# ADFactory: An Effective Framework for Generalizing Optical Flow with NeRF

Han Ling[1]    Quansen Sun[1*]    Yinghui Sun[2†]    Xian Xu[2]    Xingfeng Li[1]
[1]Nanjing University of Science and Technology    [2]Southeast University

321106010190@njust.edu.cn

## Abstract

*A significant challenge facing current optical flow methods is the difficulty in generalizing them well to the real world. This is mainly due to the lack of large-scale real-world datasets, and existing self-supervised methods are limited by indirect loss and occlusions, resulting in fuzzy outcomes. To address this challenge, we introduce a novel optical flow training framework: automatic data factory (ADF). ADF only requires RGB images as input to effectively train the optical flow network on the target data domain. Specifically, we use advanced NeRF technology to reconstruct scenes from photo groups collected by a monocular camera, and then calculate optical flow labels between camera pose pairs based on the rendering results. To eliminate erroneous labels caused by defects in the scene reconstructed by NeRF, we screened the generated labels from multiple aspects, such as optical flow matching accuracy, radiation field confidence, and depth consistency. The filtered labels can be directly used for network supervision. Experimentally, the generalization ability of ADF on KITTI surpasses existing self-supervised optical flow and monocular scene flow algorithms. In addition, ADF achieves impressive results in real-world zero-point generalization evaluations and surpasses most supervised methods[1].*

## 1. Introduction

Optical flow aims to estimate the motion of each pixel on the image plane between two consecutive frames. It is one of the oldest problems in computer vision and has essential applications in autonomous driving[10, 38, 39], video understanding[11], and human action recognition[1, 18]. In recent years, deep learning methods[31, 33] have become the mainstream solution for optical flow and achieved performance far superior to traditional handcrafted methods [6] in benchmark tests[9, 27].

Compared to traditional methods, deep optical flow methods can learn robust matching features and contextual

---

features from large-scale data, as well as infer the optical flow of occluded parts[14]. However, acquiring real-world optical flow datasets is challenging. It requires additional sensors such as LiDAR, GPS, IMU, and much manual annotations[27], which has been one of the significant limitations of applying optical flow methods[44]. Although recent works in synthetic datasets and self-supervised learning have alleviated this limitation to some extent, they are still subject to constraints in domain transferability[9, 25, 26], lighting variations, and occlusions[15, 17].

In this era where data is the gold mine, massive training datasets are a necessary factor for the success of powerful models such as ChatGPT[8] and SAM[16]. However, how to quickly and cost-effectively mine high-quality training data from the 'gold mines', and fully tap into the enormous potential of deep networks remains a challenge to be explored [48]. This paper proposes a novel automated data factory (ADF) to address this issue. ADF utilizes only photos taken by a single camera as input, and does not require manual intervention or ground-truth labels. It can quickly generate massive datasets at a meagre cost, training various geometric matching tasks. Specifically, ADF mainly consists of two parts: data generation and data filtering.

**Data generation:** We build a data generator based on the latest radiation rendering technology (NeRF)[5]. Firstly, based on NeRF, a high-resolution scene is reconstructed from photos taken by a monocular camera. Then, by rendering a depth map and randomly generating new camera pose pairs, we can calculate the corresponding optical flow results between poses. In addition, to alleviate the problem that NeRF can only reconstruct static scenes with high quality, we also added randomly shaped foreground floaters based on the Bezier curve.

**Data filtering:** In NeRF reconstruction, it is inevitable to encounter areas of reconstruction failure, resulting in further generation of erroneous training data. Therefore, constructing a comprehensive data filtering mechanism is essential. Before us, the only evaluation metric used was ambient occlusion (AO) [34, 35]. However, AO can only be used to check whether floating objects are between the viewpoint and the target depth, which cannot be widely used
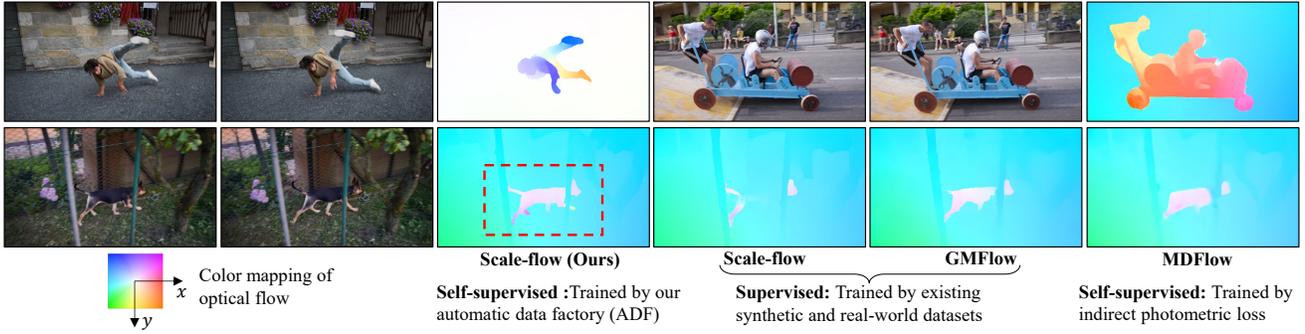
**Figure 1. Zero-Shot Generalization Results in Real World.** On the top is Scale-flow using our automatic data factory (ADF) scheme to estimate optical flow results in real-world scenarios. Below is a comparison with existing advanced supervised and self-supervised methods. ADF shows unprecedented accuracy and clarity. Moreover, it requires no manual intervention and only utilizes photos captured by a monocular camera to train optical flow tasks.

for NeRF methods and downstream tasks. In this paper, we have designed practical evaluation metrics such as reconstruction stability, depth consistency, and optical flow visual consistency based on the characteristics of the NeRF method. Experiments have shown that the combination of these metrics improves the overall performance of the optical flow method by at least 20%.

The filtered optical flow results can be directly used to train complex geometric matching tasks[21, 22, 32, 42–44]. As shown in Fig.1, the optical flow model trained on ADF-generated data achieves stunning real-world generalization effects. We believe that ADF is an important step in making optical flow methods widely applicable in real-world scenarios and a stepping stone to building matching large models. In this paper, we only used about 300 collected scenarios, which has made the ADF-supervised optical flow methods superior in the real-world zero-shot generalization to self-supervised and supervised models trained on existing datasets [9, 25, 27].

Our contributions can be summarized as follows:

- A new mode for self-supervised learning of optical flow algorithms based on NeRF.
- A fully automatic optical flow label generation pipeline ADF, which only requires images taken by a monocular camera as input to generate massive high quality datasets.
- We collected and created an optical flow dataset ADF58 containing over 300 scenes and 58800 images. Experimental results demonstrate that ADF58 achieves superior zero-shot generalization capability in optical flow task compared to existing datasets.

## 2. Preliminaries

In this section, we mainly introduce the components of the NeRF method used in this paper and analyze how to render the most critical depth values.

We use the Zip-NeRF[5] to represent 3D scenes and im-age generation. This is currently one of the most advanced NeRF methods in terms of performance, which can achieve anti-aliasing and robust reconstruction results. Secondly, it uses hash voxel networks to store color and density features. This explicit expression not only accelerates convergence speed, but also naturally conforms to multi-view geometric structure consistency better than implicit expression.

**Scene representation:** Zip-NeRF's neural radiation field is composed of the hash voxel field in instant-NGP[29], mapping the distance interval $T_i = [t_i, t_{i+1})$ on a ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ into a set of color features $\mathbf{c} \in [0,1]^3$ and volume density $\sigma \in \mathbb{R}^+$, where $\mathbf{o}$ and $\mathbf{d}$ are the origin and direction of the ray, respectively, $t$ is the distance from the origin along the ray direction. It can be formulated as:

$$(\sigma_i, \mathbf{c}_i) = \mathrm{MLP}_\theta(\gamma(\mathbf{r}(T_i))), \quad \forall T_i \in \mathbf{t} \tag{1}$$

Here, $\mathrm{MLP}_\theta$ is an shallow MLP with parameters $\theta$, $\mathbf{r}(T_i)$ maps the sampling points in the conical frustum corresponding to $T_i$, $\gamma$ is the coding function, and $\mathbf{t}$ is the set of all intervals on the ray that are included in the rendering.

**Volume rendering:** With these volume densities and colors, we can calculate the RGB values $\mathbf{C}$ corresponding to the rays based on the volume rendering formula:

$$\mathbf{C}(\mathbf{r}, \mathbf{t}) = \sum_i w_i \mathbf{c}_i \tag{2}$$

$$w_i = \left(1 - e^{-\sigma_i(t_{i+1}-t_i)}\right) E_i \tag{3}$$

$$E_i = e^{-\sum_{j<i} \sigma_j(t_{j+1}-t_j)} \tag{4}$$

Where $E_i$ is the accumulated transmittance along the ray from viewpoint to $t_i$, $\left(1 - e^{-\sigma_i(t_{i+1}-t_i)}\right)$ is the opacity at $T_i$. By construction, the sum of the weights $w_i$ on a ray is always less than or equal to 1, and when the ray points to an opaque surface, the sum of the weights $w_i$ approaches 1 [4].
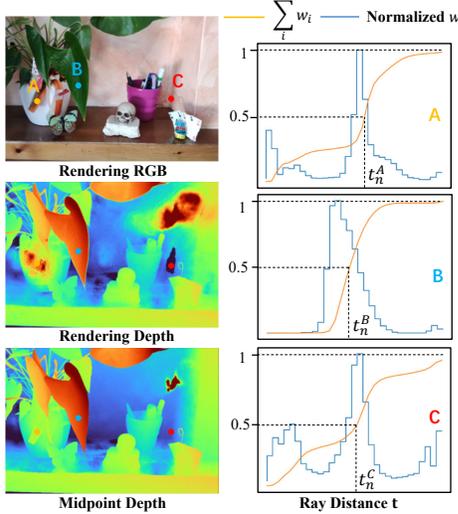
**Figure 2. Midpoint Depth vs. Rendering Depth.** Left: Rendered RGB image and two different depths. Right: The weights $w$ on three different rays: well-trained ray B, ray C with incorrect surfaces, and ray A with potential multiple surfaces. We found that rendering depth in A and C is clearly incorrect, as the weighted depth of the wrong surface interfered with the final result, especially when there were incorrect weights at the far end of the ray. And the midpoint depth can reduce the interference of these erroneous surfaces.

Usually, the scene depth calculation formula corresponding to a ray can be written as:

$$z(\mathbf{r}, \mathbf{t}) = \sum_i w_i t_{mid}, \quad t_{mid} = \frac{(t_i + t_{i+1})}{2} \quad (5)$$

Here, $z(\mathbf{r}, \mathbf{t})$ is the rendering depth of ray $\mathbf{r}$. As the sum of $w_i$ always approaches 1. We can also consider Eq (5) as calculating the expected value, and $w_i$ is the probability of the existence of an object surface at position $t_i$.

**Midpoint depth:** Although most NeRF-based generative methods[34, 35, 46, 47] use the depth described in Eq (5), it is unreasonable in Zip-NeRF. As mentioned in the previous chapter, its definition is the expected distance between all surfaces on the ray. So, when there are multiple surfaces or incorrect surfaces on a ray (due to insufficient NeRF reconstruction), it is highly likely to generate errors.

In this paper, we use midpoint depth as the rendering depth output, which is defined as follows:

$$z(\mathbf{r}, \mathbf{t}) = t_n, \quad \sum_i^n w_i = 0.5 \quad (6)$$

Due to the anti-aliasing characteristics of Zip-NeRF, the sum of weight $w$ on a well-trained ray should appear as a smoothed step function, as shown in Fig.2 B. At this point, the midpoint distance is close to the peak of $w$ (Location

with the highest potential surface probability). As shown in Fig.2 A and C, when there are potential multiple surfaces or incorrect surfaces on the ray, the midpoint depth can also avoid small errors and approach the correct depth.

## 3. Method

Fig.3 shows our data factory's workflow, which is divided into two stages. Firstly, we construct a neural radiation field based on static scene multi-view images users collected, and generate preliminary optical flow results. Then filter out the unqualified results. The filtered optical flow results can be directly used for network training.

### 3.1. Optical flow generation

After giving the camera pose pair $P_i$ and $P_j$, we can obtain the RGB images $I_i, I_j$ and the corresponding depth $Z_i, Z_j$ based on NeRF rendering. Among them $I_i(u, v) = \mathbf{C}(\mathbf{r}, \mathbf{t})$, $Z_i(u, v) = z(\mathbf{r}, \mathbf{t})$, $(u, v)$ is the pixel plane coordinate corresponding to ray $(\mathbf{r}, \mathbf{t})$, $P$ is the $3 \times 4$ transformation matrix from the camera coordinate system to the world coordinate system.

Based on the $Z_i$ and the camera pose $P_i, P_j$, we can calculate the position of the pixel $p_i(u, v) = (u, v, 1)$ in the first frame to corresponding $p_{i'}(u, v) = (u', v', 1)$ in the second frame, as follows:

$$p_{i'}(u, v) = \frac{K^{-1} Z_i(u, v) K p_i(u, v) P_i P_j^{-1}}{Z_{i'}(u, v)} \quad (7)$$

$$f_{i \to j} = p_{i'} - p_i \quad (8)$$

Where $f_{i \to j}$ is the optical flow between frames $i$ and $j$, $K$ is the camera's internal reference matrix to convert pixel points into the camera coordinate system, $Z_{i'}(u, v)$ is the depth of $p_i$ in frame $j$ calculated by projection. For ease of expression, we have moved it to the right side.

**Dynamic foreground:** Because Zip-NeRF can only model static backgrounds. To make up for this weakness, we use Bezier curves to form randomly shaped slices and make perspective changes between two frames to enhance the foreground part of the generated data, as shown in Fig.3.

In addition to the optical flow results, based on $Z_i$ and $Z_j$, we can also calculate scene flow datasets. For more information on datasets and foreground masks, please refer to **supplementary materials**.

### 3.2. Data Credibility

During shooting, the environment may change and the shooting perspective may not be perfect, which can result in artifacts in the data generated by NeRF. To address this issue, we require a reliable filtering mechanism to screen out unqualified data. In previous studies[35], researchers
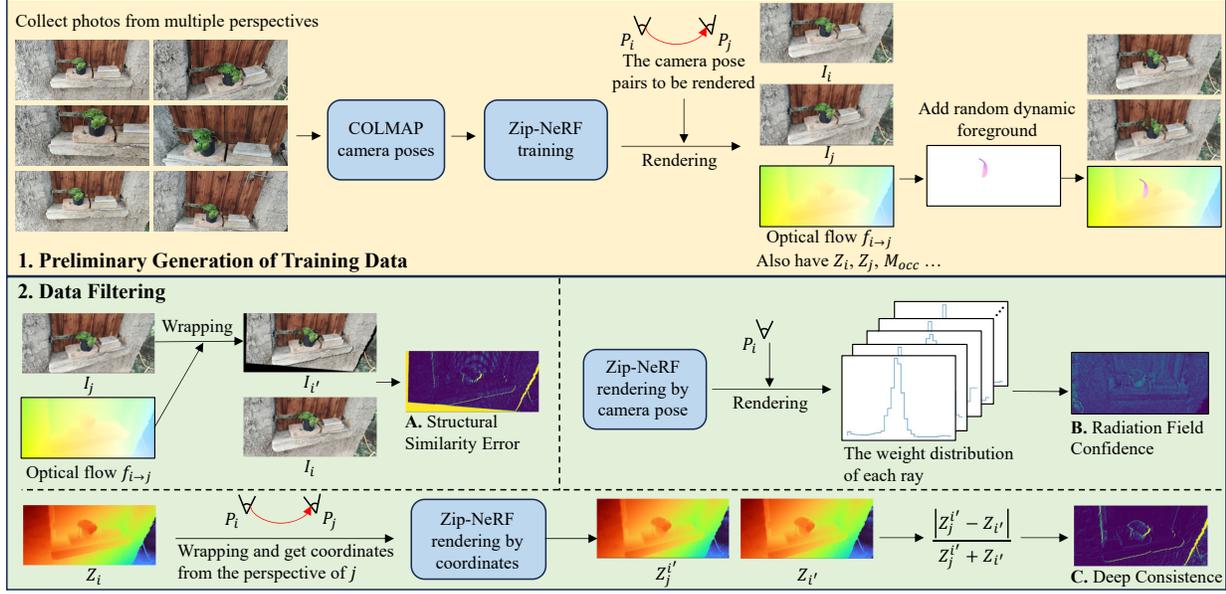
Figure 3. **Data Factory Workflow.** On the top: Training NeRF using a sequence of photos collected by a monocular camera and rendering preliminary optical flow labels. Below: Conduct confidence checks on the rendered data and labels, eliminating non-conforming parts.

have used ambient occlusion (AO) to measure the credibility of generated data. The AO can be calculated for a ray using the following formula:

$$\text{AO} = \sum_{i}^{m-1} w_i, \quad t_m = Z_i(u, v) \tag{9}$$

The meaning of AO is the probability of the presence of surfaces from the rendering depth $Z_i(u, v)$ to the viewpoint. However, AO is not applicable in anti-aliasing methods because $w$ has been smoothed, so even ordinary rays will have high AO values. In this article, we only use AO to calculate occlusion mask $M_{occ}$ between two frames. At this time, we use $Z_{i'}$ in Eq.7 to replace the rendering depth. The specific calculation process is as follows:

$$\text{AO}_{\text{occ}} = \sum_{j}^{n-1} w_j, \quad t_n = Z_{i'}(u, v) \tag{10}$$

$$M_{occ}(u, v) = \begin{cases} 0 & \text{if } \text{AO}_{\text{occ}} > th \\ 1 & \text{otherwise} \end{cases} \tag{11}$$

Where $\text{AO}_{\text{occ}}$ is the probability of the presence of a surface between the viewpoint and the depth $Z_{i'}(u, v)$, $M_{occ}$ is the occlusion mask between two frames, and in experiments, $th$ is generally set to 0.3.

In the following section, we propose different credibility masks to filter the generated data of NeRF from three perspectives: optical flow matching accuracy, NeRF rendering stability, and depth consistency. Combining these indicators
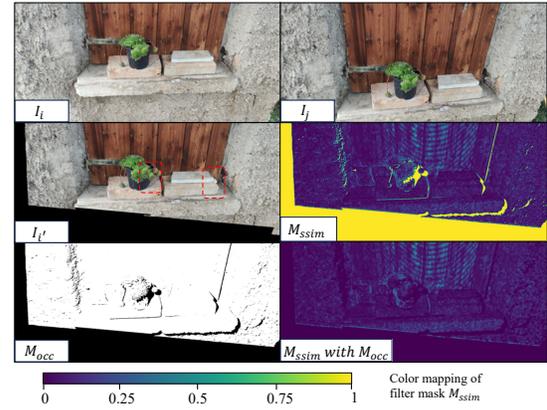


Figure 4. **SSIM Inspection of Optical Flow Labels.** We got $I_{i'}$ by warping $I_j$ to $I_i$ based on optical flow, and calculated the $M_{ssim}$ between $I_{i'}$ and $I_i$, where the darker the color, the smaller the error. Notice that the part in the red box has artifacts due to occlusion. When using $M_{ssim}$ as the evaluation standard, we filtered out these artifacts using an occlusion mask $M_{occ}$ to reflect the matching quality of the optical flow truly.

can find generated data labels that conform to geometric relationships.

**Optical flow matching accuracy:** As shown in Fig.4, to evaluate the accuracy of generating optical flow $f_{i \to j}$, we used the optical flow results to warp the second frame image $I_j$ to the first frame to obtain $I_{i'}$. Then we calculated the corresponding structural similarity masks:

$$M_{ssim} = 1 - \text{SSIM}(I_i, I_{i'}) \tag{12}$$
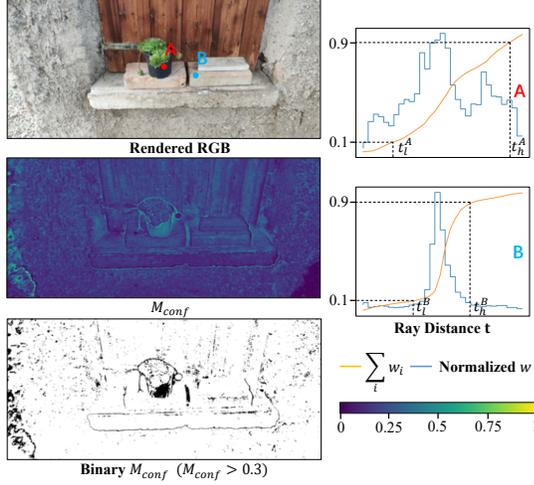
**Figure 5. Visualization of Radiation Field Confidence.** Left: RGB image rendered by neural field, $M_{conf}$ confidence mask, and binarized $M_{conf}$. Right: A represents the weight distribution of rays in dark areas without texture, and B represents the weight distribution of rays in rich texture areas. The $M_{conf}$ we proposed easily segmented those areas where the radiation field is difficult to reconstruct based on the ray weight distribution.

where SSIM[40] is the Structural Similarity Index Measure. $M_{ssim}$ can also be used as an evaluation indicator of optical flow. In this case, it needs to be used in conjunction with the $M_{occ}$ mask to eliminate artifacts caused by occlusion, as shown in Fig.4.

**Radiation field confidence:** In NeRF generative application scenarios, the quality of neural field reconstruction directly determines the method's overall performance. However, the only available AO indicators are not widely applicable for various methods and scenarios. In order to overcome the above difficulties, we propose a new metric in this section to measure the quality of neural field reconstruction, called radiation field confidence (RFC).

As shown in Fig.5B, when the neural field is well trained, the weight distribution of rays should be clustered at a central point, and the weight integration should appear as a step function, indicating that the radiation field has an exact surface on that ray. Moreover, we observed that such rays often correspond to areas with rich textures and appropriate brightness. On the contrary, it is difficult for the radiation field to learn the correct surface position for areas that are too dark or textureless.

Based on the above observations, we proposed $M_{conf}$ (RFC) to describe the reconstruction quality of the neural field, as follows:

$$M_{conf}(u,v) = \frac{t_h - t_l}{t_h + t_l} \qquad (13)$$

$$\sum_i^l w_i = th_{low}, \qquad \sum_i^h w_i = th_{high} \qquad (14)$$

Where $th_{low}$ and $th_{high}$ are the boundary thresholds, which are set to 0.1 and 0.9 in the experiments of this paper, $t_l$ and $t_h$ are the depths in the ray direction when the boundary threshold is reached.

$M_{conf}$ revealed the dispersion of weight distribution on a ray, with a range of $(0,1)$. The larger the value of $M_{conf}$, the lower the reconstruction quality of the radiation field. As shown in Fig.5, $M_{conf}$ can effectively identify the parts of where the radiation field is challenging to reconstruct.

**Geometry consistency:** Inspired by Bian[7], we calculated the geometric consistency mask of rendering depth under multiple perspectives. This is a core component for scene flow and depth estimation tasks. Although this paper focuses on optical flow applications, we hope the data factory proposed can be further applied to a broader range of downstream tasks.

Specifically, as shown in Fig.3, given the rendering depth $Z_i$ of the first frame and the pose $P_i, P_j$ between the two perspectives. We can calculate the coordinates $p_{i'}$ and depth map $Z_{i'}$ in $P_j$ corresponding to $P_i$ based on Eq.7, and then use NeRF rendering to obtain the depth value $Z_j^{i'}$ of the corresponding coordinates $p_{i'}$. The final geometric consistency is as follows:

$$M_{dc} = \frac{\left| Z_j^{i'} - Z_{i'} \right|}{Z_j^{i'} + Z_{i'}} \qquad (15)$$

Where the value range of $M_{dc}$ is $(0,1)$. The larger the value of $M_{dc}$, the lower the depth consistency.

### 3.3. NeRF-Supervised Training

After calculating the optical flow value $f_{i \to j}$ between two frames, we filter the optical flow results based on the mask $M_{conf}$, $M_{ssim}$, and $M_{dc}$. In this paper, we simply set a threshold to filter out those unqualified points, and the final optical flow label $f_{i \to j}^{gt}$ used for training is as follows:

$$f_{i \to j}^{gt} = M_{conf}^{th1} M_{ssim}^{th2} M_{dc}^{th3} f_{i \to j} \qquad (16)$$

$$M^{th} = \begin{cases} 1 & \text{if } M < th \\ 0 & \text{otherwise} \end{cases} \qquad (17)$$

Points with $f_{i \to j}^{gt}$ equal to zero do not participate in the loss calculation.

## 4. Experimental

In this section, we first introduce the implementation details of the dataset and experiment and then discuss the experimental results.

Table 1. **Ablation Study.** The best results in ablation studies are bolded. $Fl_{epe}$ is the average end-to-end optical flow error, $Fl_{all}$ is the optical flow outlier rate (errors greater than 3 pixels or greater than 5% are considered outliers)

| | | | | | | K15 | | Midd-A | | K12 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | Training Data | $M_{conf}$ | $M_{ssim}$ | $M_{dc}$ | $M_f$ | $Fl_{epe}$ | $F_{all}$ | $Fl_{epe}$ | $F_{all}$ | $Fl_{epe}$ | $F_{all}$ |
| RAFT[33] | ADF15 (ours) | | | | | 4.9 | 17.34 | 0.313 | 0.222 | 2.04 | 9.18 |
| | | | | | ✓ | 4.62 | 15.71 | 0.368 | 0.184 | 1.81 | 7.45 |
| | | | | ✓ | ✓ | 4.42 | 18.28 | **0.201** | 0.186 | 1.77 | 9.59 |
| | | | ✓ | | ✓ | 4.4 | 15.7 | 0.257 | **0.129** | 1.72 | 7.76 |
| | | ✓ | | | ✓ | 4.19 | 14.83 | 0.3 | 0.231 | 1.73 | 8.12 |
| | | ✓ | ✓ | ✓ | ✓ | **4.19** | **14.15** | 0.28 | 0.175 | **1.62** | **6.8** |
| | Sintel[9] | | | | | 8.5 | 18.31 | 0.23 | 0.22 | 2.38 | 8.01 |
| | Flyingthings3D[25] | | | | | 8.2 | 23.42 | 0.24 | 0.17 | 3.14 | 14.16 |
| | FlyingChairs2[25] | | | | | 10.87 | 36.86 | 0.46 | 0.79 | 4.2 | 26 |
| Scale-flow[20] | ADF58 (ours) | ✓ | ✓ | ✓ | ✓ | 3.88 | 13.36 | 0.218 | 0.122 | 1.59 | 6.97 |
| RAFT | ADF58 (ours) | ✓ | ✓ | ✓ | ✓ | 4.17 | 13.9 | 0.223 | 0.138 | 1.59 | 6.43 |

**Generate dataset** We have gathered and created a series of 300 scenes, featuring both indoor still life and outdoor courtyard scenes and ensured complete stillness. For each scenario, we train for 30,000 steps with a batch size of 10240 based on Zip-NeRF. Afterwards, we generate approximately 200 camera pose pairs per scene (maintaining orientation towards the center of the scene) to produce an optical flow dataset and corresponding masks. In the end, the training set consists of a total of 58800 image pairs (ADF58).

**Deep optical flow training:** We use RAFT[33] as the main architecture for ablation experiments and optical flow experiments evaluation, as it is the backbone network of most advanced methods. Hence, the evaluation on it has better representativeness. Of course, we also trained a stronger normalized scene flow baseline Scale-flow[20] to evaluate the effectiveness of ADF in more advanced methods. All methods are trained with a batch size of 6 and a crop size of 384×768, where we trained 200k iterations for RAFT and 300k iterations for Scale-flow. All training is done from scratch without using any synthesized dataset for pre-training.

**Datasets:** In the evaluation, we used 194 images from KITTI 2012 training set[12] (K12), 3 static scenes from Middlebury[3] (Midd-A), and 200 images from KITTI 2015 training set[27] (K15). Around 30k KITTI raw data (Kr) and 4k KITTI multi-frame (Km) images were used for training in other self-supervised methods[17, 24]. In addition, commonly used optical flow pre-trained synthetic datasets Sintel[9] (S), Flyingthings3D[25] (T), and Flyingchair2[25] (C2) were also included in the comparison. To facilitate the ablation experiment, we separated a subset of 15600 images (ADF15) from the complete ADF58.

## 4.1. Ablation Study

In this section, we first studied the effectiveness of various masks proposed in this article. Then, we compared the datasets ADF58 and ADF15 produced by our data factory with the commonly used synthetic dataset. Except for ADF58, all datasets only train 100K iterations.

**Use of masks:** We simply set a threshold to filter out unqualified points. Specifically, points with $M_{conf}$ greater than 0.3 will be excluded, those with $M_{ssim}$ greater than 0.1 will be excluded, and those with $M_{dc}$ greater than 0.01 will be excluded. In our experiments, the number of added motion foreground $M_f$ is 2.

**Mask analysis:** Tab.1 shows the experimental results of training RAFT using different masks. Firstly, we found that the training effect after using masks always improves in most cases. Specifically, after adding $M_f$ as a sports foreground, the performance in the KITTI dataset was significantly improved, but there was a decrease in Midd-A. This is easy to understand because $M_f$ enhances the network's perception of motion prospects, while Midd-A is all static scenes. Secondly, we found that $M_{conf}$ had the most significant performance improvement among all masks, which proves that our motivation to evaluate dataset quality from the perspective of reconstruction stability is effective. Finally, RAFT achieved optimal results after using all masks, which proves that the various masks we proposed are compatible. Although this article only proves its effectiveness by applying masks through simple thresholding, it is necessary to use masks more finely in subsequent work, such as calculating the credibility of points by combining masks in different tasks.

**Comparison with synthetic datasets:** For fairness, we compared ADF15 with the commonly used synthetic

datasets Sintel, Flyingthings3D and Flyingchairs2. All of them were trained using the same training parameters based on RAFT. Tab.1 shows the comparison results, and ADF15 is far ahead in almost all indicators, thanks to its data domain being closer to real world scenario. It has been proven that our NeRF data factory scheme can serve as a substitute and supplement to existing synthetic datasets for pre-training matching tasks.

**Number of scenes and stronger baseline:** We attempted to use more scenarios for training, and as expected, ADF58 improved all metrics compared to ADF15, representing a key advantage of our work: ADF can quickly expand the photos taken casually into the dataset and enhance the method's performance. In addition, we also tested a stronger normalized scene flow method, Scale-flow, and achieved better performance than RAFT, which proves that our method is also effective on a stronger baseline.

Table 2. **Evaluation of Normalized Scene Flow.** Above is the depth method that only uses monocular images, and below is the traditional method that uses stereo images.

| Method | Training data | $Fl_{epe}$ | $Mid_{error}$ | Time/s |
|---|---|---|---|---|
| Scale-flow | ADF58(ours) | **3.46** | **149.83** | 0.2 |
| Expansion[45] | Kr | 6.56 | 348 | 0.2 |
| SMMSF[13] | Km+Kr | 7.92 | 288.99 | **0.063** |
| OSF[28] | K15 | - | **115** | **300** |
| PRSM[37] | K15 | - | 124 | 3000 |

## 4.2. Normalized Scene Flow

This section compares the Scale-flow trained on ADF58 with other monocular self-supervised scene flow methods. For fairness, we use the same evaluation criteria as Yang[45]. Specifically, using 40 images in K15 for evaluation, the metrics include depth change rate $\tau$ and end-to-end optical flow error $Fl_{epe}$. The calculation of depth change rate loss $Mid_{error}$ is as follows:

$$Mid_{error} = ||log(\tau) - log(\tau_{GT})||_1 \cdot 10^4 \qquad (18)$$

Where $\tau = z_2/z_1$, $z_1$ is the depth of point in the first frame, and $z_2$ is the depth of the corresponding point in the second frame.

We first compared our method with the state-of-the-art monocular self-supervised scene flow method SMMSF[13]. Our Scale-flow with ADF58 outperformed them by a large margin (149.83 vs. 288.99). Because too complex models often have greater training difficulty, the architecture of general unsupervised methods tends to favour models with small parameter numbers, which is also why SMMSF is smaller and faster. Unlike them, our data factory solution can fully train more complex and more extensive models.

We also compare with traditional methods[28, 37] that use stereo images as input. They decompose the image into rigid blocks, and iteratively optimize the 3D motion of the rigid blocks based on the rigid assumption and regularization terms. As shown in Tab.2, although traditional methods use additional sensors, we have achieved accuracy close to theirs, and the speed is much faster (0.2s vs. 300s).

## 4.3. Self-supervised Optical Flow

In this section, we compare with several state-of-the-art self-supervised optical flow methods: MDFlow[17], UPFlow[24], and UFlow[15]. Because MDFlow only provides fast mode weights, the MDFlow in our paper defaults to MDFlow-fast.

Table 3. **Evaluation of Self-supervised Optical Flow.** Top is the zero-shot generalization result, and below is the result of fine-tuning using target domain data.

| Method | Training data | K15 $Fl_{epe}$ | $F_{all}$ | K12 $Fl_{epe}$ | $F_{all}$ |
|---|---|---|---|---|---|
| MDFlow[17] | Sintel | 10.05 | 23.12 | 3.49 | 12.17 |
| Scale-flow | ADF58(ours) | **3.88** | **13.36** | **1.59** | 6.97 |
| RAFT | ADF58(ours) | 4.17 | 13.9 | 1.59 | **6.43** |
| UFlow[15] | Km+Kr | 2.71 | - | 1.68 | - |
| MDFlow[17] | Km | 4.44 | **12.3** | 1.83 | **6.8** |
| UPFlow[24] | Km+Kr | **2.45** | - | **1.27** | - |
| SMMSF[13] | Km+Kr | 6.04 | 18.81 | - | - |

As shown in the upper part of Tab.3, our method has achieved absolute advantages over the self-supervised algorithm in the case of zero-shot generalization. Even after the self-supervised ways are fine-tuned in the target domain, the generalization performance of ADF is still competitive compared with these fine-tuned methods, as shown in the lower part of Tab.3.

It is not difficult to find that ADF performs weakly in the KITTI evaluation of driving scenarios compared to the stunning performance in daily scenarios. There are two main reasons for this: firstly, for safety reasons, we did not shoot road scenes and add in ADF58; secondly, due to the inherent defects of Zip-NeRF, it is impossible to reconstruct the reflective and transparent surfaces in cars (see more in **supplementary materials**).

## 4.4. Zero-shot Generalization in Real Word

In this section, we test our algorithm in real-world daily scenarios, which is also the original intention of this work: to enable the massive number of excellent optical flow algorithms and scene flow algorithms to be truly applied to our daily production and life.

**Evaluation method:** As there is currently no large-scale real-world optical flow dataset, we have adopted an indirect evaluation method to evaluate optical flow accuracy based on SSIM and photometric loss. Assuming $f$ is the optical flow between frames $I_i$ and $I_j$, in order to evaluate its accuracy, we project $I_j$ frames onto $I_i$ frames based on $f$ to obtain $I_i'$. The specific calculation method is as follows:

$$S_{loss} = M_{occ} \cdot M_{ssim} \tag{19}$$

$$P_{loss} = M_{occ} \cdot |I_i - I_{i'}| \tag{20}$$

Where $S_{loss}$ and $P_{loss}$ represent SSIM loss and photometric loss, respectively, while $M_{occ}$ represents the occlusion mask between two frames, used to eliminate artifacts caused by occlusion.

**Evaluation dataset:** We chose DAVIS[30] as the evaluation dataset because it contains many dynamic prospects and a wide distribution of data domains. Moreover, it has a foreground mask that can be used to calculate $M_{occ}$. Finally, approximately 3,400 image pairs were included in the evaluation.

Table 4. Evaluation of Zero-shot Generalization in Real-world. $fg$ and $bg$ represent the foreground and background.

| Method | Training data | $S_{loss}^{fg}$ | $S_{loss}^{bg}$ | $P_{loss}^{fg}$ | $P_{loss}^{bg}$ |
|---|---|---|---|---|---|
| Scale-flow | ADF58(ours) | **0.21** | **0.09** | **11.56** | **4.13** |
| RAFT | ADF58(ours) | 0.22 | 0.09 | 11.69 | 4.19 |
| GMFlow[44] | ADF58(ours) | 0.26 | 0.11 | 12.73 | 4.82 |
| Scale-flow | KITTI-ft | 0.26 | 0.10 | 13.62 | 4.55 |
| RAFT | KITTI-ft | 0.29 | 0.11 | 15.15 | 4.82 |
| GMFlow | KITTI-ft | 0.31 | 0.15 | 15.81 | 5.97 |

We mainly compare the performance differences under different supervision methods. As shown in Tab.4, KITTI-ft represents the commonly used fine-tuning training process, which involves pre-training on extensive synthetic data such as Sintel and Flyingthings3D, then fine-tuning on the real-world dataset KITTI. ADF58 represents our self-supervised data factory solution. RAFT, Scale-flow and GMFlow exhibits a significant difference in generalization ability under different training methods ( 0.22 vs. 0.29; 0.21 vs. 0.26; 0.26 vs. 0.31). This proves that our data factory scheme has better generalization performance in real-world scenarios compared to the previous supervised training mode.

We have visualized the comparison between ADF, supervised, and self-supervised methods in the **demo** folder of the **supplementary materials**. Welcome to check.

## 5. Limitation and Improvement

In this section, we discuss the limitations and potential improvement methods of our method.



**Image 1**　　**Image 2**　　**Scale-flow (ADF58)**　　**Scale-flow (KITTI-ft)**
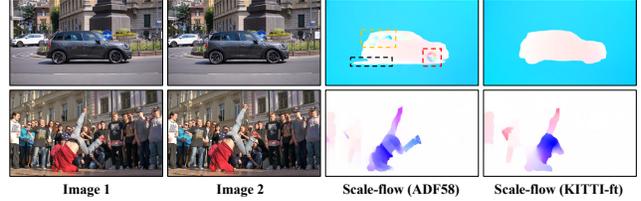
Figure 6. **Optical Flow and Object Flow:**. Observing the parts in coloured boxes, the methods trained by ADF are more faithful to the flow of light, while existing supervised methods (KITTI-ft) often tend to learn object flows with semantics. In addition, due to the lack of training samples, supervised methods often have poorer generalization in the real world.

**Optical flow and object flow:** Observing the car in Fig.6, we found that our method has more details compared to the supervised learning method. For example, the rotating wheels in the red box, neglected transparent glass in the orange box, and shadow in the black box. This is because our method is based on NeRF, so the results are more faithful to the light flow. In existing supervised learning[27], optical flow is defined as object flow, which tracks the surface of objects in space and introduces semantic information to erase some details, such as rotating car wheels and transparent windows. In future work, we believe that we can try to bring our data factory closer to supervised learning by introducing a big semantic model[19].

**NeRF:** The Zip-NeRF used in this paper cannot reconstruct reflective surfaces such as glass and water surfaces, and is limited to static scenes. These limitations directly prevent us from creating fine-tuning scenarios based on KITTI raw data. Fortunately, thanks to the rapid development of NeRF technology[2, 23, 36, 41, 46], this problem has been partially resolved.

## 6. Conclusion

We propose a groundbreaking optical flow training method: automated data factory (ADF), which utilizes scenes generated by NeRF to train deep optical flow networks without requiring manual annotation or expensive equipment. ADF can generate almost infinite high-quality optical flow datasets from ordinary monocular cameras, and directly use them for network training. This approach results in excellent real-world zero generalization performance, surpassing most self-supervised and supervised methods. Our work can help many excellent optical flow algorithms perform better in the real world, making them applicable to production and life!

# References

[1] Muhammad Haseeb Arshad, Muhammad Bilal, and Abdullah Gani. Human activity recognition: Review, taxonomy and open challenges. *Sensors*, 22(17):6463, 2022. 1

[2] Benjamin Attal, Eliot Laidlaw, Aaron Gokaslan, Changil Kim, Christian Richardt, James Tompkin, and Matthew O'Toole. Törf: Time-of-flight radiance fields for dynamic scene view synthesis. *Advances in neural information processing systems*, 34:26289–26301, 2021. 8

[3] 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)*, 92:1–31, 2011. 6

[4] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 5470–5479, 2022. 2

[5] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. *arXiv preprint arXiv:2304.06706*, 2023. 1, 2

[6] Steven S. Beauchemin and John L. Barron. The computation of optical flow. *ACM computing surveys (CSUR)*, 27(3):433–466, 1995. 1

[7] Jia-Wang Bian, Huangying Zhan, Naiyan Wang, Zhichao Li, Le Zhang, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth learning from video. *International Journal of Computer Vision (IJCV)*, 129 (9):2548–2564, 2021. 5

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

[9] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision (ECCV)*, pages 611–625, 2012. 1, 2, 6

[10] Linda Capito, Umit Ozguner, and Keith Redmill. Optical flow based visual potential field for autonomous driving. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 885–891. IEEE, 2020. 1

[11] Lijie Fan, Wenbing Huang, Chuang Gan, Stefano Ermon, Boqing Gong, and Junzhou Huang. End-to-end learning of motion representation for video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6016–6025, 2018. 1

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 3354–3361, 2012. 6

[13] Junhwa Hur and Stefan Roth. Self-supervised multi-frame monocular scene flow. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2684–2694, 2021. 7

[14] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9772–9781, 2021. 1

[15] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. In *European Conference on Computer Vision (ECCV)*, pages 557–572, 2020. 1, 7

[16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 1

[17] Lingtong Kong and Jie Yang. Mdflow: Unsupervised optical flow learning by reliable mutual knowledge distillation. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 33(2):677–688, 2022. 1, 6, 7

[18] Kanokphan Lertniphonphan, Supavadee Aramvith, and Thanarat H Chalidabhongse. Human action recognition using direction histograms of optical flow. In *International Symposium on Communications & Information Technologies (ISCIT)*, pages 574–579, 2011. 1

[19] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. Semantic-sam: Segment and recognize anything at any granularity. *arXiv preprint arXiv:2307.04767*, 2023. 8

[20] Han Ling, Quansen Sun, Zhenwen Ren, Yazhou Liu, Hongyuan Wang, and Zichen Wang. Scale-flow: Estimating 3d motion from video. In *Proceedings of the 30th ACM International Conference on Multimedia (ACMMM)*, pages 6530–6538, 2022. 6

[21] Han Ling, Yinghui Sun, Quansen Sun, and Zhenwen Ren. Learning optical expansion from scale matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 5445–5454, 2023. 2

[22] Haisong Liu, Tao Lu, Yihui Xu, Jia Liu, Wenjie Li, and Lijun Chen. Camliflow: Bidirectional camera-lidar fusion for joint optical flow and scene flow estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 5791–5801, 2022. 2

[23] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. *arXiv preprint arXiv:2308.09713*, 2023. 8

[24] Kunming Luo, Chuan Wang, Shuaicheng Liu, Haoqiang Fan, Jue Wang, and Jian Sun. Upflow: Upsampling pyramid for unsupervised optical flow learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 1045–1054, 2021. 6, 7

[25] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 4040–4048, 2016. 1, 2, 6

[26] Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and

stereo. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 4981–4991, 2023. 1

[27] Moritz Menze, Christian Heipke, and Andreas Geiger. Joint 3d estimation of vehicles and scene flow. In *ISPRS Workshop on Image Sequence Analysis (ISA)*, pages 427–434, 2015. 1, 2, 6, 8

[28] Moritz Menze, Christian Heipke, and Andreas Geiger. Object scene flow. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140:60–76, 2018. 7

[29] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2

[30] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 8

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

[32] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 8922–8931, 2021. 2

[33] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, pages 402–419, 2020. 1, 6

[34] Fabio Tosi, Alessio Tonioni, Daniele De Gregorio, and Matteo Poggi. Nerf-supervised deep stereo. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 855–866, 2023. 1, 3

[35] Prune Truong, Marie-Julie Rakotosaona, Fabian Manhardt, and Federico Tombari. Sparf: Neural radiance fields from sparse and noisy poses. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 4190–4200, 2023. 1, 3

[36] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 5481–5490, 2022. 8

[37] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3d scene flow estimation with a piecewise rigid scene model. *International Journal of Computer Vision*, 115:1–28, 2015. 7

[38] Hengli Wang, Peide Cai, Rui Fan, Yuxiang Sun, and Ming Liu. End-to-end interactive prediction and planning with optical flow distillation for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2229–2238, 2021. 1

[39] Hengli Wang, Peide Cai, Yuxiang Sun, Lujia Wang, and Ming Liu. Learning interpretable end-to-end vision-based motion planning for autonomous driving with optical flow distillation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 13731–13737. IEEE, 2021. 1

[40] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[41] Tianhao Wu, Fangcheng Zhong, Andrea Tagliasacchi, Forrester Cole, and Cengiz Oztireli. D^2nerf: Self-supervised decoupling of dynamic and static objects from a monocular video. *Advances in Neural Information Processing Systems*, 35:32653–32666, 2022. 8

[42] Gangwei Xu, Xianqi Wang, Xiaohuan Ding, and Xin Yang. Iterative geometry encoding volume for stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21919–21928, 2023. 2

[43] Gangwei Xu, Yun Wang, Junda Cheng, Jinhui Tang, and Xin Yang. Accurate and efficient stereo matching via attention concatenation volume. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[44] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 8121–8130, 2022. 1, 2, 8

[45] Gengshan Yang and Deva Ramanan. Upgrading optical flow to 3d scene flow through optical expansion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 1334–1343, 2020. 7

[46] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, and Yue Wang. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023. 3, 8

[47] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pages 1389–1399, 2023. 3

[48] Zhiqi Zhang, Pan Ji, Nitin Bansal, Changjiang Cai, Qingan Yan, Xiangyu Xu, and Yi Xu. Clip-flow: Contrastive learning by {s}emi-supervised iterative pseudo {l}abeling for optical flow estimation. *arXiv preprint arXiv:2210.14383*, 2022. 1