MPI-Flow: Learning Realistic Optical Flow with Multiplane Images

Yingping Liang[1] Jiaming Liu[2] Debing Zhang[2] Ying Fu[1*]
[1]Beijing Institute of Technology [2]Xiaohongshu Inc.

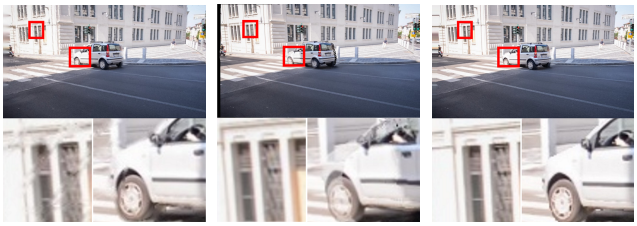
Abstract

The accuracy of learning-based optical flow estimation models heavily relies on the realism of the training datasets. Current approaches for generating such datasets either employ synthetic data or generate images with limited realism. However, the domain gap of these data with real-world scenes constrains the generalization of the trained model to real-world applications. To address this issue, we investigate generating realistic optical flow datasets from real-world images. Firstly, to generate highly realistic new images, we construct a layered depth representation, known as multiplane images (MPI), from single-view images. This allows us to generate novel view images that are highly realistic. To generate optical flow maps that correspond accurately to the new image, we calculate the optical flows of each plane using the camera matrix and plane depths. We then project these layered optical flows into the output optical flow map with volume rendering. Secondly, to ensure the realism of motion, we present an independent object motion module that can separate the camera and dynamic object motion in MPI. This module addresses the deficiency in MPI-based single-view methods, where optical flow is generated only by camera motion and does not account for any object movement. We additionally devise a depth-aware inpainting module to merge new images with dynamic objects and address unnatural motion occlusions. We show the superior performance of our method through extensive experiments on real-world datasets. Moreover, our approach achieves state-of-the-art performance in both unsupervised and supervised training of learning-based models. The code will be made publicly available at: https://github.com/Sharpiless/MPI-Flow.

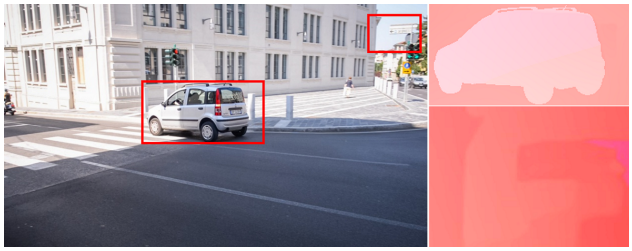
1. Introduction

Optical flow refers to the precise calculation of per-pixel motion between consecutive video frames. Its applications span a wide range of fields, including object tracking [10, 51], robot navigation [23, 41], three-dimensional (3D) reconstruction [24, 14], and visual simultaneous localization and mapping (SLAM) [52, 8, 29]. In recent years, with the rapid development of neural networks, learning-based methods [44, 45] have demonstrated significant advances compared to traditional model-based algorithms [3, 49, 50]. Conventional practices primarily rely on synthetic data, as demonstrated by [9, 18, 4]. Synthetic data contains exact optical flow labels and animated images. However, the domain gap between synthetic and real data hinders its further improvements in real-world applications.

Recent studies have aimed to extract optical flow from real-world data by employing hand-made special hardware [2, 11, 34]. However, the rigidly controlled and inefficient collection procedure limits their applicability. To address this issue, Depthstillation [1], and RealFlow [12] have been proposed, which project each pixel in the real-world image onto the novel view frame with the help of random motions



(a) Depthstill. [1] (b) RealFlow [12] (c) Ours



(d) Source image and details of generated flows from our method

Figure 1: Visual comparison of images reveals that our method, when compared with those based on real-world images [1, 12], demonstrates superior image realism. Our method also achieves fine optical flows by employing volume rendering in MPI and separate motions of the camera and object with the independent object motion module.

*Corresponding Author: fuying@bit.edu.cn

of virtual cameras or estimated flows. Nonetheless, both methods are limited by the lack of image realism, leading to issues such as collisions, holes, and artifacts, as illustrated in Figure 1. These limitations constrain the real-world performance of learning-based optical flow models [44, 45].

To achieve higher image realism, we turn our attention to the use of single-view Multiplane Images (MPI) [53, 47, 54, 6, 13]. This line of work demonstrates remarkable single-view image rendering capabilities and effectively reduces collisions, holes, and artifacts commonly found in previous methods [1, 12]. These advancements contribute to higher image realism, prompting a natural question: Can high-realistic MPI methods be adapted to generate high-quality optical flow datasets for training purposes?

To this end, we propose *MPI-Flow*, aiming to generate realistic optical flow datasets from real-world images. Specifically, we first review the image synthesis pipeline of MPI and devise an optical flow generation pipeline along with image synthesis. In this step, we build an MPI by warping single-view image features onto each layered plane with the predicted color and density. The color and density then be mapped into a realistic new image via volume rendering. With the layered planes, we extract optical flows with virtual camera motions from the rendered image and the real image. Second, as the MPI can only be applied in static scenes, which yield limited motion realism, we propose an independent object motion module and a depth-aware inpainting module to tackle this issue. The independent object motion module decouples dynamic objects from static scenes and applies different virtual camera matrices to calculate the motion of both dynamic and static parts. The depth-aware inpainting module is introduced to remove the object occlusion in the synthesized new image.

With MPI-Flow, a large number of single-view images can be used to generate large-scale training datasets with realistic images and motions. This enables learning-based optical flow models better generalization to a wide range of real-world scenes. Extensive experiments on real datasets demonstrate the effectiveness of our approach. In summary, our main contributions are as follows:

- We are the first to present a novel MPI-based optical flow dataset generation framework, namely MPI-Flow, which can significantly improve the realism of the generated images and motion.
- We present a novel independent object motion module for modeling dynamic objects in MPI, which can model realistic optical flow from camera motion and object motion simultaneously.
- We design a depth-aware inpainting module for realistic image inpainting, which can remove unnatural motion occlusions in generated images.

2. Related Work

In this section, we review the most relevant studies on optical flow networks, optical flow dataset generation, and novel view image synthesis methods.

Supervised Optical Flow Network. Early methods train deep neural networks to match patches across images [49]. FlowNet [9] first trains convolutional neural networks on the synthetic datasets with optical flow. Moreover, the follow-up methods [16, 17, 18, 32, 31] with advanced modules and network architectures make a significant improvement in supervised optical flow learning, with RAFT [45] representing state-of-the-art. However, generalization remains a cause for concern due to the domain gap between synthetic datasets and real-world applications. To address this problem, our work focuses on generating realistic optical flow datasets from real-world images.

Dataset Generation for Optical Flow. The use of fluorescent texture to record motions in real-world scenes is first described in [2] to obtain flow maps. KITTI [11, 34] provides sophisticated training data through complex lidar and camera setups. However, the aforementioned real-world datasets have limited quantities and constrained scenes, making it difficult for models trained using deep supervised learning to generalize to more expansive scenes. Synthesized training pairs, such as those in Flyingchairs [9] and Flyingthings [18], have shown promise for supervised learning. However, moving animated image patches cannot accurately match real-world scenes, leading to domain gaps. AutoFlow [43] introduces a learning-based approach for generating training data by hyper-parameters searching. However, AutoFlow relies on optical flow labels for domain adaptation, which is not practical in most scenarios where ground truth labels are unavailable.

Two recent works have proposed methods for generating training datasets based on real-world images or videos. The first, called Depthstillation [1], synthesizes paired images by estimating depth and optical flows from a single still image. Optical flows are calculated based on the virtual camera pose and depth. The second method, called RealFlow [12], synthesizes intermediate frames between two frames using estimated optical flows with RAFT. However, both methods use naive image synthesis techniques that fail to meet the demand for realism criteria due to hole-filling and artifacts. In contrast, our method improves on this approach by using a well-designed and modified multiplane image (MPI) technique to obtain realistic images.

Novel View Synthesis. View synthesis methods aim to generate new images from arbitrary viewpoints by utilizing a given scene. Several classical approaches [48, 30, 53, 35] have been proposed that utilize multiple views of a scene to render novel views with geometric consistency. However, synthesizing novel views from a single image remains challenging due to the limited scene information available.

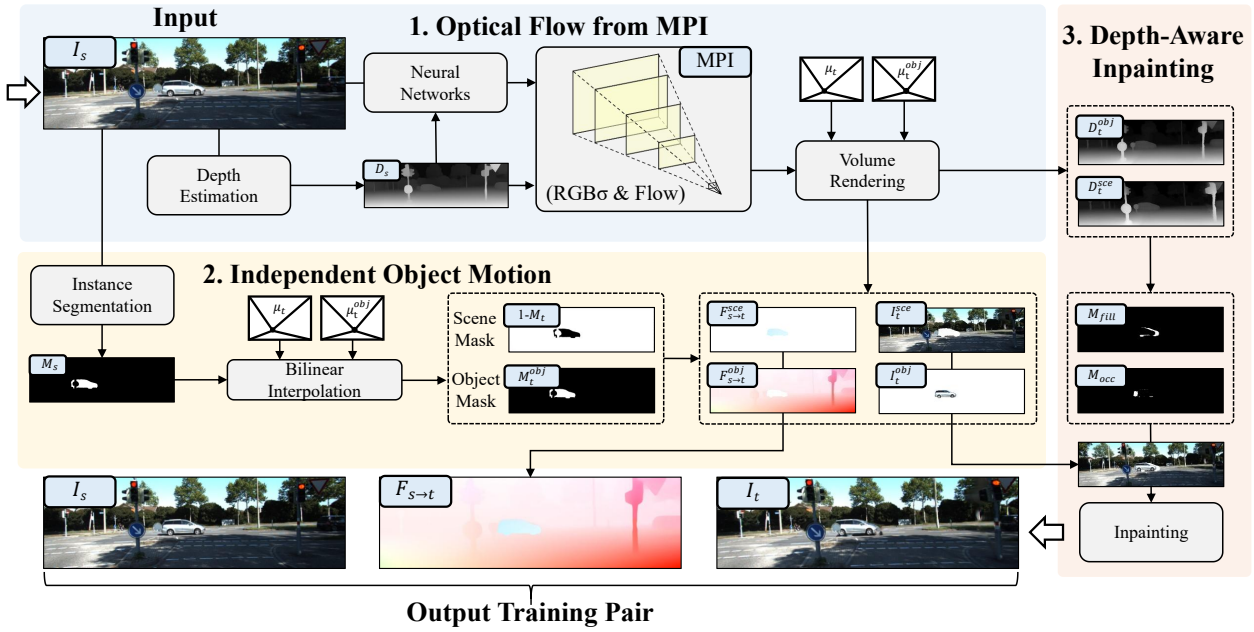


Figure 2: Illustration of our proposed MPI-Flow framework with single-view image $\mathbf{I}_s$ as input. We estimate depths to construct MPI where RGB and density of each plane are predicted by neural networks and the flow of each plane is calculated with camera matrixes. Both the novel views and flow maps are rendered by volume rendering and separated by the independent object motion module with novel view object masks. The new image is merged by depth-aware inpainting.

Pixelsynth [39] and Geometry-Free View Synthesis [40] address this challenge by optimizing the synthesizing model using multi-view supervision. However, their generalization to in-the-wild scenes is hindered by the lack of large-scale multi-view datasets. Single-view MPI [47] and MINE [26] decompose the scene into multiple layers and utilize an inpainting network to extend each occluded layer. Additionally, AdaMPI [13] addresses complex 3D scene structures through a novel plane adjustment network. These MPI-based methods have demonstrated success in synthesizing realistic images, and thus, we adopt multiplane images as our basic synthesis tool. However, to the best of our knowledge, there are currently no publicly available methods for generating optical flow datasets from MPI. To extract optical flows from MPI, we propose a novel pipeline that differs from previous MPI-based image synthesis methods by utilizing layered depths and virtual camera poses. Additionally, to enhance the realism of the generated optical flow dataset, we introduce an independent object motion module for static and dynamic decoupling, as well as a depth-aware inpainting module to remove unnatural occupations.

3. The Proposed MPI-Flow

In this section, we first briefly review the basics of our motivation and formulation for novel view image generation. Then we introduce the optical flow generation pipeline. Next, we present the details of two crucial components of our approach, including independent object motions and depth-aware inpainting.

3.1. Motivation and Formulation

Our goal is to generate a realistic novel view image $\mathbf{I}_t \in \mathbb{R}^{H\times W\times 3}$ and the corresponding optical flow maps $\mathbf{F}_{s\to t} \in \mathbb{R}^{H\times W\times 2}$ from single-view image $\mathbf{I}_s \in \mathbb{R}^{H\times W\times 3}$. H and W are the height and width of the image, respectively. The two-dimensional array on the optical flow $\mathbf{F}_{s\to t}$ represents the change of the corresponding pixel from image $\mathbf{I}_s$ to image $\mathbf{I}_t$. The input image, generated image, and optical flow together form a training pair.

To generate training pair, previous works [1, 12] wrap pixels from image $\mathbf{I}_s$ to image $\mathbf{I}_t$ with estimated flows. This inevitably leads to holes and artifacts in the image $\mathbf{I}_t$, which damages the image realism. Recent work [47, 26, 13] on Multiplane Images (MPI) reveals that the layered depth representation of the single-view image can significantly improve the realism of the generated novel view image.

We aim to tackle the image realism challenges in our methods and meanwhile enhance the optical flow realism and motion realism. Accordingly, we present an MPI-based optical flow dataset generation method, namely MPI-Flow. Figure 2 shows the MPI-Flow framework for training pair generation. To construct MPI, given the input image $\mathbf{I}_s$,

an off-the-shelf monocular depth estimation network [38] is used to estimate its depth map. Then we use a neural network to construct N fronto-parallel RGBσ planes with color, density, and depth predicted by neural networks in the rays under novel viewpoints.

To decouple dynamic objects, an instance segmentation network [7] gives the object mask. Then we use bilinear interpolation to obtain the object masks under two viewpoints respectively. Using object masks and constructed MPI, we use volume rendering to render the separate novel view images, optical flow maps, and depths of dynamic objects and the static scene, respectively. The optical flow $\mathbf{F}_{s\to t}$ can be obtained simply by adding the optical flows of the objects and the scene. However, due to different viewpoints, merging new images results in vacant areas and false occlusion. To this end, we design a depth-aware inpainting module, using rendered depths and object masks to fill the holes and repair false occlusion in the synthesized new image $\mathbf{I}_t$.

3.2. Optical Flow Data Generation

Novel View Image from MPI. To render realistic image $\mathbf{I}_t$ under a target viewpoint $\boldsymbol{\mu}_t$, we use pixel warping from the source-view MPI in a differentiable manner. Specifically, we use a neural network $\mathcal{F}$ as in [13] to construct N fronto-parallel RGBσ planes under source viewpoint $\boldsymbol{\mu}_s$ with color channels $\mathbf{c}_n$, density channel $\boldsymbol{\sigma}_n$, and depth $\mathbf{d}_n$ from the input image $\mathbf{I}_s$ and its depth map $\mathbf{D}_s$ as:

$$\{(\mathbf{c}_n, \boldsymbol{\sigma}_n, \mathbf{d}_n)\}_{n=1}^{N} = \mathcal{F}(\mathbf{I}_s, \mathbf{D}_s), \tag{1}$$

where N is a predefined parameter that represents the number of planes in MPI. Each pixel (x_t, y_t) on the novel view image plane can be mapped to pixel (x_s, y_s) on n-th source MPI plane via homography function [15]:

$$[x_s, y_s, 1]^T \sim \mathbf{K}\left(\mathbf{R} - \frac{\mathbf{t}\mathbf{n}^T}{\mathbf{d}_n}\right)\mathbf{K}^{-1}[x_t, y_t, 1]^T, \tag{2}$$

where $\mathbf{R}$ and $\mathbf{t}$ are the rotation and translation from the source viewpoints $\boldsymbol{\mu}_s$ to the target viewpoints $\boldsymbol{\mu}_t$, $\mathbf{K}$ is the camera intrinsic, and $\mathbf{n} = [0, 0, 1]$ is the normal vector. Thus, the color $\mathbf{c}'_n$ and density $\boldsymbol{\sigma}'_n$ of each new plane for the novel view $\mathbf{I}_t$ can be obtained via bilinear sampling. We use discrete intersection points between new planes and arbitrary rays passing through the scene and estimate integrals:

$$\mathbf{I}_t = \sum_{n=1}^{N}\left(\mathbf{c}'_n\boldsymbol{\alpha}'_n\prod_{m=1}^{n-1}(1-\boldsymbol{\alpha}'_m)\right), \tag{3}$$

where $\boldsymbol{\alpha}'_n = \exp(-\boldsymbol{\delta}_n\boldsymbol{\sigma}'_n)$ and $\boldsymbol{\delta}_n$ is the distance map between plane n and $n+1$ and we set the initial depth of MPI planes uniformly spaced in disparity as in [13].

Optical Flow from MPI. Although MPI-based methods synthesize realistic images, reliable optical flow maps are also needed to train learning-based optical flow estimation models. Therefore, we propose adding an additional optical channel in each plane. To this end, we compute the optical flow on the n-th plane at pixel $[x_s, y_s]$ of source image $\mathbf{I}_s$ by $\mathbf{f}_n = [x_t - x_s, y_t - y_s]$ with a backward-warp process in terms of the inverse equivalent form of Equation (2):

$$[x_t, y_t, 1]^T \sim \mathbf{K}\left(\mathbf{R}^\dagger - \frac{\mathbf{t}^\dagger\mathbf{n}^T}{\mathbf{d}_n}\right)\mathbf{K}^{-1}[x_s, y_s, 1]^T, \tag{4}$$

where x_s and y_s are uniformly sampled from a $H \times W$ grid. $\mathbf{R}^\dagger$ and $\mathbf{t}^\dagger$ are the inverses of $\mathbf{R}$ and $\mathbf{t}$, respectively.

To make sure that the optical flow maps match the novel view image $\mathbf{I}_t$ perfectly, we propose to render $\mathbf{F}_{s\to t}$ as in Equation (3) in terms of volume rendering:

$$\mathbf{F}_{s\to t} = \sum_{n=1}^{N}\left(\mathbf{f}_n\boldsymbol{\alpha}_n\prod_{m=1}^{n-1}(1-\boldsymbol{\alpha}_m)\right), \tag{5}$$

where $\mathbf{f}_n \in \mathbb{R}^{H\times W\times 2}$ is the optical flow maps on the n-th plane of image $\mathbf{I}_s$. The pipeline implemented thus far models the optical flows resulting from camera motion without considering the potential presence of independently dynamic objects. However, real-world scenes are highly likely to contain such objects. Not incorporating their motions can lead to domain gaps by unrealistic optical flows.

Independent Object Motions. To model more realistic motions, we propose applying separate virtual motions to objects and static backgrounds extracted from the scene. Therefore, we utilize an instance segmentation network Ω [7] for extracting the main object in the source image $\mathbf{I}_s$ as:

$$\mathbf{M}_s = \Omega(\mathbf{I}_s) \in \mathbb{R}^{H\times W}, \tag{6}$$

where $\mathbf{M}_s$ is a binary mask to indicate the region of the object. To model the motions of the object $\mathbf{M}_s$ in the scene, we construct separate viewpoints, including camera motion $\boldsymbol{\mu}_t^{sce}$ and object motion $\boldsymbol{\mu}_t^{obj}$. We then obtain the rendered scene novel view $\mathbf{I}_t^{sce}$ and object novel view $\mathbf{I}_t^{obj}$ as in Equation (3). The separate optical flows, $\mathbf{F}_{s\to t}^{sce}$ and $\mathbf{F}_{s\to t}^{obj}$ can also be obtained as in Equation (5). The optical flows in $\mathbf{F}_{s\to t}$ are mixed by the values in $\mathbf{F}_{s\to t}^{sce}$ and $\mathbf{F}_{s\to t}^{obj}$ in terms of mask $\mathbf{M}_s$ to get the final optical flow maps containing camera motion and dynamic objects for training.

We can then use the bilinear interpolation to get the new object masks $\mathbf{M}_t$ and $\mathbf{M}_t^{obj}$ under the new viewpoints $\boldsymbol{\mu}_t$ and $\boldsymbol{\mu}_t^{obj}$ via bilinear sampling from $\mathbf{M}_s$. Pixels in the new merged image are also selected according to the content masks $1 - \mathbf{M}_t$ and $\mathbf{M}_t^{obj}$ from $\mathbf{I}_t^{sce}$ and $\mathbf{I}_t^{obj}$. Then, a simple inpainting strategy [46] is used to fill the empty area in the new image with an inpainting mask calculated by $\mathbf{M}_{fill} = \mathbf{M}_t \odot (1 - \mathbf{M}_t^{obj})$.

Depth-Aware Inpainting Although merged images give a realistic visual effect, depth changes caused by camera motion and object motion can also cause unnatural occlusions. To solve this problem, we use volume rendering to obtain the depth $\mathbf{D}_t^{sce}$ of the scene novel view:

$$\mathbf{D}_t = \sum_{n=1}^{N} \left(\mathbf{d}'_n \boldsymbol{\alpha}'_n \prod_{m=1}^{n-1} (1 - \boldsymbol{\alpha}'_m) \right), \tag{7}$$

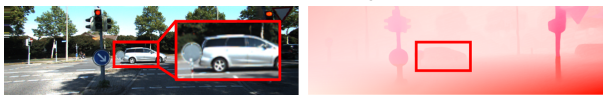
and the depth of the object novel view $\mathbf{D}_t^{obj}$ can be obtained in the same way. We then utilize both depths to compute the occupation mask between the novel views:

$$\mathbf{M}_{occ} = (1 - \mathbf{M}_t) \odot \mathbf{M}_t^{obj} \odot (\mathbf{D}_t < \mathbf{D}_t^{obj}), \tag{8}$$

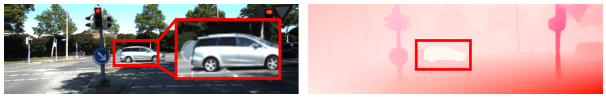
which indicates the background areas in front of the object. Therefore, we are able to restore the coincidence area between the object and the background in the new image $\mathbf{I}_t$.

Figure 3 provides a detailed illustration of the incremental effects of MPI-Flow with and without independent object motion and depth-aware inpainting. Novel view images and optical flows from single-view images can be generated with MPI-Flow and only camera motion, as shown in Figures 3(a) and 3(b). However, camera motion alone does not match the complex optical flows in real-world scenes. To address this issue, we introduce an independent object motion module, as shown in Figure 3(c), to ensure motion realism. To further enhance motion realism and address occlusion caused by object motion, we apply the depth-aware inpainting module, as shown in Figure 3(d).

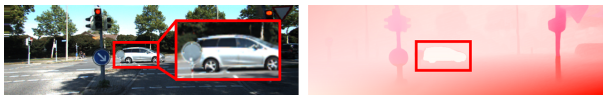
(a) Source Image



(b) + Camera Motion



(c) + Independent Object Motion



(d) + Depth-Aware Inpainting

Figure 3: Visualization of incrementally adding different modules to improve the realism of the generated data.

4. Experiments

4.1. Datasets

FlyingChairs [9] and FlyingThings3D [18] are both popular synthetic datasets that train optical flow models. As a standard practice, we use "Ch" and "Th" respectively to represent the two datasets, and "Ch→Th" means training first on "Ch" and fine-tuning on "Th". By default, we use the RAFT pre-trained on "Ch→Th" to be fine-tuned on the generated datasets and evaluated on labeled datasets.

COCO [27] is a collection of single still images and ground truth with labels for object detection or panoptic segmentation tasks. We sample 20k single-view still images from the train2017 split following Depthstillation [1] to generate virtual images and optical flow maps.

DAVIS [37] provides high-resolution videos and it is widely used for video object segmentation. We use all the 10581 images of the unsupervised 2019 challenge to generate datasets by MPI-Flow and other state-of-the-art optical flow dataset generation methods.

KITTI2012 [11] and KITTI2015 [34] are well-known benchmarks for optical flow estimation. There are multi-view extensions (4,000 for training and 3,989 for testing) datasets with no ground truth. We use the multi-view extension images (training and testing) of KITTI 2015 to generate datasets, separately. By default, we evaluate the trained models on KITTI 12 training set and KITTI 15 training set in the tables following [1] and [12], abbreviated as "KITTI 15" and "KITTI 12" in the following tables.

Sintel [5] is derived from the open-source 3D animated short film Sintel. The dataset has 23 different scenes. The stereo images are RGB, while the disparity is grayscale. Although not a real-world dataset, we use it to verify the model's generalization across domains.

4.2. Implementation Details

Firstly, we provide a description of the learning-based optical flow estimation models that were utilized in our experiments. Subsequently, we outline the experimental parameters and setup along with the evaluation formulation.

Optical Flow networks. To evaluate how effective our generated data are at training optical flow models, we select RAFT [45], which represents state-of-the-art architecture for supervised optical flow and has excellent generalization capability. By default, we train RAFT on generated data for 200K steps with a learning rate of 1×10^{-4} and weight decay of 1×10^{-5}, batch size of 6, and 288×960 image crops. This configuration is the default setting of RAFT fitting on KITTI with two GPUs but four times the number of training steps, following [12]. For the rest of the setup, we use the official implementation of RAFT without any modifications. All evaluations are performed on a single NVIDIA

Image Source	Method	Sintel.C		Sintel.F		KITTI 12		KITTI 15	
		EPE ↓	> 3 ↓	EPE ↓	> 3 ↓	EPE ↓	Fl ↓	EPE ↓	Fl ↓
COCO	Depthstillation [1]	**1.87**	5.31	3.21	9.25	1.74	6.81	3.45	13.08
	RealFlow [12]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	MPI-Flow (ours)	**1.87**	**4.59**	**3.16**	**8.29**	**1.36**	**4.91**	**3.44**	**10.66**
DAVIS	Depthstillation [1]	2.70	7.52	3.81	12.06	1.81	6.89	3.79	13.22
	RealFlow [12]	**1.73**	4.81	3.47	8.71	1.59	6.08	3.55	12.52
	MPI-Flow (ours)	1.79	**4.77**	**3.06**	**8.56**	**1.41**	**5.36**	**3.32**	**10.47**
KITTI 15 Test	Depthstillation [1]	4.02	9.08	4.96	13.23	1.77	5.97	3.99	13.34
	RealFlow [12]	3.73	7.36	5.53	11.31	1.27	5.16	2.43	8.86
	MPI-Flow (ours)	**2.25**	**5.25**	**3.65**	**8.89**	**1.24**	**4.51**	**2.16**	**7.30**
KITTI 15 Train	Depthstillation [1]	2.84	7.18	4.31	11.24	1.67	5.71	{2.99}	{9.94}
	RealFlow [12]	4.06	7.68	4.78	11.44	**1.25**	5.02	{2.17}	{8.64}
	MPI-Flow (ours)	**2.41**	**5.39**	**3.82**	**9.11**	1.26	**4.66**	{**1.88**}	{**7.16**}

Table 1: The cross-dataset validation results and comparisons with other dataset generation methods from real images or videos are presented in this study. The "Image Source" column indicates the dataset used for optical flow training data generation. The evaluation results of RAFT trained on different datasets using different methods are reported. In cases where RealFlow fails to work on single-view images from COCO, the study indicates "N/A". The curly braces "{}" represent the use of the unlabeled evaluation set, which is the KITTI 15 training set in this table.

GeForce RTX 3090 GPU [1].

Virtual Camera Motion. To generate the novel view images from COCO and DAVIS, we adopt the same settings in [1] to build the virtual camera. For KITTI, we empirically build the camera motion with three scalars where t_x, t_y are in $[-0.2, 0.2]$ and t_z are in $[0.1, 0.35]$. We build the camera rotation with three Euler angles a_x, a_y, a_z in $[-\frac{\pi}{90}, \frac{\pi}{90}]$. We use single camera motion for each image from COCO but multiple camera motions ($\times 4$ by default) from DAVIS and KITTI as in [12], due to the small number of images and the homogeneity of the scene in video data. We show how the number of camera motions impacts the optical flow network performance in the discussion.

Evaluation Metrics. We report evaluation results on the average End-Point Error (EPE) and two error rates, respectively the percentage of pixels with an absolute error greater than 3 (> 3) or both absolute and relative errors greater than 3 and 5% respectively (Fl) on all pixels.

4.3. Comparison with State-of-the-art Methods

In this section, we evaluate the effectiveness of the MPI-Flow generation pipeline on public benchmarks. We will highlight the best results in **bold** and underline the second-best if necessary among methods trained in fair conditions.

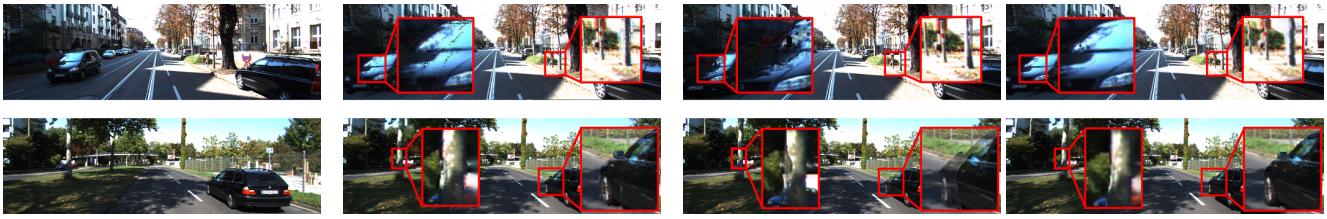
Comparison with Dataset Generation Methods. As a method for generating datasets from real-world images, we compare MPI-Flow with Depthstillation [1] and RealFlow [12], which are representative works in real-world dataset generation. In order to evaluate the effectiveness of MPI-Flow, we follow the procedures of Depthstillation and RealFlow to construct training sets from four different datasets, including COCO, DAVIS, KITTI 15 multi-view train set, and KITTI 15 multi-view test set. To ensure fair comparisons, we conduct experiments with a similar number of training sets as our competitors. Specifically, for DAVIS and KITTI, we generate four motions for each image to match RealFlow, which trains RAFT for four EM iterations with four times the amount of data. For COCO, we follow the exact same setup as Depthstillation. Since Depthstillation does not provide details for KITTI 15 train, we use its default settings to obtain the results. Trained models are evaluated on the training sets of Sintel, KITTI 12, and KITTI 15. We report the evaluation results of models with the best performance in the paper for Depthstillation and RealFlow. Furthermore, we conduct cross-dataset experiments where RAFT is trained with one generated dataset and evaluated with another.

As shown in Table 1, even with the same amount of data, our approach gains significant improvements and generalization over multiple datasets. When trained and tested with

Dataset	KITTI 12		KITTI 15	
	EPE ↓	Fl ↓	EPE ↓	Fl ↓
Ch→Th [18]	2.08	8.86	5.00	17.44
dDAVIS [1]	1.81	6.89	3.79	13.22
MF-DAVIS	1.61	6.41	3.77	12.40
dCOCO [1]	1.80	6.66	3.80	12.44
MF-COCO	**1.59**	**6.22**	**3.68**	**11.95**

Table 2: Results on RAFT trained from scratch under the same setting. "dX" are from Depthstillation [1] while "MF-X" are from our proposed MPI-Flow

[1] RAFT with the same parameters loaded on different GPUs yield slightly different evaluation results. Therefore, to ensure fair comparison, we download the official model weights with the best performance provided by the compared methods and evaluate them on the same GPU.



(a) Source Image (b) Depthstillation (c) RealFlow (d) MPI-Flow (ours)

Figure 4: Qualitative results of generated images and optical flows from the KITTI 2015 training set. (a) contains the input source images. (b), (c), and (d) contain generated images of RealFlow [12], Depthstillation [1], and our proposed MPI-Flow. MPI-Flow eliminates the artifact in the new image and guarantees image realism. (Best viewed with zoom-in)

(a) Source Image from KITTI 15

(b) Depthstillation (c) RealFlow (d) MPI-Flow (ours)

Figure 5: Visualization of generated images with Depthstillation, RealFlow, and our proposed MPI-Flow from KITTI. Note that RealFlow generates such data using two frames with estimated optical flow.

(a) Depthstillation (b) MPI-Flow (ours)

Figure 6: Visualization of generated images with Depthstillation and our proposed MPI-Flow using a single view.

the same KITTI 15 Train image source, our EPE outperforms the second-best by a remarkable 0.29. When trained and tested with different image sources, our MPI-Flow demonstrates clear improvements over the competitors on almost all the evaluation settings. Notably, MPI-Flow achieves much better performance, even though RealFlow requires two consecutive frames to generate datasets, while MPI-Flow needs only one still image.

Method	KITTI 12		KITTI 15	
	EPE ↓	Fl ↓	EPE ↓	Fl ↓
SemiFlowGAN [25]	-	-	{16.02}	{38.77}
FlowSupervisor [20]	-	-	{3.35}	{11.12}
DistractFlow [21]	-	-	{3.01}	{11.7}
Meta-Learning [36]	-	-	{2.81}	-
SimFlow [19]	-	-	{5.19}	-
ARFlow [28]	1.44	-	{2.85}	-
UFlow [22]	1.68	-	2.71	9.05
UpFlow [33]	1.27	-	2.45	-
SMURF [42]	-	-	{2.00}	{**6.42**}
MPI-Flow	**1.18**	**4.46**	{**1.80**}	{6.63}

Table 3: Comparison with semi-supervised and unsupervised methods. '-' indicates no results reported.

It is worth comparing MPI-Flow with Depthstillation under the same settings to evaluate their performance. For this comparison, we use the exact same settings as Depthstillation. Specifically, we generate MPI-Flow datasets with 1) no object motion, 2) the same camera parameters, 3) one camera motion per image, and 4) without pre-training on Ch→Th. The datasets generated are named MF-DAVIS and MF-COCO, while dCOCO and dDAVIS are generated from Depthstillation. The models are trained from scratch with the respective datasets. RealFlow is not included in this table as it requires a pre-trained optical flow model to generate datasets. The evaluation results are shown in Table 2. Our method outperforms Depthstillation with significant improvements on both COCO and DAVIS, demonstrating the importance of image realism.

Comparison with Unsupervised Methods. Another way to utilize real-world data is through unsupervised learning, which learns optical flow pixels directly without the need for optical flow labels. In order to further demonstrate the effectiveness of MPI-Flow, we compare our method with the existing literature on unsupervised methods. The results of this comparison can be seen in Table 3. All methods are evaluated under the condition that only images from

Dynamic Objects	Depth-Aware Inpainting	Multiple Objects	Sintel.C		Sintel.F		KITTI 12		KITTI 15	
			EPE ↓	> 3 ↓	EPE ↓	> 3 ↓	EPE ↓	Fl ↓	EPE ↓	Fl ↓
×	×	×	2.58	6.10	4.04	9.67	**1.20**	**4.34**	{2.46}	{8.38}
✓	×	×	2.41	5.39	3.82	9.11	1.26	4.66	{**1.88**}	{7.16}
✓	✓	×	2.37	5.44	3.71	9.05	1.25	4.58	{1.91}	{7.06}
✓	✓	✓	**2.20**	**5.35**	**3.67**	**9.03**	1.23	4.46	{1.92}	{**7.05**}

Table 4: Ablation experiments. Settings used are marked with a checkmark. Here we only perform four camera motions per image for these experiments due to the limitation of computational resources.

Method	KITTI 15		KITTI 15 test
	EPE ↓	Fl ↓	Fl ↓
PWC-Net [44]	{2.16}	{9.80}	9.60
LiteFlowNet [16]	{1.62}	{5.58}	9.38
IRR-PWC [17]	{1.63}	{5.32}	7.65
RAFT [45]	{0.63}	{1.50}	5.10
Depthstillation [12]	-	-	-
AutoFlow [43]	-	-	4.78
RealFlow [12]	{**0.58**}	{1.35}	4.63
MPI-Flow	{**0.58**}	{**1.30**}	**4.58**

Table 5: Comparison with supervised methods fine-tuned or trained on KITTI 15 train set.

Motions Per Image	KITTI 12		KITTI 15	
	EPE ↓	Fl ↓	EPE ↓	Fl ↓
1	1.29	4.58	2.02	7.35
4	1.23	4.46	1.92	7.05
10	1.21	4.49	1.82	7.02
20	1.20	**4.41**	**1.79**	6.86
40	**1.18**	4.46	1.80	**6.63**

Table 6: Effect of amount of virtual camera motions (training pairs) per source image.

Objects Per Image	KITTI 12		KITTI 15	
	EPE ↓	Fl ↓	EPE ↓	Fl ↓
1	1.25	4.58	**1.91**	7.06
2	1.26	4.76	2.02	7.19
4	**1.23**	**4.46**	1.92	**7.05**

Table 7: Effect of amount of dynamic objects per image.

the evaluation set could be used, without access to ground truth labels. For evaluation, we train the RAFT on our generated dataset with images from the KITTI 15 training set. Our MPI-Flow outperform all unsupervised methods in terms of EPE on both the KITTI 12 training set and KITTI 15 training set, with no need for any unsupervised constraints. However, our method performs better on EPE but has slightly lower Fl than SMURF, mainly because SMURF employs multiple frames for training.

Comparison with Supervised Methods. To further prove the effectiveness of MPI-Flow, we use KITTI 15 train set to fine-tune RAFT pre-trained by our generated dataset with images from KITTI 15 test set. The evaluation results on KITTI 15 train and KITTI 15 test are shown in Table 5. We achieve state-of-art performance on KITTI 2015 test benchmark compared to supervised methods on training RAFT.

Qualitative Results. Figure 4 show the generated images from the methods utilizing real-world images, as presented in Table 1. In this comparison, we use images from the KITTI 15 dataset as source image input. The images generated by RealFlow [12] and Depthstillation [1] with artifacts degrade the image realism. In contrast, MPI-Flow generates more realistic images than the other two methods. More results are shown in Figures 5 and 6.

4.4. Discussion

To verify the effectiveness of the proposed MPI-Flow, we discuss the performance of models trained with generated datasets with different settings. We show more discussion and evaluation results in the supplementary material.

Object Motion and Depth-Aware Inpainting. We conduct a series of ablation studies to analyze the impact of different choices in our proposed MPI-Flow for new image synthesis, including object motion, depth-aware inpainting, and multiple objects. "Multiple objects" indicates multiple moving objects in each image. To measure the impact of these factors, we generate new images from the KITTI 15 training set to train RAFT and evaluate them on the KITTI 12 training set and KITTI 15 training set. Because there are multiple combinations of these factors, we test by incrementally adding components of our approach, as shown in Table 4. In the first row, we show the performance achieved by generating new images without dynamic objects, thus assuming that optical flow comes from camera motion only. Then we incrementally add single-object motion, depth-aware inpainting, and multi-object motion to model a more realistic scenario. There are considerable improvements in almost all datasets on various metrics, except for the KITTI 12 training set, possibly due to the infrequent dynamic object motion in this dataset. The EPE on KITTI 15 remains relatively stable within the normal margin after adding the depth-aware inpainting module and the multi-object trick.

Camera Motion Parameters. We also conduct a series of

t_z	KITTI 12		KITTI 15	
	EPE ↓	Fl ↓	EPE ↓	Fl ↓
$0.1 \sim 0.45$	1.25	4.60	{1.79}	{6.68}
$\underline{0.1 \sim 0.35}$	1.18	4.46	**{1.80}**	**{6.63}**
$0.1 \sim 0.25$	1.19	4.50	{1.86}	{6.71}
$0.0 \sim 0.35$	**1.12**	**4.32**	{1.94}	{6.92}

Table 8: Effect of motion parameters t_z.

t_x and t_y	KITTI 12		KITTI 15	
	EPE ↓	Fl ↓	EPE ↓	Fl ↓
$-0.3 \sim 0.3$	1.22	**4.42**	{1.92}	{6.83}
$\underline{-0.2 \sim 0.2}$	**1.18**	4.46	**{1.80}**	**{6.63}**
$-0.1 \sim 0.1$	1.32	4.99	{1.98}	{7.02}

Table 9: Effect of motion parameters t_x and t_y.

Dataset	KITTI 12		KITTI 15	
	EPE ↓	Fl ↓	EPE ↓	Fl ↓
Ch→Th [18]	4.14	21.38	10.35	33.67
dDAVIS [1]	2.81	11.29	**6.88**	21.87
MF-DAVIS	**2.70**	**11.25**	6.92	**20.97**
dCOCO [1]	3.16	13.30	8.49	26.06
MF-COCO	**3.02**	**11.20**	**8.22**	**23.40**

Table 10: Results on PWC-Net trained on generated datasets from scratch under the same setting.

ablation studies to analyze the impact of different parameters on camera motions. We first show the effect of t_z, which indicates the range of moving forward and backward distances, as shown in Table 4. We set the minimum t_z to 0.1 by default, considering that most cameras on vehicles only move forward in the KITTI dataset. We also show the effect of t_x and t_y, representing the distance from left to right and from up to down, respectively. Because there are multiple combinations of parameters, we only test a specific parameter of our method in isolation. Settings used by default are underlined. The results show that a more reasonable range of camera motion leads to better performance.

Model Architectures The default model used in our paper is RFAT [45]. Table 10 shows that our proposed MPI-Flow also works for PWC-Net [44] compared with depthstillation [1] in a fair setting. We generate data from COCO and DAVIS and train PWC-Net, in which the results are still better than PWC-Net trained on Ch→Th or datasets generated from depthstillation.

Amount of Virtual Motions. We can generate multiple camera motions with multiple dynamic objects for any given single image and thus a variety of paired images and ground-truth optical flow maps. Thus we can increase the number of camera motions to generate more data. Table 6 shows the impact of different amounts of camera motions on model performance. Interestingly, MPI-Flow with 4 motions per image already allows for strong generalization to real domains, outperforming the results achieved using synthetic datasets shown in the previous evaluation results. Increasing the motions per image by factors 10 and 20 both lead to better performance on KITTI 12 and KITTI 15 compared to 4 and 1. Using 40 motions per image gives the best performance on KITTI 15 in terms of Fl. It indicates that a more variegate image content in the generated dataset may be beneficial for generalization to real applications. Table 7 shows the effect of amounts of dynamic objects on model performance. Increasing the number of dynamic objects improves the performance of the model on KITTI 12 but slightly decreases it on KITTI 15 in terms of EPE.

Dataset	Quantity	KITTI 12		KITTI 15	
		EPE ↓	Fl ↓	EPE ↓	Fl ↓
Ch→Th [18]	47K	2.08	8.86	5.00	17.44
MF-COCO	20K	1.59	6.22	3.68	11.95
MF-COCO	120K	**1.51**	**5.94**	**3.41**	**11.16**

Table 11: Effect of amount of source images.

Quantity of Source Images The number of source images affects the scene diversity of the generated dataset. Empirically, more source images will be more conducive to model learning, as verified in Table 11. We verify the effect of the number of source images on the MF-COCO. The model performance is significantly improved by increasing the number of source images.

5. Conclusion

In this paper, we present a new framework for generating optical flow datasets, which addresses two main challenges: image realism and motion realism. Firstly, we propose an MPI-based image rendering pipeline that generates realistic images with corresponding optical flows from novel viewpoints. This pipeline utilizes volume rendering to address image artifacts and holes, leading to more realistic images. Secondly, we introduce an independent object motion module that separates dynamic objects from the static scene. By decoupling object motion, we further improve motion realism. Additionally, we design a depth-aware inpainting module that handles unnatural occlusions caused by object motion in the generated images. Through these novel designs, our approach achieves superior performance on real-world datasets compared to both unsupervised and supervised methods for training learning-based models.

Acknowledgments This work was supported by the National Natural Science Foundation of China (12202048, 62171038, and 62171042), the R&D Program of Beijing Municipal Education Commission (KZ202211417048), and the Fundamental Research Funds for the Central Universities.

References

[1] Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning optical flow from still images. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15201–15211, 2021. 1, 2, 3, 5, 6, 7, 8, 9

[2] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision (IJCV)*, 92(1):1–31, 2011. 1, 2

[3] Thomas Brox, Christoph Bregler, and Jitendra Malik. Large displacement optical flow. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 41–48, 2009. 1

[4] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *the European Conference on Computer Vision (ECCV)*, pages 611–625, 2012. 1

[5] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *the European Conference on Computer Vision (ECCV)*, pages 611–625, 2012. 5

[6] Gustavo Sutter P. Carvalho, Diogo Carbonera Luvizon, Antonio Joia Neto, André G. C. Pacheco, and Otávio Augusto Bizetto Penatti. Learning multiplane images from single views with self-supervision. In *British Machine Vision Conference (BMVC)*, 2021. 2

[7] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1299, 2022. 4

[8] Jiyu Cheng, Yuxiang Sun, and Max Q-H Meng. Improving monocular visual slam in dynamic environments: an optical-flow-based approach. *Advanced Robotics*, 33(12):576–589, 2019. 1

[9] 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 *the IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015. 1, 2, 5

[10] Bo Du, Shihan Cai, and Chen Wu. Object tracking in satellite videos based on a multiframe optical flow tracker (j-stars). *the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(8):3043–3055, 2019. 1

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012. 1, 2, 5

[12] Yunhui Han, Kunming Luo, Ao Luo, Jiangyu Liu, Haoqiang Fan, Guiming Luo, and Shuaicheng Liu. Realflow: Em-based realistic optical flow dataset generation from videos. In *the European Conference on Computer Vision (ECCV)*, pages 288–305, 2022. 1, 2, 3, 5, 6, 7, 8

[13] Yuxuan Han, Ruicheng Wang, and Jiaolong Yang. Single-view view synthesis in the wild with learned adaptive multiplane images. In *ACM SIGGRAPH*, 2022. 2, 3, 4

[14] Toshihide Hanari, Kuniaki Kawabata, and Keita Nakamura. Image selection method from image sequence to improve computational efficiency of 3d reconstruction: Analysis of inter-image displacement based on optical flow for evaluating 3d reconstruction performance. In *the IEEE International Symposium on System Integration (SII)*, pages 1041–1045, 2022. 1

[15] Anders Heyden and Marc Pollefeys. Multiple view geometry. *Emerging topics in computer vision*, 3:45–108, 2005. 4

[16] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8981–8989, 2018. 2, 8

[17] Junhwa Hur and Stefan Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5754–5763, 2019. 2, 8

[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. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2462–2470, 2017. 1, 2, 5, 6, 9

[19] Woobin Im, Tae-Kyun Kim, and Sung-Eui Yoon. Unsupervised learning of optical flow with deep feature similarity. In *the European Conference on Computer Vision (ECCV)*, pages 172–188, 2020. 7

[20] Woobin Im, Sebin Lee, and Sung-Eui Yoon. Semi-supervised learning of optical flow by flow supervisor. In *the European Conference on Computer Vision (ECCV)*, pages 302–318, 2022. 7

[21] Jisoo Jeong, Hong Cai, Risheek Garrepalli, and Fatih Porikli. Distractflow: Improving optical flow estimation via realistic distractions and pseudo-labeling. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13691–13700, 2023. 7

[22] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. In *the European Conference on Computer Vision (ECCV)*, pages 557–572, 2020. 7

[23] Artúr I Károly, Renáta Nagyné Elek, Tamás Haidegger, Károly Széll, and Péter Galambos. Optical flow-based segmentation of moving objects for mobile robot navigation using pre-trained deep learning models. In *the IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pages 3080–3086, 2019. 1

[24] Filippos Kokkinos and Iasonas Kokkinos. Learning monocular 3d reconstruction of articulated categories from motion. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1737–1746, 2021. 1

[25] Wei-Sheng Lai, Jia-Bin Huang, and Ming-Hsuan Yang. Semi-supervised learning for optical flow with generative adversarial networks. *Advances in Neural Information Processing Systems (NIPS)*, 30, 2017. 7

[26] Jiaxin Li, Zijian Feng, Qi She, Henghui Ding, Changhu Wang, and Gim Hee Lee. Mine: Towards continuous depth

mpi with nerf for novel view synthesis. In *the IEEE International Conference on Computer Vision (ICCV)*, pages 12578–12588, 2021. 3

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *the European Conference on Computer Vision (ECCV)*, pages 740–755, 2014. 5

[28] 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 *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6489–6498, 2020. 7

[29] Yubao Liu and Jun Miura. Rdmo-slam: Real-time visual slam for dynamic environments using semantic label prediction with optical flow. *IEEE Access*, 9:106981–106997, 2021. 1

[30] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *arXiv preprint arXiv:1906.07751*, 2019. 2

[31] Ao Luo, Fan Yang, Xin Li, and Shuaicheng Liu. Learning optical flow with kernel patch attention. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8906–8915, 2022. 2

[32] Ao Luo, Fan Yang, Kunming Luo, Xin Li, Haoqiang Fan, and Shuaicheng Liu. Learning optical flow with adaptive graph reasoning. In *the AAAI Conference on Artificial Intelligence*, volume 36, pages 1890–1898, 2022. 2

[33] Kunming Luo, Chuan Wang, Shuaicheng Liu, Haoqiang Fan, Jue Wang, and Jian Sun. Upflow: Upsampling pyramid for unsupervised optical flow learning. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1045–1054, 2021. 7

[34] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3070, 2015. 1, 2, 5

[35] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[36] Chaerin Min, Taehyun Kim, and Jongwoo Lim. Meta-learning for adaptation of deep optical flow networks. In *the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 2145–2154, 2023. 7

[37] Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 724–732, 2016. 5

[38] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 44(3):1623–1637, 2020. 4

[39] Chris Rockwell, David F Fouhey, and Justin Johnson. Pixelsynth: Generating a 3d-consistent experience from a single image. In *the IEEE International Conference on Computer Vision (ICCV)*, pages 14104–14113, 2021. 3

[40] Robin Rombach, Patrick Esser, and Björn Ommer. Geometry-free view synthesis: Transformers and no 3d priors. In *the IEEE International Conference on Computer Vision (ICCV)*, pages 14356–14366, 2021. 3

[41] Nitin J Sanket, Chahat Deep Singh, Cornelia Fermüller, and Yiannis Aloimonos. Prgflow: Unified swap-aware deep global optical flow for aerial robot navigation. *Electronics Letters*, 57(16):614–617, 2021. 1

[42] Austin Stone, Daniel Maurer, Alper Ayvaci, Anelia Angelova, and Rico Jonschkowski. Smurf: Self-teaching multi-frame unsupervised raft with full-image warping. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3887–3896, 2021. 7

[43] Deqing Sun, Daniel Vlasic, Charles Herrmann, Varun Jampani, Michael Krainin, Huiwen Chang, Ramin Zabih, William T Freeman, and Ce Liu. Autoflow: Learning a better training set for optical flow. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10093–10102, 2021. 2, 8

[44] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2018. 1, 2, 8, 9

[45] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *the European Conference on Computer Vision (ECCV)*, pages 402–419, 2020. 1, 2, 5, 8, 9

[46] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of graphics tools*, 9(1):23–34, 2004. 4

[47] Richard Tucker and Noah Snavely. Single-view view synthesis with multiplane images. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 551–560, 2020. 2, 3

[48] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *the European Conference on Computer Vision (ECCV)*, pages 302–317, 2018. 2

[49] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *the IEEE International Conference on Computer Vision (ICCV)*, pages 1385–1392, 2013. 1, 2

[50] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Pattern Recognition*, pages 214–223, 2007. 1

[51] Jimuyang Zhang, Sanping Zhou, Xin Chang, Fangbin Wan, Jinjun Wang, Yang Wu, and Dong Huang. Multiple object tracking by flowing and fusing. *arXiv preprint arXiv:2001.11180*, 2020. 1

[52] Tianwei Zhang, Huayan Zhang, Yang Li, Yoshihiko Nakamura, and Lei Zhang. Flowfusion: Dynamic dense rgb-d

slam based on optical flow. In *the IEEE International Conference on Robotics and Automation (ICRA)*, pages 7322–7328, 2020. 1

[53] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *ACM SIGGRAPH*, 2018. 2

[54] Yuemei Zhou, Gaochang Wu, Ying Fu, Kun Li, and Yebin Liu. Cross-mpi: Cross-scale stereo for image super-resolution using multiplane images. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14842–14851, 2021. 2