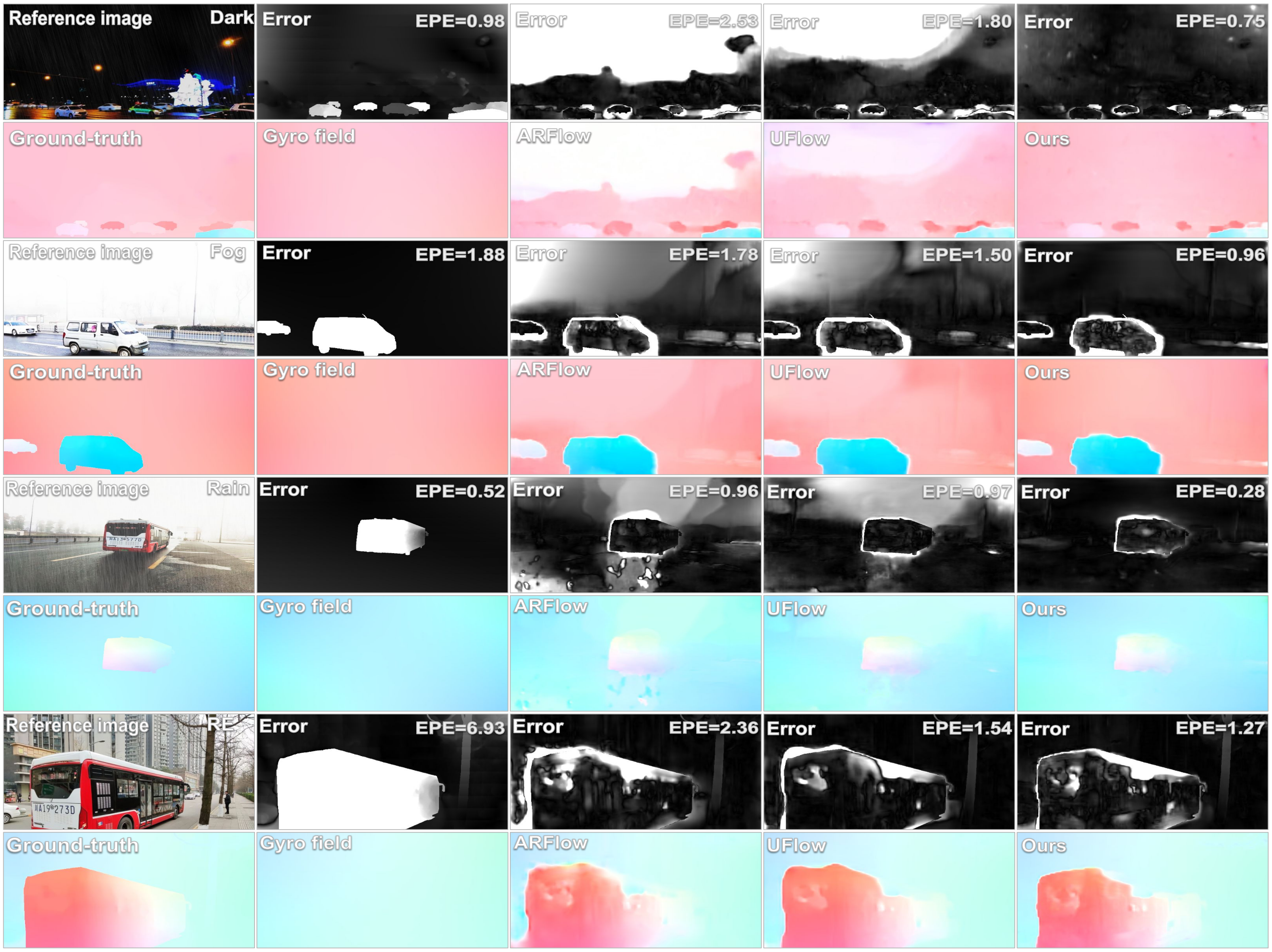

Figure 7. Visual comparison of our method with gyro field, ARFlow [30], and UFlow [19] on the GOF evaluation set. For the first 3 challenging cases, we notice that our method achieves convincing results by fusing the background motion from the gyro field and the motion details from the optical flow. For the last example, in regular scenarios, fusing gyro field helps the learning of optical flow where the network produces accurate and sharp flow around the boundary of objects.

methods, including FlowNet2 [6], IRRPWC [13], SelFlow [32], and recent state-of-the-art method RAFT [43] (Table 1, line $5 \sim 8$). For the lack of ground-truth labels during training, we cannot refine these methods on our trainset. So for each method, we search different pre-trained models and test them on the evaluation set. Here, we only report the best results. RAFT pre-trained on FlyingChairs [6] performs the best, but it is still not as good as ours.

We also compare our method to deep unsupervised optical flow methods, including DDFlow [31], UnFlow [37], ARFlow [30] and UFlow [19] (Table 1, $9 \sim 12$). Here, we refine the models on our training set. UFlow achieves 3 second-best results. However, it is still not comparable with ours due to the unstable performance in challenging scenes.

As discussed in Sec. 2, RainFlow [28] is designed to estimate the optical flow under rainy scenes. FogFlow [46] aims for foggy environments, and DarkFlow [50] intends to compute flows in the low-light scenarios. We also compare these methods. Note that all these methods are not open source. For DarkFlow [50], the authors do not provide source codes but offer a pre-trained version on FlyingChairs, the result is reported at line 13 in Table 1. For the other two methods [28, 46], no replies are received from the authors. We try to implement them, but the results are not satisfactory due to the uncertainty of some implementation details. Therefore, results are not illustrated in Table 1.

We find that our GyroFlow model is robust in all scenes and computes a 0.717 EPE error which is 26.46% better than the second-best method on average. Notably, for 'Dark' scenes that consist of poor image texture, the 'Gyro Field' alone achieves the second-best performance, indicating the importance of incorporating gyro motion, especially when the image contents are not reliable.

For the further comparison to supervised methods, we

Method	RE	Dark	Fog	Rain	Avg
1) $\mathcal{I}_{3\times3}$	4.962(+457.53%)	3.278(+228.13%)	7.358(+643.23%)	5.567(+425.68%)	5.665(+481.03%)
2) Gyro Field	2.583(+190.22%)	0.999(+0.00%)	1.279(+29.19%)	1.703(+60.81%)	1.922(+97.13%)
3) DIS [23]	2.374(+166.74%)	2.442(+144.44%)	4.677(+372.42%)	3.004(+183.66%)	3.399(+248.62%)
4) DeepFlow [45] - Sintel[3]	3.521(+295.62%)	3.425(+242.84%)	3.029(+205.96%)	11.812(+1015.39%)	4.858(+398.26%)
5) FlowNet2 [6] - Sintel[3]	11.140(+1151.69%)	44.641(+4368.57%)	2.633(+165.96%)	5.767(+444.57%)	6.701(+587.28%)
6) IRRPWC [13] - FlyingChairs [6]	12.487(+1303.03%)	69.864(+6893.39%)	1.916(+93.54%)	9.799(+825.31%)	8.234(+744.51%)
7) SelFlow[32] - Sintel [3]	4.186(+370.34%)	2.747(+174.97%)	7.307(+638.08%)	4.787(+352.03%)	5.626(+477.03%)
8) RAFT [43] - FlyingChairs[6]	1.246(+40.00%)	1.297(+29.83%)	1.136(+14.75%)	1.187(+12.09%)	1.349(+38.36%)
9) DDFlow [31] - GOF	2.273(+155.39%)	2.843(+184.58%)	3.070(+210.10%)	2.422(+128.71%)	2.527(+159.18%)
10) UnFlow [37] - GOF	1.120(+25.84%)	1.671(+67.17%)	0.990(+0%)	1.343(+26.53%)	1.221(+25.13%)
11) ARFlow [30] - GOF	0.972(+9.21%)	1.205(+20.62%)	1.186(+19.80%)	1.093(+3.21%)	1.035(+6.15%)
12) UFlow [19] - GOF	0.890(+0.00%)	1.641(+64.26%)	0.994(+0.40%)	1.059(+0.00%)	0.975(+0.00%)
13) DarkFlow [50] - FlyingChairs[6]	4.127(+363.71%)	4.346(+335.04%)	7.316(+638.99%)	4.891(+361.85%)	5.758(+490.56%)
14) Ours	0.742(−16.63%)	0.902(−9.71%)	0.658(−33.54%)	0.730(−31.07%)	0.717(−26.46%)

Table 1. Quantitative comparisons on the evaluation dataset. We mark the best performance in red and the second-best in blue. The percentage in the bracket indicates the improvements over second-best results. We use '-' to indicate which dataset the model is trained on.

expand the evaluation set to 400 pairs, then it is divided into 2 parts, GOF-clean (for training) and GOF-final (for testing). We pre-train the supervised methods on FlyingChairs [6], then fine-tune them on GOF-clean. We also fine-tune UFlow [19] and GyroFlow on GOF-clean. Results evaluated on GOF-final are shown in Table 2. As seen, for unsupervised methods, we are better than UFlow. For supervised methods, we are better than FlowNet2 [6] and IRRPWC [13]. RAFT [43] achieves the best on average. Note that, supervised methods have label guidance during the entire training while we do not.

Model	RE	Dark	Fog	Rain	Avg
FlowNet2 [6] - GOF-clean	0.67	4.74	5.21	3.73	3.36
IRRPWC [13] - GOF-clean	0.64	5.00	5.00	4.40	3.62
RAFT [43] - GOF-clean	0.14	1.20	0.88	1.33	0.74
UFlow [19] - GOF-clean	0.72	3.54	1.50	3.51	2.37
Ours	0.64	2.50	0.55	3.03	1.78

Table 2. Comparisons on GOF-final (200 pairs). We use '-' to indicate which dataset the model is fine-tune on.

4.3.2 Qualitative Comparisons

In Fig. 7, we illustrate the qualitative results on the evaluation set. We choose one example for each of four different scenes, including the low-light scene (Dark), the foggy scene (Fog), the rainy scene (Rain), and the regular scene (RE). To compare methods, we choose the gyro field and 2 recent unsupervised methods, i.e., ARFlow [30] and UFlow [19] which are refined on our training set. In Fig. 7, we show optical flow along with corresponding error maps and also report the EPE error for each example. As shown, for challenge cases, our method can fuse the background motion from the gyro field with the motion of dynamic objects from the image-based optical flow, delivering both better visual quality and lower EPE errors.

The unsupervised optical flow methods [31, 30, 19] are supposed to work well in RE scenes given sufficient texture. However, we notice that, even for the RE category, our method outperforms the others, especially at the motion boundaries. With the help of the gyro field that solves the global motion, the network can focus on challenging regions. As a result, our method still achieves better visual quality and produces lower EPE errors in RE scenarios.

Method	RE	Dark	Fog	Rain	Avg
DWI	3.77	3.15	5.59	4.24	4.38
DPGF	0.95	1.67	1.32	0.89	0.98
SGF-Fuse	0.72	0.99	0.99	0.94	0.80
SGF-Map	1.07	1.02	1.19	0.70	0.90
SGF-Dense	0.77	1.69	0.87	1.00	0.89
GyroFlow without SGF	0.79	1.71	1.35	1.06	0.95
Our SGF	0.74	0.90	0.66	0.73	0.72

Table 3. Comparison with alternative designs of the SGF module.

4.4. Ablation Studies

To evaluate the effectiveness of the design for each module, we conduct ablation experiments on the evaluation set. EPE errors are reported under 5 categories, including Dark, Fog, Rain, and RE, along with the average error.

4.4.1 The Design of SGF

For SGF, we test several designs and report results in Table 3. First of all, two straightforward methods are adopted to build the module. DWI refers that we directly warp the I_a with gyro field, then we input the warped image and I_b to produce a residual optical flow. DPGF denotes that, for each pyramid layer, we directly add the gyro field onto the optical flow. As shown in Table 3, for DWI, the result is not good. Except for the absence of gyroscope guidance during training, another possibility is that the warping oper-

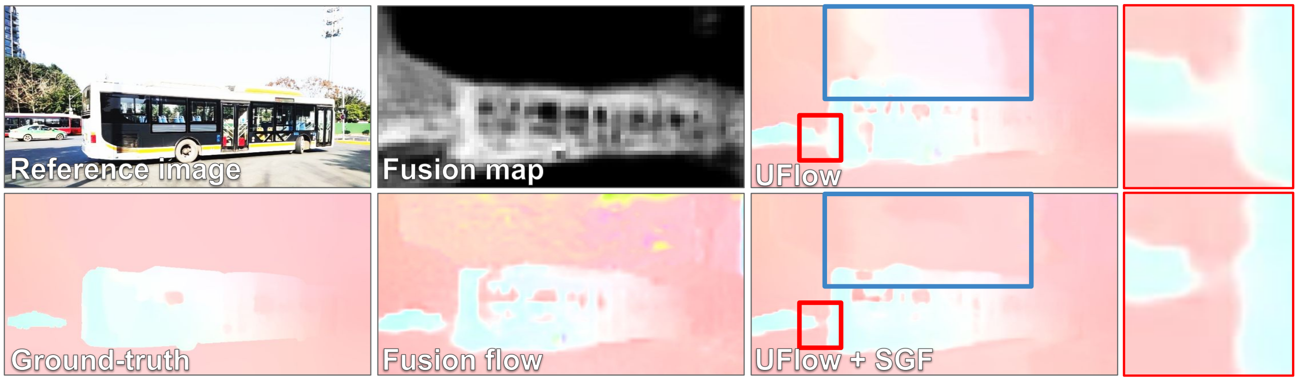

Figure 8. Visual example of our self-guided fusion module(SGF). Results of UFlow and UFlow with SGF are shown. The fusion map is used to guide the network to focus on motion details.

Model	RE	Dark	Fog	Rain	Avg
UnFlow [37]	1.12	1.67	0.99	1.34	1.22
UnFlow [37] + SGF	0.83	1.33	0.94	0.94	0.90
ARFlow [30]	0.97	1.21	1.19	1.09	1.04
ARFlow [30] + SGF	0.77	1.54	0.85	0.94	0.86
UFlow [19]	0.89	1.64	0.99	1.06	0.98
UFlow [19] + SGF	0.89	0.95	0.71	0.78	0.80
Our baseline	0.79	1.71	1.35	1.06	0.95
Ours	0.74	0.90	0.66	0.73	0.72

Table 4. Comparison with unsupervised methods when equipped with our SGF module.

ation breaks the image structure such as blurring and noising. DPGF gets a better result but is still not comparable to our SGF design because the gyro field registers background motion that should not be concatenated to dynamic object motion. Furthermore, we compare our SGF with three variants: (1) SGF-Fuse, we remove the map block, and the final fusion procedure. Although it computes a 0.8 EPE error, it performs unstable in challenging scenes; (2) SGF-Map, where the fusion block is removed. It results in worse performance because the fusion map M_{ab} tends to be inaccurate except for the rainy scene. (3) SGF-Dense, we integrate the two blocks into one unified dense block, which produces a 3 channels tensor of which the first two channels represent the fusion flow O_{ab}, and the last channel denotes the fusion map M_{ab}. Our SGF is much better on average.

4.4.2 Unsupervised Methods with SGF.

We insert the SGF module into unsupervised methods [37, 30, 19], and the baseline represents our GyroFlow without SGF. In particular, similar to Fig. 2, we add the SGF before the decoder **D** for each pyramid layer. Several unsupervised methods are trained on our dataset, and we report EPE errors in Table 4. After inserting our SGF module into these models, noticeable improvements can be observed in Table 1 and Table 4, which proves the effectiveness of our proposed SGF module. Fig. 8 shows an example. Both background motion and boundary motion are improved after integrating our SGF.

Pyramid Layer	RE	Dark	Fog	Rain	Avg
Baseline	0.79	1.71	1.35	1.06	0.95
1/32 resolution	1.03	1.04	0.85	1.03	0.95
1/16 resolution	0.94	0.98	0.95	0.93	0.92
1/8 resolution	0.89	1.17	1.19	0.87	0.89
1/4 resolution	0.81	1.13	0.94	0.91	0.87
All resolutions	0.74	0.90	0.66	0.73	0.72

Table 5. Adding gyro filed to different pyramid layers. The baseline indicates GyroFlow without SGF.

4.4.3 Gyro Field Fusion Layer

Intuitively, it is also possible to fuse the gyro field only once during the training, so we add our SGF module to a specific pyramid layer. As illustrated in Table 5, we notice that the more bottom layer we add SGF to, the lower EPE error it produces. The best results can only be obtained when we add the gyro field at all layers.

5. Conclusion

We have presented a novel framework GyroFlow for unsupervised optical flow learning by fusing the gyroscope data. We have proposed a self-guided fusion module to fuse the gyro field and optical flow. For the evaluation, we have proposed a dataset GOF and labeled 400 ground-truth optical flow for quantitative metrics. The results show that our proposed method achieves state-of-art in all regular and challenging categories compared to the existing methods.

Acknowledgement: This research was supported in part by National Natural Science Foundation of China (NSFC) under grants No.61872067 and No.61720106004, in part by Research Programs of Science and Technology in Sichuan Province under grant No.2019YFH0016.

References

[1] Aseem Behl, Omid Hosseini Jafari, Siva Karthik Mustikovela, Hassan Abu Alhaija, Carsten Rother, and Andreas Geiger. Bounding boxes, segmentations and object coordinates: How important is recognition for 3d scene flow estimation in autonomous driving scenarios? In *Proc. ICCV*, pages 2574–2583, 2017. 1

[2] Michael Bloesch, Sammy Omari, Péter Fankhauser, Hannes Sommer, Christian Gehring, Jemin Hwangbo, Mark A Hoepflinger, Marco Hutter, and Roland Siegwart. Fusion of optical flow and inertial measurements for robust egomotion estimation. In *Proc. IROS*, pages 3102–3107, 2014. 1, 2, 3

[3] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Proc. ECCV*, pages 611–625, 2012. 1, 2, 4, 5, 7

[4] Jason Campbell, Rahul Sukthankar, and Illah Nourbakhsh. Techniques for evaluating optical flow for visual odometry in extreme terrain. In *Proc. IROS*, pages 3704–3711, 2004. 1

[5] Jian S Dai. Euler–rodrigues formula variations, quaternion conjugation and intrinsic connections. *Mechanism and Machine Theory*, 92:144–152, 2015. 3

[6] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. ICCV*, pages 2758–2766, 2015. 1, 2, 4, 6, 7

[7] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proc. CVPR*, pages 3354–3361, 2012. 2, 4, 5

[8] Hari Prabhat Gupta, Haresh S Chudgar, Siddhartha Mukherjee, Tanima Dutta, and Kulwant Sharma. A continuous hand gestures recognition technique for human-machine interaction using accelerometer and gyroscope sensors. *IEEE Sensors Journal*, 16(16):6425–6432, 2016. 1

[9] Dennis Guse and Benjamin Müller. Gesture-based user authentication on mobile devices using accelerometer and gyroscope. In *Informatiktage*, pages 243–246, 2012. 2

[10] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 1

[11] Weibo Huang and Hong Liu. Online initialization and automatic camera-imu extrinsic calibration for monocular visual-inertial slam. In *Proc. ICRA*, pages 5182–5189, 2018. 2

[12] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proc. CVPR*, pages 8981–8989, 2018. 2

[13] Junhwa Hur and Stefan Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *Proc. CVPR*, pages 5754–5763, 2019. 2, 3, 6, 7

[14] Myung Hwangbo, Jun-Sik Kim, and Takeo Kanade. Inertial-aided klt feature tracking for a moving camera. In *Proc. IROS*, pages 1909–1916, 2009. 1, 3

[15] Woobin Im, Tae-Kyun Kim, and Sung-Eui Yoon. Unsupervised learning of optical flow with deep feature similarity. In *Proc. ECCV*, pages 172–188, 2020. 2

[16] Joel Janai, Fatma Guney, Anurag Ranjan, Michael Black, and Andreas Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *Proc. ECCV*, pages 690–706, 2018. 2

[17] 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 *Proc. ECCV*, pages 3–10, 2016. 2

[18] Chao Jia and Brian L Evans. Online calibration and synchronization of cellphone camera and gyroscope. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 731–734, 2013. 2

[19] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. In *Proc. ECCV*, pages 557–572, 2020. 2, 3, 5, 6, 7, 8

[20] Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, and *et al.* imgaug. https://github.com/aleju/imgaug, 2020. Online; accessed 01-Feb-2020. 5

[21] Alexandre Karpenko, David Jacobs, Jongmin Baek, and Marc Levoy. Digital video stabilization and rolling shutter correction using gyroscopes. *CSTR*, 1(2):13, 2011. 2, 3

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[23] Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. Fast optical flow using dense inverse search. In *Proc. ECCV*, pages 471–488, 2016. 1, 5, 7

[24] László Kundra and Péter Ekler. Bias compensation of gyroscopes in mobiles with optical flow. *Aasri Procedia*, 9:152–157, 2014. 3

[25] Fabrizio La Rosa, Maria Celvisia Virzì, Filippo Bonaccorso, and Marco Branciforte. Optical image stabilization (ois). 2015. STMicroelectronics. Available online; accessed on 31 October 2015. 2

[26] Robert Patton Leland. Adaptive control of a mems gyroscope using lyapunov methods. *IEEE Trans. on Control Systems Technology*, 14(2):278–283, 2006. 1

[27] Ping Li and Hongliang Ren. An efficient gyro-aided optical flow estimation in fast rotations with auto-calibration. *IEEE Sensors Journal*, 18(8):3391–3399, 2018. 1, 3, 5

[28] Ruoteng Li, Robby T Tan, Loong-Fah Cheong, Angelica I Aviles-Rivero, Qingnan Fan, and Carola-Bibiane Schonlieb. Rainflow: Optical flow under rain streaks and rain veiling effect. In *Proc. ICCV*, pages 7304–7313, 2019. 2, 4, 5, 6

[29] Ce Liu, William T Freeman, Edward H Adelson, and Yair Weiss. Human-assisted motion annotation. In *Proc. CVPR*, pages 1–8, 2008. 2, 5

[30] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *Proc. CVPR*, pages 6489–6498, 2020. 1, 2, 4, 5, 6, 7, 8

[31] Pengpeng Liu, Irwin King, Michael R Lyu, and Jia Xu. Ddflow: Learning optical flow with unlabeled data distillation. In *Proc. AAAI*, pages 8770–8777, 2019. 2, 6, 7

[32] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *Proc. CVPR*, pages 4571–4580, 2019. 6, 7

[33] Shuaicheng Liu, Haipeng Li, Zhengning Wang, Jue Wang, Shuyuan Zhu, and Bing Zeng. Deepois: Gyroscope-guided deep optical image stabilizer compensation. *IEEE Trans. on Circuits and Systems for Video Technology*, DOI: 10.1109/TCSVT.2021.3103281, 2021. 2, 4

[34] Shuaicheng Liu, Kunming Luo, Nianjin Ye, Chuan Wang, Jue Wang, and Bing Zeng. Oiflow: Occlusion-inpainting optical flow estimation by unsupervised learning. *IEEE Trans. on Image Processing*, 30:6420–6433, 2021. 2

[35] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. In *Proc. IJCAI*, 1981. 1, 2

[36] Kunming Luo, Chuan Wang, Shuaicheng Liu, Haoqiang Fan, Jue Wang, and Jian Sun. Upflow: Upsampling pyramid for unsupervised optical flow learning. In *Proc. CVPR*, pages 1045–1054, 2021. 1, 2, 3, 5

[37] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Proc. AAAI*, 2018. 2, 5, 6, 7, 8

[38] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proc. CVPR*, pages 3061–3070, 2015. 2, 4

[39] Janne Mustaniemi, Juho Kannala, Simo Särkkä, Jiri Matas, and Janne Heikkila. Gyroscope-aided motion deblurring with deep networks. In *Proc. WACV*, pages 1914–1922, 2019. 2, 3

[40] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proc. CVPR*, pages 4161–4170, 2017. 1, 2

[41] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. In *Proc. AAAI*, 02 2017. 1, 2

[42] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proc. CVPR*, pages 8934–8943, 2018. 2, 3, 5

[43] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proc. ECCV*, pages 402–419, 2020. 1, 2, 5, 6, 7

[44] Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, Peng Wang, and Wei Xu. Occlusion aware unsupervised learning of optical flow. In *Proc. CVPR*, pages 4884–4893, 2018. 2

[45] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proc. ICCV*, pages 1385–1392, 2013. 5, 7

[46] Wending Yan, Aashish Sharma, and Robby T Tan. Optical flow in dense foggy scenes using semi-supervised learning. In *Proc. CVPR*, pages 13259–13268, 2020. 3, 4, 5, 6

[47] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proc. CVPR*, pages 1983–1992, 2018. 2

[48] Fuzhen Zhang. Quaternions and matrices of quaternions. *Linear algebra and its applications*, 251:21–57, 1997. 3

[49] Rong Zhang, Christian Vogler, and Dimitris Metaxas. Human gait recognition. In *Proc. CVPRW*, pages 18–18, 2004. 2

[50] Yinqiang Zheng, Mingfang Zhang, and Feng Lu. Optical flow in the dark. In *Proc. CVPR*, pages 6749–6757, 2020. 2, 4, 6, 7

[51] Yiran Zhong, Pan Ji, Jianyuan Wang, Yuchao Dai, and Hongdong Li. Unsupervised deep epipolar flow for stationary or dynamic scenes. In *Proc. CVPR*, pages 12095–12104, 2019. 2