RealFlow: EM-based Realistic Optical Flow Dataset Generation from Videos

Yunhui Han[1], Kunming Luo[2], Ao Luo[2], Jiangyu Liu[2], Haoqiang Fan[2], Guiming Luo[1], and Shuaicheng Liu[3,2,†]

[1] School of Software, Tsinghua University, Beijing 100084, China
{hanyh19@mails.tsinghua.edu.cn, gluo@tsinghua.edu.cn}
[2] Megvii Technology, Beijing, China
{luokunming,luoao02,liujiangyu,fhq}@megvii.com
[3] University of Electronic Science and Technology of China, Chengdu, China
liushuaicheng@uestc.edu.cn
†Corresponding Author

Abstract. Obtaining the ground truth labels from a video is challenging since the manual annotation of pixel-wise flow labels is prohibitively expensive and laborious. Besides, existing approaches try to adapt the trained model on synthetic datasets to authentic videos, which inevitably suffers from domain discrepancy and hinders the performance for real-world applications. To solve these problems, we propose RealFlow, an Expectation-Maximization based framework that can create large-scale optical flow datasets directly from any unlabeled realistic videos. Specifically, we first estimate optical flow between a pair of video frames, and then synthesize a new image from this pair based on the predicted flow. Thus the new image pairs and their corresponding flows can be regarded as a new training set. Besides, we design a Realistic Image Pair Rendering (RIPR) module that adopts softmax splatting and bi-directional hole filling techniques to alleviate the artifacts of the image synthesis. In the E-step, RIPR renders new images to create a large quantity of training data. In the M-step, we utilize the generated training data to train an optical flow network, which can be used to estimate optical flows in the next E-step. During the iterative learning steps, the capability of the flow network is gradually improved, so is the accuracy of the flow, as well as the quality of the synthesized dataset. Experimental results show that RealFlow outperforms previous dataset generation methods by a considerably large margin. Moreover, based on the generated dataset, our approach achieves state-of-the-art performance on two standard benchmarks compared with both supervised and unsupervised optical flow methods. Our code and dataset are available at https://github.com/megvii-research/RealFlow.

1 Introduction

Deep optical flow methods [47,46] adopt large-scale datasets to train networks, which have achieved good computational efficiency and state-of-the-art performances in public benchmarks [34,3]. One key ingredient of these deep learning

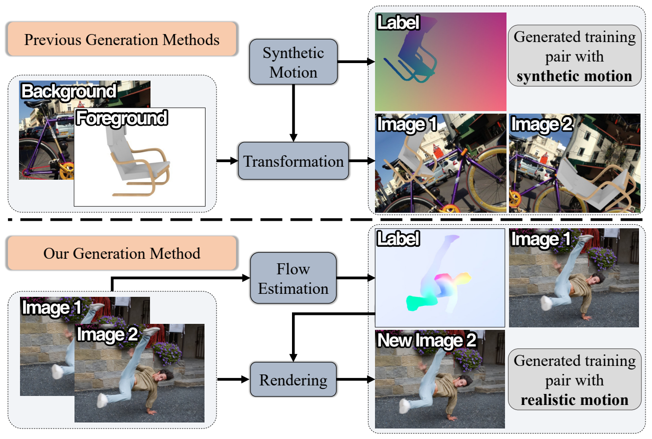

Fig. 1. Illustration of our motivation. Top: previous methods use synthetic motion to produce training pairs. Bottom: we propose to construct training pairs with realistic motion labels from the real-world video sequence. We estimate optical flow between two frames as the training label and synthesize a 'New Image 2'. Both the new view and flow labels are refined iteratively in the EM-based framework for mutual improvements.

methods is the training dataset. We summarize **four** key characteristics of flow datasets that have significant impacts on the success of deep learning algorithms: **1)** the *quantity* of labeled pairs; **2)** the *quality* of flow labels; **3)** the *image realism*; and **4)** the *motion realism*. We refer to the first two as the label criteria and the latter two as the realism criteria.

However, we find it is difficult for existing flow datasets to be satisfactory in all aspects. For example, FlyingThings [31] synthesizes the flows by moving a foreground object on top of a background image. Sintel [3] is purely rendered from virtual 3D graphic animations. AutoFlow [44] presents a learning approach searching for hyperparameters to render synthetic training pairs. As a result, these methods can produce large amounts of training data with accurate flow labels, satisfying the label criteria. However, they failed to meet the demand of realism criteria, as both the scene objects and their motions are synthesized. If flow networks are trained on these datasets, they may suffer from the domain gap between the synthetic and authentic scenes [18], resulting in sub-optimal performance on real-world images.

To achieve realism, some methods propose to manually annotate flow labels using realistic videos [2,21]. Although these methods can naturally satisfy the realism criteria, the process of manual labeling is time-consuming, and neither the quality nor the quantity can be guaranteed, potentially at odds with the requirements for label criteria. Recently, Aleotti *et al.* [1] propose to create training pairs from a single image. It randomly generates transformations as the flow labels, based on which the image is warped to produce the other image, yielding a pair of images together with the flow labels. In this way, the label

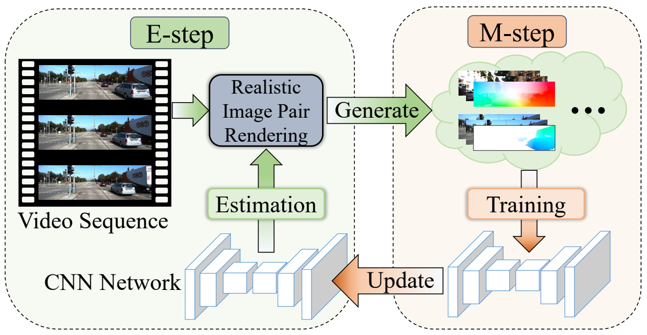

Fig. 2. Illustration of the EM-based framework. In the E-step, we estimate flow labels and synthesize new views to generate training data. In the M-step, we use the training data to train a network, which can update flow labels for the next E-step.

criteria are satisfied. However, the realism criteria can only be satisfied partially. Because, the synthesized images sometimes contain artifacts, and more importantly, the generated motions cannot resemble realistic object motion behaviors of real-world scenarios.

To address these issues, we propose RealFlow, an iterative learning framework to simultaneously generate the training pairs from realistic video frames and obtain an enhanced flow network with the generated flow data, which can satisfy both the label and realism criteria. Fig. 1 shows an illustration of our RealFlow in comparison with existing methods. Previous work [44,5] (Fig. 1 top) synthesizes the flows (motions) by pasting foreground objects on top of backgrounds at different positions, where the motions are manually generated. In our approach (Fig. 1 bottom), we first estimate the optical flows F between frame pairs (I_1 and I_2) from existing videos, and then exploit the predicted flows as labels to synthesize I_2', a set of new images of the 'frame 2'. After that, we abandon the original I_2, and use the I_1 and I_2' as image pairs, together with the estimated F as the flow labels to compose a sample (I_1, I_2', F) of the new optical flow dataset. Note that the flow labels are naturally accurate for warping I_1 to I_2', since the pixels in I_2' are synthesized based on the F and I_1.

This strategy faces two challenges: 1) image synthesis may introduce artifacts, *e.g.,* disparity occlusions; 2) the motion realism is affected by the quality of estimated flows. For the first challenge, we design Realistic Image Pair Rendering (RIPR) method to robustly render a new image I_2'. Specifically, we employ softmax splatting and bi-directional hole filling techniques, based on the flow and depth maps predicted from the image pairs I_1 and I_2, to generate the new images, where most of the artifacts can be effectively alleviated. For the second one, we design an Expectation-Maximization (EM) based learning framework, as illustrated in Fig. 2. Specifically, during the E-step, RIPR renders new images to create the training samples, and during M-step, we use the generated data to train the optical flow network that will estimate optical flows for the next

E-step. During the iterative learning steps, the capability of the flow network is gradually improved, so is the accuracy of the flow, as well as the quality of synthesized dataset. Upon the convergence of our EM-based framework, we can obtain a new flow dataset generated from the input videos and a high-precision optical flow network benefiting from the new training data.

By applying RealFlow, huge amounts of videos can be used to generate training datasets, which allows supervised optical flow networks to be generalized to any scene. In summary, our main contributions are:

– We propose **RealFLow**, an EM-based iterative refinement framework, to effectively generate large-scale optical flow datasets with realistic scene motions and reliable flow labels from real-world videos.
– We present Realistic Image Pair Rendering (**RIPR**) method for high-quality new view synthesis, overcoming issues such as occlusions and holes.
– RealFlow leads to a significant performance improvement compared against prior dataset generation methods. We generate a large real-world dataset, with which we set new records on the public benchmarks using widely-used optical flow estimation methods.

2 Related Work

Supervised Optical Flow Network. FlowNet [5] is the first work to estimate optical flow by training a convolutional network on synthetic dataset. Following FlowNet, early approaches [11,38,10,9] improve the flow accuracy with the advanced modules and network architectures. Recent works [28,16,27] propose graph and attention-based global motion refinement approaches in the recurrent framework [47], making large progress on supervised learning. However, for the existing supervised networks, the domain gap between synthetic datasets and realistic datasets is non-negligible and inevitably degrades the performance. Our work aims to generate datasets from realistic videos to solve this problem.
Unsupervised Optical Flow Network. The advantage of unsupervised methods is that no annotations are required for the training [15,40]. Existing works [24,52,23,39,19,25] present multiple unsupervised losses and image alignment constraints to achieve competitive results. However, there are many challenges for unsupervised methods, including but not limited to, occlusions [48,13,26], lack of textures [12], and illumination variations [33], all of which break the basic hypothesis of brightness constancy assumption. Therefore, supervised networks achieve better performance than unsupervised ones.
Dataset Generation for Optical Flow. Middlebury [2] records objects with fluorescent texture under UV light illumination to obtain flow labels from real-world scenes. Liu. *et al.* [21] propose a human-in-loop methodology to annotate ground-truth motion for arbitrary real-world videos. KITTI [7,34] is a popular autonomous driving dataset, which provides sophisticated training data through complex device setups. However, the quantity of the above real-world datasets is small, which is insufficient for deep supervised learning. Flyingchairs [5] makes

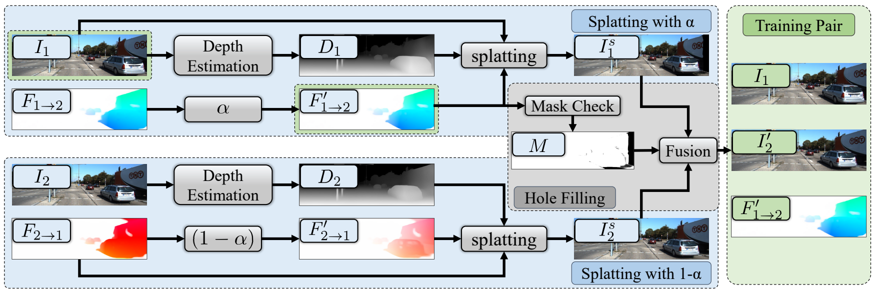

Fig. 3. Illustration of Realistic Image Pair Rendering (RIPR). Given two frames I_1 and I_2 from real-world videos and their estimated flow fields $F_{1\to2}$ and $F_{2\to1}$, we first obtain the depth D_1 and D_2 by the monocular depth network[36]. Then, we modify the flow maps to $F'_{1\to2}$ and $F'_{2\to1}$ and use $F'_{1\to2}$ to check the hole mask M. Finally, splatting method is used to generate new views I_1^s and I_2^s, which are further fused to render 'new image 2' I'_2. The (I_1,I'_2,F) serves as the new generated training pair.

the first attempt to show that synthesized training pairs can be used for supervised learning. Flyingthings [31] further improves the quantity. Virtual KITTI [6] uses Unity game engine to create a large driving dataset. AutoFlow [44] presents a learning approach to search the hyperparameter for rendering training data. However, these datasets are all synthetic. There is a constant shift from synthetic scene towards real-world scene. SlowFlow [14] attempts to collect large-scale dataset using a high-speed video camera, but the flow labels are not totally reliable. Efforts to tackle the above problem is Depthstillation [1], which synthesizes a new image from a single real image. The motion labels are the sampled parametric transformations for the foreground and background. However, the sampled motions are not real, and the synthesis sometimes introduces artifacts. In contrast, our method obtains reliable flow labels from real videos and synthesizes the new view from two frames instead of a single image.

3 Method

3.1 RealFlow Framework

The pipeline of the proposed RealFlow framework is illustrated in Fig. 2. Given a set of real-world videos, our goal is to generate a large-scale training dataset and learn an optical flow estimation network at the same time. The key idea behind RealFlow is that a better training dataset can help learn a better optical flow network and inversely, a better network can provide better flow predictions for dataset generation. Therefore, we integrate the dataset generation procedure and optical flow network training procedure as a generative model, which can be iteratively optimized by Expectation-Maximization (EM) algorithm [32].

As illustrated in Fig. 2, RealFlow is an iterative framework that contains two main steps: E-step and M-step. In iteration t, we first conduct the E-step to generate a training dataset $X^t = \{x^t\}$. Given a consecutive image pair (I_1, I_2) sampled from the input videos, the training data generation procedure can be formulated as follows:

$$x^t = \mathcal{R}(I_1, I_2, \Theta^{t-1}), \tag{1}$$

where Θ^{t-1} is the learned optical flow network Θ in previous iteration $t-1$, x^t is the generated training sample, and $\mathcal{R}$ represents our training pair rendering method Realistic Image Pair Rendering (RIPR), illustrated in Sec. 3.2.

Then, in M-step, we use the newly generated dataset X^t to train and update the optical flow estimation network in a fully supervised manner:

$$\Theta^t = \arg\min_{\Theta} \mathcal{L}(X^t, \Theta), \tag{2}$$

where $\mathcal{L}$ is the learning objective of the optical flow network. Finally, an optical flow dataset and a high-precision optical flow network can be obtained by RealFlow with several EM iterations.

3.2 Realistic Image Pair Rendering

The pipeline of the proposed RIPR method is shown in Fig. 3. Given a pair of consecutive images (I_1, I_2) and an optical flow estimation network Θ, our goal is to generate an image pair with its flow label for network training. The main idea is to render a new image I_2' based on the image pair (I_1, I_2) and an estimated flow F between (I_1, I_2), so that F can be used as the training label of the new image pair (I_1, I_2'). Specifically, the reference image I_1 is first forward-warped to the target view I_2'. Then, in order to ensure the realism of the synthesized view I_2', we need to remove the occlusions and holes caused by dynamic moving objects as well as depth disparities. Fig. 4 illustrates an example.

Here, we use the **Splatting** method to identify foreground and background for these occlusion regions based on a monocular depth network [36]. Moreover, we design a **Bi-directional Hole Filling (BHF)** method to fill these hole regions using backward flow and image content from I_2. Finally, after the target view generation, the reference image, synthesized new view, and the estimated flow (I_1, I_2', F) are chosen as a training pair for dataset construction.

As detailed in Fig. 3, we first estimate the forward flow, backward flow, and the depth of I_1 and I_2 as follows:

$$F_{1\to2} = \Theta(I_1, I_2), \qquad F_{2\to1} = \Theta(I_2, I_1), \tag{3}$$

$$D_1 = \Psi(I_1), \qquad D_2 = \Psi(I_2), \tag{4}$$

where $F_{1\to2}$ and $F_{2\to1}$ are the estimated forward and backward flow, and D_1 and D_2 are the estimated depth results by the monocular depth network Ψ. Note that D_1 and D_2 are the inverse depth maps so that the pixel with larger value is closer to the camera.

In order to increase the diversity of the generated dataset, we use a factor α to add a disturbance to the estimated flow, so that the generated view is not

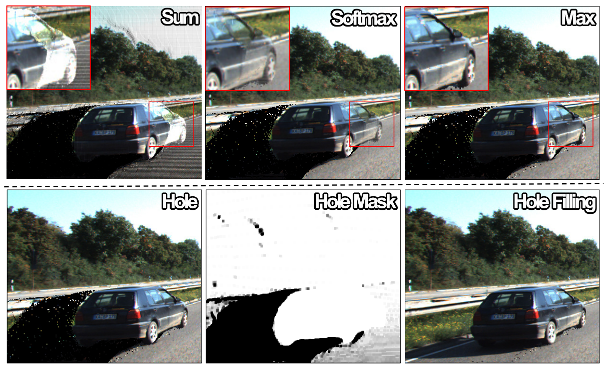

Fig. 4. Examples of the splatting results (top) and the hole filling results (bottom). For the splatting, summation is a conventional approach that produces brightness inconsistency results. Softmax leads to transparent artifacts. Max splatting renders a natural image. After hole filling by our proposed BHF, a new view image with few artifacts can be generated.

exactly the original I_2 but a new view controlled by the factor α. Thus we obtain new flow fields by the follows:

$$F'_{1\to2} = \alpha F_{1\to2}, \quad F'_{2\to1} = (1-\alpha)F_{2\to1}. \tag{5}$$

Then, we use flow fields $F'_{1\to2}$ and $F'_{2\to1}$ to render the new view by splatting method, which can be represented as:

$$I_1^s = \mathcal{S}(I_1, F'_{1\to2}, D_1), \quad I_2^s = \mathcal{S}(I_2, F'_{2\to1}, D_2), \tag{6}$$

where $\mathcal{S}$ represents the splatting method, I_1^s and I_2^s are the same view rendered from different directions. Note that the occlusion problem is addressed after the splatting operation which we will introduce later. Finally, the result view can be generated by our BHF method, which is formulated as:

$$I'_2 = \mathcal{B}(I_1^s, F'_{1\to2}, I_2^s), \tag{7}$$

where $\mathcal{B}$ represents our BHF method, I'_2 is the new image and $(I_1, I'_2, F'_{1\to2})$ is the training pair generated of RIPR.

Splatting. Splatting can be used to forward-warp the reference image I_1 into a new view I_s according to a given flow field $F'_{1\to2}$. As shown in Fig. 4 (top), the conventional sum operation for splatting often produces brightness inconsistency results. The softmax splatting method[35] is proposed to ease this problem. Assuming $\boldsymbol{q}$ is a coordinate in I_1, and $\boldsymbol{p}$ is a coordinate in the target view. The softmax splatting operation can be formulated as follows:

$$\text{let } \boldsymbol{u} = \boldsymbol{p} - (\boldsymbol{q} + F'_{1\to2}(\boldsymbol{q})), \tag{8}$$

$$b(\boldsymbol{u}) = \max(0, 1 - |\boldsymbol{u}_x|) \cdot \max(0, 1 - |\boldsymbol{u}_y|), \tag{9}$$

$$I_s(\boldsymbol{p}) = \frac{\sum_{\boldsymbol{q}} \exp D_1(\boldsymbol{q}) \cdot I_1(\boldsymbol{q}) \cdot b(\boldsymbol{u})}{\sum_{\boldsymbol{q}} \exp D_1(\boldsymbol{q}) \cdot b(\boldsymbol{u})}, \tag{10}$$

where $b(\boldsymbol{u})$ is the bilinear kernel, D_1 is the depth map of I_1 and I_s is the forward-warp result. By applying Eq. 10, background pixels that are occluded in the target view can be compressed by incorporating the depth map and the softmax operation, compared with the original sum splatting operation as illustrated in Fig. 4 (top). However, Softmax splatting in Eq. 10 may still cause unnatural results in occlusion regions. To this end, we propose to use max splatting as an alternative option of the splatting method:

$$\text{let } k = \begin{cases} 1, & \text{if } |\boldsymbol{q} + F'_{1\to 2}(\boldsymbol{q}) - \boldsymbol{p}| \leq \frac{\sqrt{2}}{2} \\ 0, & \text{otherwise,} \end{cases} \tag{11}$$

$$I_s(\boldsymbol{p}) = I_1(\boldsymbol{r}), \text{ where } \boldsymbol{r} = \arg\max_{\boldsymbol{q}} D(\boldsymbol{q}) \cdot k, \tag{12}$$

where k is the nearest kernel. Eq. 12 means that when multiple pixels are located to position $\boldsymbol{p}$, we only assign the pixel with the largest depth value to the target view. As such, the resulting image is more natural compared with the softmax version as shown in Fig. 4 (top). However, we find that the dataset generated by softmax splatting performs better than the max version in our experiments. Detailed analysis will be discussed in our experiment Sec. 6.

Bi-directional Hole Filling. Apart from occlusions, there is another problem called holes, which are produced when no pixels from original image are projected to these regions. Previous method [1] adopted an inpainting model to solve this problem, which often introduces artifacts that reduce the quality of the generated dataset. Here, we design a bi-directional hole filling method to handle these empty regions. As in Eq. 7, the input of BHF is the forward flow $F'_{1\to 2}$, and the target views I_1^s and I_2^s generated by splatting with forward and backward flows, respectively. We first check a hole mask M from $F'_{1\to 2}$ using the range map check method[48], which is formulated as follows:

$$M(\boldsymbol{p}) = \min(1, \sum_{\boldsymbol{q}} b(\boldsymbol{u})), \tag{13}$$

where $b(\boldsymbol{u})$ is the bilinear kernel described in Eq. 9. In the hole mask M, the hole pixels are labeled as 0 and others as 1. Then, we can generate a novel view image I'_2 by fusing I_1^s and I_2^s as follows:

$$I'_2 = I_1^s + (1 - M) \cdot I_2^s, \tag{14}$$

which means that the hole regions in I_1^s are filled with regions in I_2^s. By applying our BHF, realistic images can be generated, which is shown in Fig. 4 (bottom).

4 Experiments

4.1 Datasets

Flying Chairs [5] and **Flying Things** [50]**:** These two synthetic datasets are generated by randomly moving foreground objects on top of a background image. State-of-the-art supervised networks usually train on Chairs and Things.

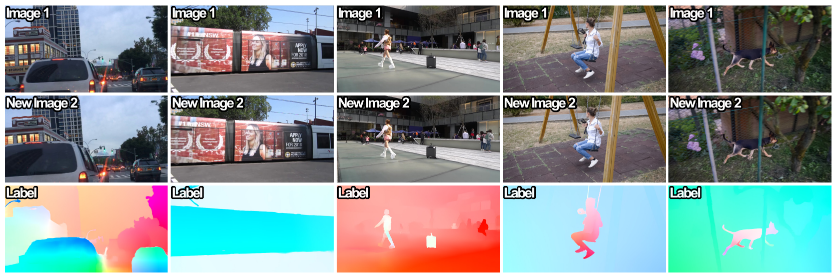

Fig. 5. Example training pairs from our generated RF-AB and RF-DAVIS. The first sample contains large motion and complex scenes.

Virtual KITTI [6]**:** Virtual KITTI is a synthetic dataset, which contains videos generated from different virtual urban environments.
DAVIS [4]**:** DVAIS dataset consists of high-quality video sequences under various kinds of scenes. No optical flow label is provided. We use 10,581 images from DAVIS challenge 2019 to generate **RF-DAVIS**.
ALOV [41] **and BDD100K** [51]**:** ALOV and BDD100K datasets are large-scale real-world video databases. We capture 75,581 image pairs from ALOV dataset and 86,128 image pairs from BDD100K dataset. There is no flow label for these image pairs, so we use RealFlow to create a large diverse real-world dataset with flow label, named **RF-AB**.
KITTI [7,34]**:** KITTI2012 and KITTI2015 are benchmarks for optical flow estimation. There are multi-view extensions (4,000 training and 3,989 testing) datasets with no ground truth. We use the multi-view extension videos (training and testing) of KITTI 2015 to generate **RF-Ktrain** and **RF-Ktest** datasets.
Sintel [3]**:** Sintel is a synthetic flow benchmark derived from 3D animated film, which contains 1,041 training pairs and 564 test pairs. We use the images from the training set to generate our **RF-Sintel**.

4.2 Implementation Details

Our RIPR consists of a depth estimation module, a flow estimation module, a splatting module, and a hole filling module. For the flow estimation module, we select RAFT [47] which represents state-of-the-art architecture for supervised optical flow. We train the RAFT using official implementation without any modifications. We initialized the RealFlow framework using RAFT pre-trained on FlyingChairs and FlyingThings unless otherwise specified. For the depth estimation module, we select DPT [36] monocular depth network, which represents the state-of-the-art architecture. For the splatting module, softmax splatting [35] is used due to the better performance. For the hole filling, our BHF uses the bi-directional flow estimated from RAFT. We will show the performance of our RIPR method affected by the different settings of above modules in Sec. 4.4.

Table 1. Comparison with previous dataset generation method[1]. We use the same source images to generate dataset and train the same network for comparison. The best results are marked in red.

Model	Dataset	KITTI12		KITTI15	
		EPE	F1	EPE	F1
RAFT	dDAVIS[1]	1.78	6.85%	3.80	13.22%
RAFT	RF-DAVIS	1.64	5.91%	3.54	9.23%
RAFT	dKITTI[1]	1.76	5.91%	4.01	13.35%
RAFT	RF-Ktest	1.32	5.41%	2.31	8.65%

Table 2. Comparison with Unsupervised Methods.The best results are marked in red and the second best are in blue, '-' indicates no results. End-point error (epe) is used as the evaluation metric.

Method	KITTI12	KITTI15	Sintel C.	Sintel F.
ARFlow [22]	1.44	2.85	2.79	3.87
SimFlow [12]	–	5.19	2.86	3.57
UFlow [17]	1.68	2.71	2.50	3.39
UpFlow [29]	1.27	2.45	2.33	2.67
SMURF [42]	–	2.00	1.71	2.58
IRRPWC(C+T) [10]	3.49	10.21	1.87	3.39
IRRPWC(C+T)+UpFlow	1.87	2.62	1.79	3.31
Ours(IRRPWC)	1.83	2.39	1.74	3.20
RAFT(C+T)[47]	2.15	5.04	1.43	2.71
Ours(RAFT)	1.20	2.16	1.34	2.38

4.3 Comparison with Existing Methods

In this section, we evaluate the effectiveness of RealFlow generation framework on the public benchmarks.

Comparison with Dataset Generation Methods. Due to the scarcity of real-world dataset generation methods for optical flow, we only select the Depthstillation method [1] for comparison. Depthstillation generated optical flow dataset dDAVIS and dKITTI from DAVIS and KITTI multi-view test. For fair comparison, we also choose the DAVIS and KITTI multi-view test videos to generate our RF-DAVIS and RF-Ktest. Fig. 5 shows our rendered training pairs. We evaluate our method on KITTI12-training and KITTI15-training sets. Quantitative results are shown in Table 1, where our method outperforms Depthstillation [1], proving the importance of realism of object motion behavior.

Comparison with Unsupervised Methods. When supervised optical flow networks are trained on synthetic datasets, they are hard to be generalized to real-world data due to the domain gap and motion discrepancy between synthetic and authentic datasets. To some extent, the effectiveness of our method depends on domain adaptation. Given the rich literature of unsupervised methods, we compare our method with them to exclude the influence of the domain. We train the RAFT [47] on RF-Ktrain and RF-Sintel by RealFlow framework. As shown in Table 2, RealFlow outperforms all the unsupervised methods on Sintel-training. We obtain a competitive result on KITTI15-training which surpass

Table 3. Comparison of our method with supervised methods on KITTI2015 train set and test set. '-' indicates no results reported.

method	KITTI15(train)		KITTI15(test)
	EPE	F1	F1
PWC-Net[45]	2.16	9.80%	9.60%
LiteFlowNet[9]	1.62	5.58%	9.38%
IRR-PWC[10]	1.63	5.32%	7.65%
RAFT[47]	0.63	1.50%	5.10%
RAFT-RVC[43]	-	-	5.56%
AutoFlow[44]	-	-	4.78%
Ours	0.58	1.35%	4.63%

all the unsupervised methods except SMURF[42]. One reason is that SMURF adopted multiple frames for training, while RealFlow only uses two frames.

Since our method is based on a model pre-trained on C+T (FlyChairs and FlyThings) in a supervised manner, we also provide the results of unsupervised methods that are pre-trained with groundtruth on C+T for a fair comparison. Because we cannot implement SMURF, we use IRRPWC [10] structure and UpFlow [29] for comparison. Specifically, we use IRRPWC pre-trained on C+T as a baseline, which is 'IRRPWC(C+T)' in Table 2. Then we train IRRPWC from the C+T pre-trained weights using unsupervised protocol provided by UpFlow on KITTI 2015 multi-view videos and Sintel sequences and do evaluation on KITTI 2012/2015 train and Sintel train data sets, which is 'IRRPWC(C+T)+UpFlow'. Finally, we perform our method using IRRPWC(C+T), which is 'Ours(IRRPWC)'. As a result, Our method can achieve better performance than unsupervised method trained from C+T pre-trained weights.

Comparison with Supervised Methods. To further prove the effectiveness of RealFlow, we use KITTI15-training to fine-tune the RAFT model pre-trained by our RF-Ktrain. Note that RF-Ktrain is generated without any sequence that contains the frames in KITTI test set. The evaluation results on KITTI15-training and KITTI15-testing are shown in Table 3. We achieve state-of-art performance on KITTI 2015 test benchmark compared to previous supervised methods.

Comparison on Large Datasets. To make our trained networks general, we collect a large-scale realistic dataset named RF-AB. We train the RAFT from scratch using our RF-AB as official implementation. Because of the scarcity of real-world evaluation benchmarks, we only evaluate our dataset on KITTI and Sintel. As summarized in Table 4, RAFT trained on RF-AB is more accurate than on other datasets when evaluated on KITTI12-training and KITTI15-training, which demonstrates the generalization ability of our method on real-world scenes. We also obtain comparable results on Sintel, which only surpass dCOCO [1]. RF-AB and dCOCO are both real-world datasets. The networks trained on them are hardly adapted to Sintel (synthetic data). So their performance is worse than C+T and Autoflow. Moreover, AutoFlow [44] learns the hyperparameters to render training data using the average end-point error (AEPE) on Sintel as the learning metric. FlyChairs and FlyingThings are also

Table 4. Comparison with large datasets. '-' indicates no results. End-point error (epe) is used as the evaluation metric.

Model	Dataset	KITTI12	KITTI15	Sintel C.	Sintel F.
RAFT	C+T[47]	2.15	5.04	1.43	2.71
RAFT	AutoFlow[44]	–	4.23	1.95	2.57
RAFT	dCOCO[1]	1.82	3.81	2.63	3.90
RAFT	RF-AB	1.80	3.48	1.80	3.28

rendered to match the displacement distribution of Sintel. Mayer *et al.* [30] shows that matching the displacement statistics of the test data is important.

Impact on Different Optical Flow Networks. In Table 5, we also provide experiment results to prove that our method can improve other supervised networks on real-world scenes not only on specific architecture such as RAFT. For fair comparison, we trained IRR-PWC [10] and GMA [16] on RF-AB and RF-Sintel with the official settings. Table 5 shows that RAFT and GMA trained on RF-AB outperform the original variants trained on C+T when testing on real-world data KITTI. Moreover, there is a significant improvement on IRR-PWC which is effective as trained RAFT on C+T. This fact proves that a better dataset is crucial to a supervised network.

4.4 Ablation Study

In this section, we conduct a series of ablation studies to analyze the impact of different module choices of the RIPR method. We measure all the factors using RF-Ktrain to train RAFT and evaluate on KITTI12-training and KITTI15-training. Because there are multiple combinations of these factors, we only test a specific component of our approach in isolation. As shown in Table 6, default settings are underlined and detail experiment settings will be discussed below.

Render. We conduct an experiment named 'Render Off' where we use original image pairs and their estimated flows to train the network. When applying our RIPR method, as the 'Render On' in Table 6, the accuracy of the network can be improved significantly. Moreover, our rendering method is also related to the video interpolation methods [49,8]. We replace our rendering method with QVI [49] for fair comparison. Note that 'QVI(RAFT)' uses RAFT pre-trained on C+T for optical flow estimation, which is the same model as the initial model of RealFlow. As a result, our method outperforms QVI for optical flow dataset generation because the frame synthesis process in QVI may cause the content of the generated frame to not match the optical flow label.

Depth. To measure the effectiveness of the depth estimation in the splatting method, we conduct three different experiments: DPT[36], Midas[37], and 'Occ-bi'. The 'Occ-bi' means that the depth map is replaced by the occlusion map produced by the bi-directional flow check method[33]. DPT is a state-of-art method that outperforms Midas in the task of monocular depth estimation. From Table 6, we can notice that with more accurate depth estimation results, our RealFlow can generate better dataset for optical flow learning, which proves that depth is a crucial cue in our framework.

Table 5. Impact on different optical flow networks. The value in the bracket means the percentage of improvement.End-point error (epe) is used as the evaluation metric.

Model	Dataset	KITTI12	KITTI15	Sintel C.	Sintel F.
IRR-PWC[10]	C+T	3.49	10.21	1.87	3.39
IRR-PWC	RF-AB	2.13	5.09	3.68	4.72
IRR-PWC	RF-Sintel	2.67	7.06	1.74	3.20
GMA[16]	C+T	1.99	4.69	1.30	2.73
GMA	RF-AB	1.82	3.64	1.93	3.45
GMA	RF-Sintel	1.74	4.39	1.23	2.32
RAFT[47]	C+T	2.15	5.04	1.43	2.71
RAFT	RF-AB	1.80	3.48	1.80	3.28
RAFT	RF-Sintel	1.76	4.36	1.34	2.38

Table 6. Ablation experiments. Settings used in our final framework are underlined. Here we only perform one EM iteration for these experiments due to the limitation of computational resources.

Experiment	Method	KITTI12		KITTI15	
		EPE	F1	EPE	F1
Render	Off	1.80	7.78%	3.93	14.38%
	QVI [49]	2.84	10.7%	7.27	20.36%
	QVI(RAFT)	4.03	14.0%	9.03	24.70%
	<u>On</u>	1.44	5.90%	2.79	10.66%
Depth	Occ-bi	1.51	6.22%	3.01	11.12%
	MiDas[37]	1.49	6.25%	2.90	11.18%
	<u>DPT</u>[36]	1.44	5.90%	2.79	10.66%
Splatting	Max	1.62	5.90%	3.03	11.04%
	<u>Softmax</u>[35]	1.44	5.90%	2.79	10.66%
Hole Filling	w/o Filling	1.45	6.06%	2.95	10.80%
	RFR[20]	1.53	6.07%	2.95	11.23%
	<u>BHF</u>	1.44	5.90%	2.79	10.66%
Range of α	[1]	1.57	6.34%	3.38	12.30%
	[-2,2]	1.45	5.92%	2.83	10.90%
	<u>[0,2]</u>	1.44	5.90%	2.79	10.66%

Splatting. We compared two versions of splatting: Max and Softmax[35]. Max splatting leads to the right rendering result of visual appearance. However, we find that Softmax splatting outperforms Max splatting as in Table 6. The reason is that max splatting may cause tearing of texture when the depth is incorrect, while softmax splatting can alleviate this problem by generating a translucent fusion result. Please refer to supplementary materials for more details.

Hole Filling. The optical flow network learns a per-pixel matching of two images. The hole in the newly generated image means that there is no pixel matched to the reference image. Although it happens, the context information can also help the network. For fair comparison, we use the RFR[20] fine-tuned on KITTI dataset for inpainting these holes. As summarized in Table 6, our designed BHF method achieves the best results. We also conduct an experiment without hole filling which leads to a moderate improvement over RFR. It suggests that a worse hole filling result may introduce negative effects.

Table 7. Iteration times. The best iteration time is underlined. Our RealFlow can converge to similar results with different initialization settings.

Model	Initialize Dataset	Iteration Times	KITTI12 EPE	KITTI12 F1	KITTI15 EPE	KITTI15 F1
RAFT	C+T	init	2.15	9.29%	5.04	17.4%
		Iter.1	1.44	5.90%	2.79	10.7%
		Iter.1∗4	1.45	5.59%	2.86	10.4%
		Iter.2	1.31	5.28%	2.36	8.46%
		Iter.3	1.28	5.02%	2.20	8.27%
		<u>Iter.4</u>	1.27	5.17%	2.16	8.45%
RAFT	VKITTI	init	1.81	5.04%	3.13	8.63%
		<u>Iter.1</u>	1.26	4.47%	2.11	7.50%
GMA	C+T	init	1.99	9.28%	4.69	17.1%
		<u>Iter.1</u>	1.46	5.56%	2.79	10.2%

Range of α. To increase the diversity of our generated dataset, we add a disturbance to our RealFlow by α, which is introduced in Sec. 3.1. We use three different settings in Table 6 'Range of α'. '[1]' means that α is always set as 1. The other two settings mean that we randomly sample a value within that range. As can be seen, factor α sampled from range $[0, 2]$ achieves better result.
EM Iteration Times. In RealFlow framework, the generated dataset and the optical flow network are gradually improved after iterations. However, a certain upper limit exists in RealFlow and it will converge after several iterations. As summarized in Table 7, after 4 iterations, RealFlow converges and the result cannot be further improved. 'Iter.1∗4' means that the network is trained 4 times longer (more training steps) with the data of 'Iter.1'. As can be seen, simply training 4 times longer cannot bring improvement compared with 4 EM iterations of RealFlow (see 'Iter.4'), which demonstrates the effectiveness of our approach.
Initial Model. It is well-known that the initialization is important for EM algorithm. In Table 7, we implement RAFT pre-trained on Virtual KITTI (VKITTI) and GMA pre-trained on C+T as the initial model of RealFlow. As can be seen, the performance can be improved after learning with our RealFlow.

5 Conclusions

In this work, we have presented RealFlow, an EM-based framework for optical flow dataset generation on realistic videos. We have proposed a Realistic Image Pair Rendering (RIPR) method to render a new view from a pair of images, according to the estimated optical flow. We have trained optical flow networks on the synthesized dataset. Experiment results show that the trained networks and the generated datasets can be improved iteratively, yielding a large-scale high-quality flow dataset as well as a high-precision optical flow network. Experiments show that the performance of existing methods can be largely improved on widely-used benchmarks while using our RealFlow dataset for training.
Acknowledgement: This work was supported by the National Natural Science Foundation of China (NSFC) No.62173203, No.61872067 and No.61720106004.

References

1. Aleotti, F., Poggi, M., Mattoccia, S.: Learning optical flow from still images. In: Proc. CVPR. pp. 15201–15211 (2021) 2, 5, 8, 10, 11, 12
2. Baker, S., Scharstein, D., Lewis, J., Roth, S., Black, M.J., Szeliski, R.: A database and evaluation methodology for optical flow. International Journal of Computer Vision **92**(1), 1–31 (2011) 2, 4
3. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: Proc. ECCV. pp. 611–625 (2012) 1, 2, 9
4. Caelles, S., Pont-Tuset, J., Perazzi, F., Montes, A., Maninis, K.K., Van Gool, L.: The 2019 davis challenge on vos: Unsupervised multi-object segmentation. arXiv:1905.00737 (2019) 9
5. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Smagt, P.v.d., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proc. ICCV. pp. 2758–2766 (2015) 3, 4, 8
6. Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: Virtual worlds as proxy for multi-object tracking analysis. In: Proc. CVPR. pp. 4340–4349 (2016) 5, 9
7. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Proc. CVPR. pp. 3354–3361 (2012) 4, 9
8. Huang, Z., Zhang, T., Heng, W., Shi, B., Zhou, S.: Real-time intermediate flow estimation for video frame interpolation. In: Proc. ECCV (2022) 12
9. Hui, T.W., Tang, X., Loy, C.C.: Liteflownet: A lightweight convolutional neural network for optical flow estimation. In: Proc. CVPR. pp. 8981–8989 (2018) 4, 11
10. Hur, J., Roth, S.: Iterative residual refinement for joint optical flow and occlusion estimation. In: Proc. CVPR. pp. 5754–5763 (2019) 4, 10, 11, 12, 13
11. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: Proc. CVPR. pp. 2462–2470 (2017) 4
12. Im, W., Kim, T.K., Yoon, S.E.: Unsupervised learning of optical flow with deep feature similarity. In: Proc. ECCV. pp. 172–188 (2020) 4, 10
13. Janai, J., Guney, F., Ranjan, A., Black, M., Geiger, A.: Unsupervised learning of multi-frame optical flow with occlusions. In: Proc. ECCV. pp. 690–706 (2018) 4
14. Janai, J., Guney, F., Wulff, J., Black, M.J., Geiger, A.: Slow flow: Exploiting high-speed cameras for accurate and diverse optical flow reference data. In: Proc. CVPR. pp. 3597–3607 (2017) 5
15. Jason, J.Y., Harley, A.W., Derpanis, K.G.: Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In: Proc. ECCV. pp. 3–10 (2016) 4
16. Jiang, S., Campbell, D., Lu, Y., Li, H., Hartley, R.: Learning to estimate hidden motions with global motion aggregation. In: Proc. ICCV. pp. 9772–9781 (2021) 4, 12, 13
17. Jonschkowski, R., Stone, A., Barron, J.T., Gordon, A., Konolige, K., Angelova, A.: What matters in unsupervised optical flow. In: Proc. ECCV. pp. 557–572 (2020) 10
18. Lai, W.S., Huang, J.B., Yang, M.H.: Semi-supervised learning for optical flow with generative adversarial networks. In: Proc. NeurIPS. pp. 353–363 (2017) 2
19. Li, H., Luo, K., Liu, S.: Gyroflow: Gyroscope-guided unsupervised optical flow learning. In: Proc. ICCV. pp. 12869–12878 (2021) 4
20. Li, J., Wang, N., Zhang, L., Du, B., Tao, D.: Recurrent feature reasoning for image inpainting. In: Proc. CVPR. pp. 7760–7768 (2020) 13

21. Liu, C., Freeman, W.T., Adelson, E.H., Weiss, Y.: Human-assisted motion annotation. In: Proc. CVPR. pp. 1–8 (2008) 2, 4
22. Liu, L., Zhang, J., He, R., Liu, Y., Wang, Y., Tai, Y., Luo, D., Wang, C., Li, J., Huang, F.: Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In: Proc. CVPR. pp. 6489–6498 (2020) 10
23. Liu, P., King, I., Lyu, M., Xu, J.: Ddflow:learning optical flow with unlabeled data distillation. In: Proc. AAAI. pp. 8770–8777 (2019) 4
24. Liu, P., Lyu, M., King, I., Xu, J.: Selflow:self-supervised learning of optical flow. In: Proc. CVPR. pp. 4571–4580 (2019) 4
25. Liu, S., Luo, K., Luo, A., Wang, C., Meng, F., Zeng, B.: Asflow: Unsupervised optical flow learning with adaptive pyramid sampling. IEEE Transactions on Circuits and Systems for Video Technology **32**(7), 4282–4295 (2021) 4
26. Liu, S., Luo, K., Ye, N., Wang, C., Wang, J., Zeng, B.: Oiflow: Occlusion-inpainting optical flow estimation by unsupervised learning. IEEE Trans. on Image Processing **30**, 6420–6433 (2021) 4
27. Luo, A., Yang, F., Li, X., Liu, S.: Learning optical flow with kernel patch attention. In: Proc. CVPR. pp. 8906–8915 (2022) 4
28. Luo, A., Yang, F., Luo, K., Li, X., Fan, H., Liu, S.: Learning optical flow with adaptive graph reasoning. In: Proc. AAAI. pp. 1890–1898 (2022) 4
29. Luo, K., Wang, C., Liu, S., Fan, H., Wang, J., Sun, J.: Upflow: Upsampling pyramid for unsupervised optical flow learning. In: Proc. CVPR. pp. 1045–1054 (2021) 10, 11
30. Mayer, N., Ilg, E., Fischer, P., Hazirbas, C., Cremers, D., Dosovitskiy, A., Brox, T.: What makes good synthetic training data for learning disparity and optical flow estimation? International Journal of Computer Vision **126**(9), 942–960 (2018) 12
31. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proc. CVPR. pp. 4040–4048 (2016) 2, 5
32. McLachlan, G.J., Krishnan, T.: The EM algorithm and extensions, vol. 382. John Wiley & Sons (2007) 5
33. Meister, S., Hur, J., Roth, S.: Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In: Proc. AAAI (2018) 4, 12
34. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: Proc. CVPR. pp. 3061–3070 (2015) 1, 4, 9
35. Niklaus, S., Liu, F.: Softmax splatting for video frame interpolation. In: Proc. CVPR. pp. 5437–5446 (2020) 7, 9, 13
36. Ranftl, R., Bochkovskiy, A., Koltun, V.: Vision transformers for dense prediction. In: Proc. ICCV. pp. 12179–12188 (2021) 5, 6, 9, 12, 13
37. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE Trans. on Pattern Analysis and Machine Intelligence **44**(3), 1623–1637 (2022) 12, 13
38. Ranjan, A., Black, M.J.: Optical flow estimation using a spatial pyramid network. In: Proc. CVPR. pp. 4161–4170 (2017) 4
39. Ren, Z., Luo, W., Yan, J., Liao, W., Yang, X., Yuille, A., Zha, H.: Stflow: Self-taught optical flow estimation using pseudo labels. IEEE Trans. on Image Processing **29**, 9113–9124 (2020) 4
40. Ren, Z., Yan, J., Ni, B., Liu, B., Yang, X., Zha, H.: Unsupervised deep learning for optical flow estimation. In: Proc. AAAI. pp. 1495–1501 (2017) 4

41. Smeulders, A.W., Chu, D.M., Cucchiara, R., Calderara, S., Dehghan, A., Shah, M.: Visual tracking: An experimental survey. IEEE Trans. on Pattern Analysis and Machine Intelligence **36**(7), 1442–1468 (2013) 9
42. Stone, A., Maurer, D., Ayvaci, A., Angelova, A., Jonschkowski, R.: Smurf: Self-teaching multi-frame unsupervised raft with full-image warping. In: Proc. CVPR. pp. 3887–3896 (2021) 10, 11
43. Sun, D., Herrmann, C., Jampani, V., Krainin, M., Cole, F., Stone, A., Jonschkowski, R., Zabih, R., Freeman, W.T., Liu, C.: Tf-raft: A tensorflow implementation of raft. In: ECCV Robust Vision Challenge Workshop (2020) 11
44. Sun, D., Vlasic, D., Herrmann, C., Jampani, V., Krainin, M., Chang, H., Zabih, R., Freeman, W.T., Liu, C.: Autoflow: Learning a better training set for optical flow. In: Proc. CVPR. pp. 10093–10102 (2021) 2, 3, 5, 11, 12
45. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: Proc. CVPR. pp. 8934–8943 (2018) 11
46. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Models matter, so does training: An empirical study of cnns for optical flow estimation. IEEE Trans. on Pattern Analysis and Machine Intelligence **42**(6), 1408–1423 (2020) 1
47. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: Proc. ECCV. pp. 402–419 (2020) 1, 4, 9, 10, 11, 12, 13
48. Wang, Y., Yang, Y., Yang, Z., Zhao, L., Wang, P., Xu, W.: Occlusion aware unsupervised learning of optical flow. In: Proc. CVPR. pp. 4884–4893 (2018) 4, 8
49. Xu, X., Siyao, L., Sun et al., W.: Quadratic video interpolation. Proc. NeurIPS **32** (2019) 12, 13
50. Yang, G., Song, X., Huang, C., Deng, Z., Shi, J., Zhou, B.: Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In: Proc. CVPR. pp. 899–908 (2019) 8
51. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: Proc. CVPR. pp. 2636–2645 (2020) 9
52. Zhong, Y., Ji, P., Wang, J., Dai, Y., Li, H.: Unsupervised deep epipolar flow for stationary or dynamic scenes. In: Proc. CVPR. pp. 12095–12104 (2019) 4