

Reviving Unsupervised Optical Flow: Concept Reevaluation, Multi-Scale Advances and Full Open-Source Release

Azin Jahedi¹ Marc Rivinius² Noah Berenguel Senn¹ Andrés Bruhn¹

¹ University of Stuttgart, VIS, Computer Vision Group ² University of Stuttgart, SEC

{Azin.Jahedi, Noah.Berenguel-Senn, Andres.Bruhn}@vis.uni-stuttgart.de
Marc.Rivinius@sec.uni-stuttgart.de

Abstract

Unsupervised optical flow methods have become more popular in the last decade, enabling the training of models across domains without ground truth data. Although RAFT and its successors have achieved significant success in the supervised settings, many unsupervised approaches continue to use older backbones such as PWC-Net. One reason for this architectural stagnation is that the current RAFT-based SOTA approach has proven challenging for the community to reproduce. In this paper, we revive and advance unsupervised optical flow: First, we introduce Sun-RAFT: a simple unsupervised RAFT. Second, building on Sun-RAFT, we present Muun-RAFT: a novel multi-scale unsupervised RAFT, where we propose a gradual context-based upsampling to refine the flow, further improving both accuracy and preservation of details. Third, we reexamine previously advised unsupervised strategies to identify effective training settings. In terms of results, both our methods demonstrate strong generalization capabilities and set a new SOTA for unsupervised two-frame approaches on MPI-Sintel, with Muun-RAFT surpassing even the current multi-frame SOTA by up to 28%. Finally, we open-source our PyTorch code, enabling further developments in the field: <https://cv-stuttgart.github.io/Reviving-Unsupervised-OpticalFlow>.

1. Introduction

Optical flow estimation is a fundamental task in computer vision, with applications such as object tracking [28], video processing [33], medical image registration [23] and autonomous driving [39]. In the past decade, supervised methods have gained enormous success; however, they rely on ground truth for training which is hard to obtain. Although accurate within their training domains, they often exhibit limited cross-domain generalization. In this context, while with extensive training on vast amounts of synthetic data they achieve great out-of-domain performance on other synthetic data [27], their accuracy on real-world data with a

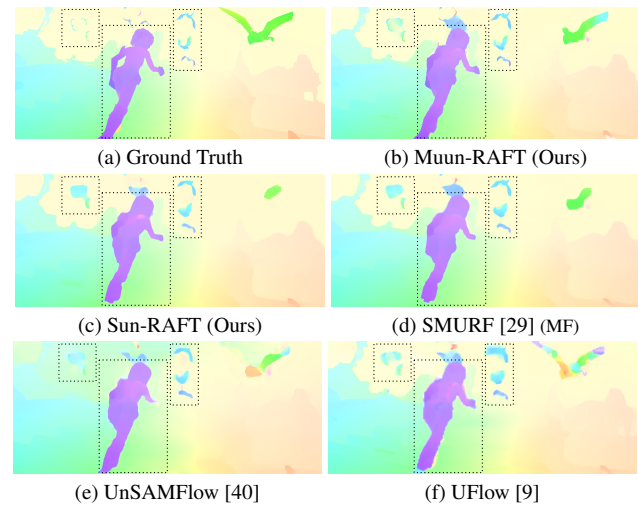


Figure 1. Visual comparison of our results vs. recent unsupervised methods from the literature on a sample of Sintel Clean (test).

comparable architecture remains moderate¹. One way to tackle domain-discrepancy is the tedious task of generating carefully tailored synthetic datasets for specific target domains in real-world applications [24, 25]. Another option is domain-adaptation, which has been recently considered for optical flow [37, 43]; however, those methods typically work with similar source and target domains and have yet to extend their adaptation to vastly different targets where suitable pre-training is lacking. Given these observations and considering the wide range of optical flow applications, unsupervised training for flow estimation remains a valuable research direction, as it offers flexibility to adapt to different domains without requiring ground truth data, while demonstrating good generalization capabilities [9, 29].

In the past decade, advances in supervised optical flow estimation have consistently triggered research on unsupervised methods by adopting the underlying architectures as effective matching backbones. For instance, shortly after the pioneering learning-based FlowNet approach [3]

¹RAFT (extensive supervised training [27]) vs. unsupervised SMURF (two-frame [29]): 2.71 vs. 2.45 (EPE), 9.16 vs. 7.53 (FL) on KITTI (train).

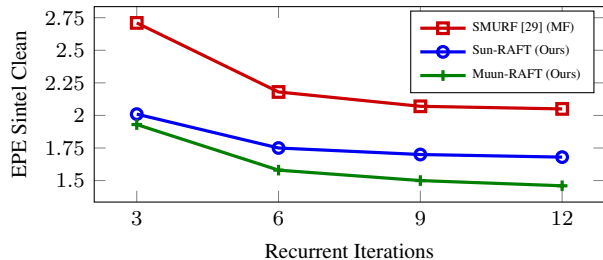


Figure 2. Accuracy comparison on Sintel Clean by increasing inference recurrent iterations between our two-frame Sun-RAFT and Muun-RAFT models and the multi-frame SMURF approach [29].

was introduced, first unsupervised methods based on the corresponding encoder-decoder architecture were proposed [8, 26, 38]. Similarly, inspired by PWC-Net [30], unsupervised methods started to use a cost-volume-based hierarchical estimation relying on feature warping. However, despite the paradigm shift to utilize iterative recurrent schemes in the supervised setting triggered by RAFT [34], recent works are still based on PWC-Net [31, 39, 40]. In fact, only a few unsupervised methods employed RAFT as their architectural backbone [17, 29, 41]. The most accurate among them is SMURF [29] which was presented in 2021, and is still SOTA in unsupervised optical flow estimation. Unfortunately, SMURF has proven challenging to reproduce. Besides, training with its current implementation demands substantial resources for both two-frame and multi-frame settings, further restricting its accessibility. These limitations hinder its use as a foundation for further progress in the field (see also supplementary material). In the absence of a simple, open source, and reproducible SOTA backbone, progress in the field has plateaued to a noticeable extent.

Contributions. In this paper we revive and advance unsupervised optical flow in five ways. **(i)** We reinvestigate RAFT as a backbone for unsupervised training and propose Sun-RAFT, as a simple yet effective unsupervised RAFT. **(ii)** Inspired by the success of multi-scale recurrent models [5–7], we also present Muun-RAFT: a multi-scale unsupervised RAFT. Taking MS-RAFT [5] as backbone, Muun-RAFT introduces a novel gradual context-based up-sampling scheme to refine the flow from all scales to full resolution, enabling the multi-scale architecture to achieve SOTA results in the unsupervised setting. **(iii)** To train both models effectively, we reexamine previously advised concepts for unsupervised training allowing us to present successful configurations for training our models. **(iv)** Both our two-frame Sun-RAFT and Muun-RAFT models set a new SOTA on MPI-Sintel, by Muun-RAFT surpassing the multi-frame SMURF approach on MPI-Sintel Clean by 28% (EPE of 1.44 vs. 1.99). As shown in Figure 2, both methods outperform SMURF with only 3 recurrent iterations. Furthermore, both models exhibit excellent cross-dataset generalization, with Muun-RAFT outperforming the original su-

pervised RAFT [34] on the Spring dataset at all standard training stages, including when RAFT is fine-tuned on the similar Sintel dataset. **(v)** Finally, unlike previous works, that are either not fully accessible or rely on more complex training schedules [9, 16, 29, 32, 39, 40], we fully release our PyTorch code along with a short training schedule; less than 100K iterations over all training stages, enabling the community to build upon our work.

2. Related Work

Early optical flow approaches go back to variational formulations [1, 4] incorporating data and smoothness constraints in a global optimization framework. However, with the rise of deep learning such methods were surpassed by both supervised and unsupervised learning-based approaches by a significant margin. Early unsupervised methods [21, 26, 38] leveraged architecture of FlowNet [3], the first supervised neural network for flow, while considering data and smoothness terms akin to those in variational methods for training. Afterwards, methods exploited PWC-Net’s more effective backbone architecture [30] and introduced teacher-student models utilizing data augmentation as self-supervision [11–13], an element that remains crucial in the unsupervised training of later methods [9, 16, 29], including ours. Other investigations sought to improve the estimate by extensive experimentation to identify impactful settings [9], introducing a better inter-scale upsampling [16], adding occluded super-pixels [13] or additional frames for self-supervision [8, 11, 13, 29, 31, 32], distilling self-supervision content [10], using content-aware teacher-student regularization [14], targeting brightness changes and low visibility conditions [15, 17], or introducing semantic information obtained by off-the-shelf segmentation methods [39, 40].

Comparing both our models to more recent unsupervised methods, we distinguish two classes of single-scale and multi-scale methods. Considering recent single-scale models, BrightFlow [17] employs RAFT and introduces brightness corrections for more robustness against illumination changes. MRDFlow [41] is based on a modified RAFT backbone, processing single-scale features by a hierarchical recurrent unit. SMURF [29], on the other hand, is the most accurate approach to employ RAFT as backbone, where it computes the photometric loss from warped full-images, applies a zoom-and-crop augmentation for self-supervision without consistency masking and integrates a multi-frame self-supervision based on ProFlow [18]. Instead, our SOTA Sun-RAFT method is much simpler. We employ RAFT and compute the photometric loss without the more complex full-image-warping technique. We use more diverse augmentations for self-supervision for a broader learning and utilize forward-backward masking for a more reliable self-supervision. We show the positive impact of each of these choices via extensive ablations. Also, we do not employ

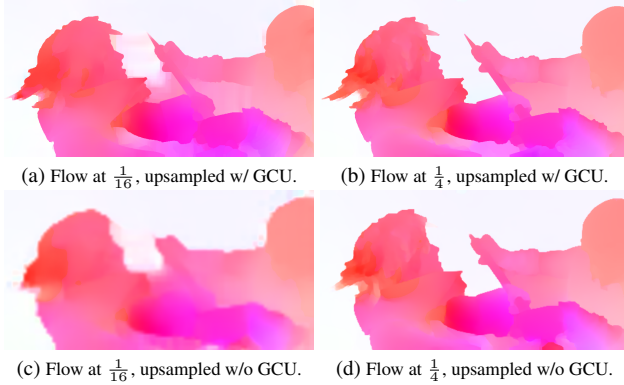


Figure 3. The impact of our gradual context-based upsampling.

any expensive multi-frame self-supervision. On the other hand, our multi-scale Muun-RAFT model is based on the MS-RAFT backbone [5] incorporating multi-scale features, and it introduces a novel gradual context-based upsampling, which refines the flow from all scales and iterations to *full-resolution* for computing all loss terms. Hence, it differs from recent multi-scale methods [14, 15, 31, 39, 40], that are based on ARFlow [11] utilizing a lightweight PWC-Net as backbone. Those methods also follow longer training schedules and do not utilize the estimate of all scales at full-resolution for computing the unsupervised objectives. Note that our gradual context-based upsampling differs from the sequential upsampling in MRDFlow [41], where the learned features are originally coarse. It is also different from [39], where a convex upsampling is applied only to the estimate of the last scale in a lightweight variant of PWC-Net.

3. Approach

Given two frames I_1, I_2 of size (h, w) of an image sequence, our goal is to compute the motion field between them, i.e. the optical flow f . In the following, we first explain model architectures of Sun-RAFT and Muun-RAFT. Afterwards, we elaborate on the unsupervised training procedure where we also discuss our effective self-supervision settings, as the result of our concept reevaluation.

3.1. Sun-RAFT

Our single-scale Sun-RAFT model adopts RAFT’s architecture [34]: Two images I_1, I_2 are passed to a feature encoder, the former is also processed by a context encoder and features at $(\frac{h}{8}, \frac{w}{8})$ are obtained. The matching costs and correlation pyramid are computed from image-features, and looked up based on an initial or current flow estimate. Then, context features, looked-up costs and the current flow estimate are further processed via a GRU, its hidden-state is updated and the flow increment is computed. Based on the updated hidden-state, also a convex $\times 8$ mask is learned to upsample the flow to full resolution. We refer to this type of

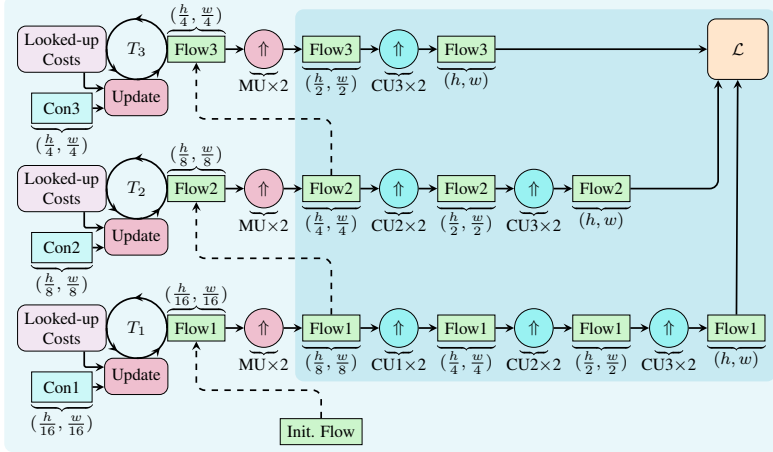
upsampling as a matching-based convex upsampling (MU), because the upsampling mask is learned from the updated hidden-state at each matching iteration and relies on matching costs between the two frames. Importantly, each intermediate estimate guides the cost lookup and therefore, not only the final flow, but also all intermediate flows are important for supervision. Hence, similar to [29, 34], Sun-RAFT considers all flow estimates in the unsupervised training.

3.2. Muun-RAFT

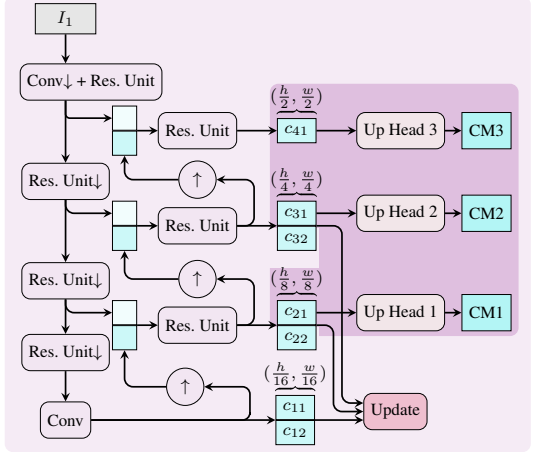
We also present Muun-RAFT, a context-enhanced multi-scale recurrent network based on MS-RAFT [5], where we introduce a novel gradual context-based convex upsampling strategy that upsamples the intermediate and final flows to full resolution for computing the unsupervised training losses. In this case, the flow is iteratively estimated in a coarse-to-fine manner with three scales (as in MS-RAFT [5]), from the coarsest scale at $\frac{1}{16}$ to the finest at $\frac{1}{4}$, where at each scale, the flow is estimated similar to RAFT [34] (see also supplementary material, Sec. 4).

Unlike the common practice in unsupervised multi-scale methods [9, 16], we consider intermediate *and* final results at *full resolution* to compute the unsupervised loss terms. Note that MS-RAFT [5] employs bilinear upsampling up to factor $\times 8$ to upsample the intermediate and final flow to full-resolution. Such a drastic bilinear upsampling, however, degrades the quality of the flow fields and is problematic because the losses computed from these flows steer the training. This particularly holds for the *unsupervised* case where no ground truth guides the matching at each scale. Instead, we propose a novel gradual context-based upsampling, where learned *context-based* $\times 2$ upsampling masks are utilized. This strategy *enables* the multi-scale scheme in [5] to achieve accurate results in the *unsupervised* setting, by preventing inaccuracies caused by (large-factor) bilinear interpolation leading to inappropriate back-propagation. Besides, context features, derived from I_1 , are a suitable source of information to guide the upsampling, as motion is typically aligned with the structure of I_1 ; the reference frame. Figure 3 compares intermediate and final outputs of Muun-RAFT with our gradual context-based upsampling (GCU) at the top, and without GCU as in [5] at the bottom, which we trained for comparison. While in both cases the flow improves from scale $\frac{1}{16}$ to $\frac{1}{4}$, the estimate with GCU is much more detailed at both scales. In Sec. 4.2 we present the corresponding ablation, where this improvement also becomes explicit in terms of the end-point-error. We now discuss how our novel gradual context-based upsampling is utilized in the recurrent multi-scale architecture.

Detailed Overview. The architecture of Muun-RAFT, with our novel parts (compared to its backbone) highlighted, are illustrated in Fig. 4. First, image features and context features (*Con* in Fig. 4a) are pre-computed at scales $[\frac{1}{16}, \frac{1}{8}, \frac{1}{4}]$.



(a) Coarse-to-fine flow computation via gradual context-based convex upsampling.



(b) Computation of context-based masks.

Figure 4. Architecture of Muun-RAFT. Note that in (b), (c_{11}, c_{12}) , (c_{21}, c_{22}) , (c_{31}, c_{32}) represent *Con1*, *Con2* and *Con3* in (a), respectively, and are directly used in the update block for flow estimation. Whereas c_{21} , c_{31} and c_{41} are passed to learned blocks to compute the upsampling masks for gradual upsampling from $\frac{1}{8}$ to $\frac{1}{4}$ (via CM1), $\frac{1}{4}$ to $\frac{1}{2}$ (via CM2) and $\frac{1}{2}$ to full-resolution (via CM3).

Image features (not shown in Fig. 4a for compactness) are utilized for computing the matching costs that are looked up (*Looked-up Costs* in Fig. 4a) based on an initial or current flow estimate at each iteration. Moreover, the context-based upsampling masks (CM) at different scales are also pre-computed from context features (see Fig. 4b). The flow is initialized with zero at the coarsest scale at $\frac{1}{16}$. After T_1 recurrent steps, this flow is upsampled with MU and employed to initialize the next finer scale at $\frac{1}{8}$. In addition, for computing the unsupervised losses, after *each* of the T_1 recurrent steps the flow is upsampled using MU to $\frac{1}{8}$ and then gradually upsampled via the context-based upsampling operator CU1 (from $\frac{1}{8}$ to $\frac{1}{4}$) using CM1, via CU2 (from $\frac{1}{4}$ to $\frac{1}{2}$) using CM2, and via CU3 (from $\frac{1}{2}$ to full resolution) using CM3, resulting in T_1 intermediate flows at full resolution. This procedure repeats for the other scales until the final flow at $\frac{1}{4}$ is estimated and upsampled to full-resolution. Besides, all $T_1 + T_2 + T_3$ (intermediate and final) flows are then used for computing the losses discussed in the next section. Note that, unlike matching-based convex upsampling (MU, discussed in Sec. 3.1) used in inter-scale initializations, context-based convex upsampling (CU) does not depend on the matching process at any scale. Therefore, it is suitable to be utilized for upsampling at any scale. Notably, the sequential use of $\times 2$ MUs is not feasible due to its matching-cost dependency (see supplementary material).

3.3. Unsupervised Training

Having discussed both architectures of Sun-RAFT and Muun-RAFT with a focus on how the intermediate and final flows are computed at full resolution, we now elaborate on the unsupervised objectives for training the models as well as our settings for a more effective training, as the result of our concept reevaluation.

Photometric Loss. We compute the soft Hamming distance from patches of the census-transformed image I_1 and its reconstruction by warping I_2 by the forward flow f_{12} , similar to [21]. Evidently, as there are no correspondences in forward-backward inconsistent regions, we mask out the occluded parts for computing the photometric loss. Similar to [9], we estimate the occlusions using a range-map of backward flow [36] with gradient stopping for training on Chairs [3] and Sintel [2] and forward-backward checking based on [1] when we train on KITTI 2015 [22]. The photometric loss for the forward flow estimate is given by

$$\mathcal{L}_{ph, fw} = \frac{1}{n} \sum O_{12} \odot \rho(I_1, \tilde{I}_2^{f_{12}}), \quad (1)$$

where O_{12} refers to the forward occlusion mask with zeroes at occlusions, $\frac{1}{n}$ computes the average over all pixels, \odot is the element-wise multiplication, ρ refers to sub-linear penalization [12] of the soft Hamming distance between input census-transformed patches, and $\tilde{I}_2^{f_{12}}$ is I_2 warped by f_{12} . Note that, unlike [29], we do not perform full-image-warping for computing the photometric loss, as experiments showed that it did not lead to improvements, in our case.

Smoothness Loss. We consider the edge-aware l -th order (o_l) smoothness terms in the spirit of [35] similar to [9, 29]:

$$\begin{aligned} \mathcal{L}_{sm_{o_l}, fw} = & \frac{1}{n} \sum \exp\left(-\frac{\alpha}{3} \sum_{c=1}^3 \left| \frac{\partial I_{1,c}}{\partial x} \right| \odot \left| \frac{\partial^l f_{12}}{\partial x^l} \right| \right) \\ & + \exp\left(-\frac{\alpha}{3} \sum_{c=1}^3 \left| \frac{\partial I_{1,c}}{\partial y} \right| \odot \left| \frac{\partial^l f_{12}}{\partial y^l} \right| \right), \end{aligned} \quad (2)$$

where α denotes the edge-sensitivity which we set to 150 [9, 29]. Here, the l -th derivative of the flow is down-weighted at locations with large RGB derivatives to avoid penalization of flow changes across edges of I_1 .



Figure 5. Use of augmentations in training. g_1, g_2 and a_1, a_2 denote geometric and appearance (photometric, occlusion) augmentation functions applied to images, respectively. g_2^* refers to the equivalent of g_2 applied to the flow to obtain the teacher flow.

Self-supervision Loss. The objective which is responsible for self-supervision is defined as follows:

$$\mathcal{L}_{self, fw} = \frac{1}{n} \sum \|\mathcal{S}(f_{12}^T) - f_{12}^S\| \odot M_{12}^T \odot (1 - M_{12}^S), \quad (3)$$

where f^T and f^S indicate the teacher and student flow, respectively, M refers to the forward-backward consistency mask for each flow and \mathcal{S} denotes gradient stopping. The student flow is computed from a set of geometric and photometric transformed images, and the teacher flow is obtained by applying the same geometric transformation to the flow computed from clean images [11] (without the geometric and photometric augmentation applied to the input of the student model).

Augmentations for Unsupervised Learning. We follow the augmentation settings for the *photometric and smoothness loss* of [29]. For computing the flow during training, we apply photometric, occlusion and geometric augmentations to the input images, while the actual images used in the loss were only geometrically augmented. This enables the network to learn to ignore occlusions and photometric changes; see Figure 5 (top).

Importantly, in the case of the self-supervision loss, unlike the settings in [9, 12, 29], we do not restrict ourselves to a fixed 64-pixel zoom and crop transformation as the only augmentation between the teacher and the student. Instead, we apply *more diverse geometric augmentations* such as random resize, stretch in independent directions, flipping, cropping and on top of that further photometric and occlusion augmentation *between the student and the teacher model*, inspired by [11] (see Fig. 5 bottom). Our ablations in Sec. 4.2 reveal the importance of our augmentation procedure compared to the zoom and crop setting, alone.

Forward-Backward Masking. We apply self-supervision only in areas where the teacher is not occluded but the student is, inspired by [12]. In this way, the student model is not only supervised to handle occlusions via the dedicated zoom-and-crop operation but also to deal with diverse geometric and photometric augmentations in those areas. The

reexamination of performing forward-backward masking in our ablations shows its clear positive impact which is in contrast with the findings in [29].

Overall Loss. During training, beside the mentioned loss terms in the forward direction, they are also computed in the backward direction. We denote the sum of the forward and backward terms by $\mathcal{L}_{ph}, \mathcal{L}_{sm}, \mathcal{L}_{self}$. Thereby, our employed multi-scale multi-iteration loss is computed via

$$\mathcal{L} = \sum_{s=1}^{N_{scales}} \sum_{i=1}^{T_s} \sum_{t \in \{ph, sm, self\}} \omega_t \gamma_t^{s,i} \mathcal{L}_t^{s,i}, \quad (4)$$

with $\gamma_t^{s,i} = \gamma_t^{T_{tot} - T_{s,i}}$, where T_{tot} is the total number of iterations over all scales, $T_{s,i} = \sum_{k=1}^{s-1} T_k + i$ denotes the number of iterations performed so far and ω stands for the weight for each term. We set $\gamma_{ph} = \gamma_{sm} = 0.8$ and $\gamma_{self} = 0.95$ for both models and all training stages, except for Sun-RAFT on KITTI, where $\gamma_{self} = 0.7$ yielded better results. Note that in the case of Sun-RAFT, the model has only one scale (see Section 3.1) with 12 recurrent iterations and for Muun-RAFT there are 3 scales with 4 iterations at each scale (see also Sec. 3 in the supplementary material).

4. Evaluation

Training Evaluation Strategy. We trained both Sun-RAFT and Muun-RAFT models for 75K iterations on the Chairs [3] dataset followed by fine-tuning on either MPI-Sintel [2] for 20K or the multi-view extension of KITTI [22] for 15K iterations². Our experiments indicated that neither longer training nor training on Things [19] yielded further improvements. Thereby, our overall training spans less than 100K iterations. For evaluation, we adopt the same strategy as DDFlow [42], which was also followed by UFlow [9] and SMURF [29]. Therefore for our experiments, we *trained the model on the test set of Sintel / KITTI (multi-view extension) and validated the results on the training set*. For Chairs, we trained the models on the *training* split and validated them on the *validation* split. Furthermore, to validate our models on the MPI-Sintel and KITTI *benchmarks*, we trained the models on the *training sets*. Hence, we did not use any data for validation that have been used for training.

For training our models, we use the same learning-rate schedule, optimizer and initialization as RAFT [34]³. The whole training process (pre-training and fine-tuning) took less than two days on two Nvidia A100 GPUs⁴ (each 40GB VRAM) for our Sun-RAFT model and less than three days

²Note that our observation is different from SMURF [29], where the network was tuned on Sintel for 15K iterations to avoid overfitting but benefited from a longer 75K-iteration training on KITTI.

³Initial experiments showed that these settings are more effective than the configuration used in [29].

⁴Note that our hardware usage is moderate, compared to [29] which uses 8 GPUs for its two-frame training (see supplementary material).

Method	Chairs Checkpoint			Sintel Checkpoint			KITTI Checkpoint		
	Spring EPE	Spring FL	Spring 1px	Spring EPE	Spring FL	Spring 1px	Spring EPE	Spring FL	Spring 1px
RAFT [34] (Supervised)	1.38	3.07	9.123	<u>0.63</u>	1.52	4.474	4.52	4.20	9.149
SMURF [29] (MF)	0.93	<u>2.25</u>	<u>5.464</u>	0.67	2.06	4.931	3.50	3.70	6.804
Sun-RAFT (Ours)	<u>0.89</u>	2.31	5.783	0.75	1.80	4.582	<u>2.31</u>	<u>3.23</u>	<u>6.457</u>
Muun-RAFT (Ours)	0.58	2.16	5.422	0.49	<u>1.72</u>	<u>4.553</u>	1.44	2.65	6.022

Table 1. Cross-data generalization on the Spring dataset. The best results are highlighted in bold, while the second-best are underlined.

Method	Trained on	Chairs Val	Sintel train		KITTI train	
			Clean	Final	EPE	FL
DDFlow	Chairs train	2.97	4.83	4.85	17.26	–
UFlow		2.55	3.43	4.17	15.68	32.69
SMURF-EAS (2F)		1.72	2.19	3.35	7.94	26.51
SMURF (2F)		1.99	2.48	<u>3.32</u>	12.71	31.04
Sun-RAFT		<u>1.90</u>	<u>2.22</u>	3.52	<u>9.64</u>	<u>28.71</u>
Muun-RAFT		1.71	2.02	3.08	8.61	22.06
DDFlow	Sintel test	3.46	{2.92}	{3.98}	12.69	–
UFlow		3.25	3.01	4.09	7.67	17.41
SMURF-EAS (MF)		1.99	1.99	2.80	4.47	12.55
SMURF-EAS (2F)		–	2.15	2.99	–	–
Sun-RAFT		2.23	<u>1.69</u>	<u>2.60</u>	<u>4.76</u>	<u>12.63</u>
Muun-RAFT		2.03	1.44	2.56	4.39	12.35
DDFlow	KITTI test	6.35	6.20	7.08	{5.72}	–
UFlow		5.05	6.34	7.01	2.84	9.39
SMURF-EAS (MF)		3.26	3.38	4.47	2.01	6.72
SMURF-EAS (2F)		–	–	–	2.45	7.53
Sun-RAFT		<u>3.49</u>	<u>3.97</u>	<u>5.20</u>	2.17	7.66
Muun-RAFT		3.32	3.35	4.89	2.34	8.64

Table 2. In-domain accuracy and cross-data generalization. In-domain results are highlighted. Results with EAS (evaluation with arbitrary scaling), are marked (not comparable to the results obtained by standard evaluation).

on three Nvidia A100 GPUs for our Muun-RAFT model. We thereby used a batch size of 8 for pre-training the Sun-RAFT method on Chairs; in all other cases, we used a batch size of 6. We considered the same patch sizes as [29] for training Sun-RAFT, while we used slightly smaller patch sizes (352, 448) for Muun-RAFT for training on Chairs and Sintel to save memory and to speed up training. We set $\omega_{ph} = 1$ in all training stages and both models, $\omega_{sm} = [2, 5, 0.5]$ and $\omega_{self} = [1.2, 1.2, 0.9]$ for Sun-RAFT Chairs, Sintel and KITTI training, respectively. The settings for Muun-RAFT are $\omega_{sm} = [2, 8, 2]$ and $\omega_{self} = [1.2, 1.2, 0.3]$. For both models, we employed the first order smoothness term for Chairs and Sintel and second order for training on KITTI. Importantly, unlike SMURF [29], we do not perform any arbitrary scaling during inference at any stage and infer the flow at the original input size (*cf.* [29], Section 4). Similar to RAFT [34], we use warm-start strategy for computing the flow on Sintel. Note, however, that our method does not exploit temporal information during training and hence is fundamentally different from multi-frame methods such as [29, 32]. Sun-RAFT and Muun-RAFT take 0.13 and 0.17 seconds using 12 iterations for both models to infer Sintel-sized images with the warm-

start setting using an Nvidia A100 GPU. For further details on the models, see the supplementary material.

4.1. Results

Cross-Data Generalization on Spring. We evaluate our models’ cross-dataset generalization on the training split of the Spring dataset [20] across various training stages in Table 1. Specifically, we compare the performance of Sun-RAFT and Muun-RAFT against multi-frame SMURF and RAFT [34] (using official checkpoints), at different training stages: Chairs, Sintel, and KITTI. Remarkably, Muun-RAFT surpasses supervised RAFT⁵ at all training stages in terms of average end-point error (EPE), even when RAFT has been fine-tuned on the similar Sintel domain. Our Sun-RAFT model also shows strong performance across training stages, highlighting the models’ excellent generalization capabilities. Moreover, although the image- and motion-domain discrepancy between KITTI and Spring yields a clear degradation of accuracy across methods, both our models perform favorably by a large margin.

Chairs, Sintel and KITTI. Table 2 demonstrates the in-domain accuracy and cross-data generalization on Chairs *validation split*, Sintel *train* and KITTI *train*, while the models are trained on Chairs *training split*, Sintel *test* and the multi-view extension of KITTI *test*, respectively, such that *no evaluation samples were used for training*. We compare Sun-RAFT and Muun-RAFT to DDFlow [12], UFlow [9] and SMURF [29] as the few methods following the same strict validation strategy. Note that the Chairs SMURF model does not exploit multi-frame self-supervision and EAS stands for evaluation with arbitrary scaling performed by SMURF. Using the published code, we evaluated the official Chairs checkpoint of SMURF with the original-resolution images and reported the results, denoted by SMURF (2F). In the case of Sintel and KITTI, as the two-frame checkpoints were not available, we reported the available results from their ablations (*cf.* [29], Table 5). In all training stages, both our Sun-RAFT and Muun-RAFT models perform favorably. Considering the model trained on *Chairs*, Muun-RAFT’s performance on Sintel Clean is on par with the *multi-frame SMURF model after fine-tuning on Sintel*. In this case, the excellent generalization performance

⁵Note that we do not claim achieving better out-of-domain results than supervised methods in general, rather we compare to RAFT [34] as a reference with comparable training checkpoints.

Method	Sintel Clean		Sintel Final		KITTI 2015			
	EPE		EPE		EPE	FL		
	train	test	train	test	train	train	test	
Multi-Frame	Back2Future-UFO (MF) [8] ECCV2018	{3.89}	7.23	{5.52}	8.81	[6.59]	–	22.94
	SelfFlow (MF) [13] CVPR2019	[2.88]	[6.56]	{3.87}	{6.57}	[3.84]	–	14.19
	ARFlow (MF) [11] CVPR2020	{2.73}	{4.49}	{3.69}	{5.67}	[2.85]	–	†
	CARFlow (MF) [14] Sensors2023	{2.25}	3.46	{3.23}	4.95	{2.11}	–	†
	M2Flow (MF) [31] AAAI2025	[2.01]	[3.38]	[3.12]	[5.01]	[1.95]	–	[7.37]
	SMURF (MF) [29] CVPR2021	{1.71}	3.15	{2.58}	4.18	[2.00]	[6.42]	6.83
Two-Frame	DDFlow [12] AAAI2019	{2.92}	6.18	{3.98}	7.40	[5.72]	–	14.29
	UFlow [9] ECCV2020	{2.50}	5.21	{3.39}	6.50	{2.71}	{9.05}	11.13
	BrightFlow (RAFT) [17] WACV2023	3.25	–	3.33	–	2.88	<u>7.98</u>	–
	CARFlow [14] Sensors2023	{2.36}	3.69	{3.28}	5.21	{2.34}	–	9.09
	UPFlow [16] CVPR2019	{2.33}	4.68	{2.67}	5.32	[2.45]	–	†
	MRDFlow [41] CSVT2022	{2.30}	3.76	{3.15}	5.26	[2.39]	[8.07]	†
	MDFlow [10] CSVT2022	{2.17}	4.16	{3.14}	5.46	[2.45]	–	<u>8.91</u>
	UnSAMFlow [40] CVPR2024	[2.21]	3.93	[3.07]	5.20	[2.01]	–	†
	Sun-RAFT (Ours, test)	<u>1.69</u>	–	<u>2.60</u>	–	2.17	7.66	–
	Sun-RAFT (Ours, train)	{1.68}	<u>2.94</u>	{2.52}	<u>4.23</u>	[2.22]	[7.53]	7.99
	Muun-RAFT (Ours, test)	1.44	–	2.56	–	<u>2.34</u>	8.64	–
	Muun-RAFT (Ours, train)	{1.41}	2.50	{2.45}	3.91	[2.37]	[8.43]	9.33

MF: Multi-frame {·}: Trained on unlabeled evaluation set [·]: Trained on related data e.g., a superset

†: Evaluation results not present on the benchmark

The best / second best results excluding ones in brackets are bold-faced / underlined.

Table 3. Results on the MPI-Sintel and KITTI 2015 benchmarks.

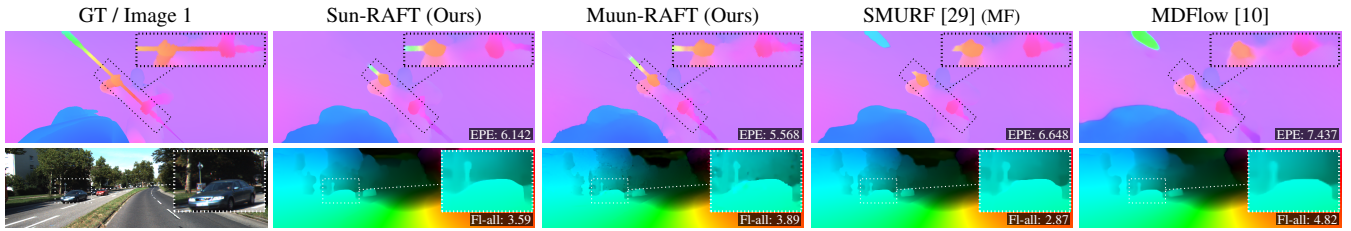


Figure 6. Visual comparison to other unsupervised methods on Sintel Final (top) and KITTI (bottom). Best viewed as PDF by zooming.

on KITTI is also noticeable. Considering the models trained on Sintel-test, both our variants outperform both two-frame and multi-frame SMURF by a large margin. In this case, Muun-RAFT outperforms both two-frame and multi-frame SMURF on the clean pass by 33% and 28% respectively, while it shows worthwhile generalization results on Chairs and KITTI. Finally, both methods yield strong KITTI results in terms of EPE, though Muun-RAFT is less accurate in the FL (percentage of outlier pixels) metric, despite maintaining good generalization on Chairs and Sintel.

Benchmark Results. The previous findings are confirmed by the results of our models on the official Sintel and KITTI benchmarks shown in Table 3. Sun-RAFT outperforms the best results on Sintel by CARFlow [14] by 20% and 19% for the clean and final pass, respectively. The improvements for Muun-RAFT are 32% and 25%, respectively. Sun-RAFT also performs well on the KITTI benchmark, ranking first among all two-frame results available on the benchmark. In this case, Muun-RAFT achieves fair results.

Visual Quality. We show examples of the estimated flows on the MPI-Sintel and KITTI benchmarks in Figure 6, where our models yield good results (see boxes).

4.2. Ablations

We perform two sets of ablations using the same training settings as in Table 2.

Sun-RAFT: Augmentations, Masking, and Warping. In our first set, which is only performed on Sun-RAFT, we investigate applying more diverse geometric, photometric and occlusion augmentations between the teacher and the student model, followed by assessing forward-backward masking during self-supervision and the impact of full-image-warping, introduced by SMURF, for Sintel and KITTI; see Table 4. Using the best settings, we then investigate architectural and training settings for Muun-RAFT in Table 5.

Considering the self-supervision settings, Table 4 shows that not including the more diverse geometric augmentations between teacher and student models, where we instead used the same zoom-and-crop settings of [9, 12, 29], yields

Method	Chairs Val	Sintel train		KITTI train	
		Clean	Final	EPE	FL
Sun-RAFT (Ours)	1.90	1.69	2.60	2.17	7.66
Sun-RAFT w/o DGA	2.46	1.90	2.94	3.20	9.03
Sun-RAFT w/o Ph (A_2)	2.07	1.74	2.64	2.24	7.74
Sun-RAFT w/o Occ (A_2)	1.92	1.80	2.66	2.38	7.91
Sun-RAFT w/o FBM	2.01	1.71	2.64	2.71	8.49
Sun-RAFT w/o FBM-ST	1.95	1.71	2.61	2.40	7.92
Sun-RAFT w/ FIW	–	1.87	2.73	2.41	7.79

Table 4. Ablating self-supervision settings and full-image-warping on Sun-RAFT. DGA: Diverse Geometric Augmentation. FBM: Forward-Backward Masking, FIW: Full-Image-Warping.

Method	Chairs Val	Sintel train		KITTI train	
		Clean	Final	EPE	FL
Supervision on Scale					
Last Native Scale	<i>div</i>	<i>div</i>	<i>div</i>	<i>div</i>	<i>div</i>
All Native Scales	5.91	5.84	6.19	<i>div</i>	<i>div</i>
All Full-Resolution (Ours)	1.71	1.44	2.56	2.34	8.64
SM Native Scales	1.79	1.54	2.72	2.39	8.62
Architecture: Upsampling Strategy					
MU $\times 2$ - Bilin. [5]	2.03	1.64	2.67	2.96	9.68
MU $\times 2$ - CU $\times 2$ (Ours)	1.71	1.44	2.56	2.34	8.64
MU $\times 2$ - CU $\times 2$ - Last Scale	1.70	1.52	2.64	3.14	9.82
Architecture: Context Network					
Individual Units [5]	1.89	1.53	2.65	2.68	8.86
Shared Units (Ours)	1.71	1.44	2.56	2.34	8.64

Table 5. Ablating on supervision scales and architectural details of Muun-RAFT.

degradation of the results in all cases. Also discarding the occlusion and photometric augmentations (A_2 in Figure 5) in this case worsens the results. In addition, not applying forward-backward masking during self-supervision (w/o FBM) clearly reduces the accuracy. Besides, applying the self-supervision in all areas where the teacher is forward-backward consistent (and not where the student is also forward-backward inconsistent: w/o FBM-ST) has a negative impact on KITTI. Finally, we assessed the impact of applying warping to full-images to compute the photometric loss. Our experiment shows that, in contrast to the findings in [29], this strategy does not improve the results using our training settings.

Muun-RAFT: Supervision and Architecture. In the next set of ablations which are performed upon Muun-RAFT (shown in Tab. 5), we first investigate the use of native-scale flow estimates without upsampling. Thereby, we assess the case where the photometric and smoothness terms were applied to the last scale (Last Native Scale). This experiment did not converge, highlighting the importance of

supervising intermediate outcome of the network. Taking the native-scale flows (at each scale of $\frac{1}{16}$, $\frac{1}{8}$ and $\frac{1}{4}$) and applying the loss terms at all scales accordingly (All Native Scales), although more stable, yielded poor results on Chairs and Sintel and did not converge for KITTI. Applying the losses on full-resolution intermediate and final flows based on our gradual context-based upsampling yielded the best results. We also investigated considering the smoothness loss on native-scales (SM Native Scales) similar to the suggestion in [9]. This strategy yielded on-par results on KITTI but was slightly worse on Chairs and Sintel.

Next, we show that using our gradual context-based upsampling (CU) is crucial for accuracy gains of the multi-scale architecture: the results of the unsupervised MS-RAFT backbone (MU $\times 2$ and bilinear interpolation, row 5; see also Fig. 3, bottom row) are inferior to the single-scale Sun-RAFT model. Interestingly, applying the CU mask appears to be important for upsampling both intermediate as well as the final flow for computing the loss. Applying a CU only for the final flow (Last Scale) yields on par results for Chairs but leads to inferior performance on Sintel and especially on KITTI. Utilizing context-based upsampling also on coarser scales introduces additional supervision signals for learning the context-based mask (Up Head in Fig. 4b), which is also used at the finest scale, where the final flow is computed. The next ablation shows that sharing the residual units of the context encoder (Fig. 4b) across scales is beneficial for all cases, while at the same time yields a more compact architecture. For more architectural ablations, please see the supplementary material.

5. Conclusion

Motivated by the widely experienced difficulty in training and reproducing SOTA unsupervised optical flow methods, we reinvestigated RAFT as a backbone to obtain a simple, yet effective unsupervised method, yielding Sun-RAFT. We furthermore presented Muun-RAFT, a multi-scale unsupervised RAFT, in which we propose a novel gradual context-based upsampling to obtain more accurate estimates at full-resolution. Our extensive concept reinvestigation enables an effective training with a short schedule. Both our open-source methods set a new SOTA for two-frame unsupervised optical flow on MPI-Sintel, and we hope they serve as a stepping stone for further advances in the field.

Acknowledgements. We thank the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - Project-ID 533085500 for its support. We also thank the DFG – Project-ID 251654672 – TRR 161 (B04) for funding Noah Berenguel Senn. Finally, we thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Azin Jahedi.

References

- [1] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proc. European Conference on Computer Vision (ECCV)*, pages 25–36, 2004.
- [2] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *Proc. European Conference on Computer Vision (ECCV)*, pages 611–625, 2012.
- [3] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2758–27, 2015.
- [4] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence (AI)*, 17(1-3):185–203, 1981.
- [5] Azin Jahedi, Lukas Mehl, Marc Rivinius, and Andrés Bruhn. Multi-scale raft: Combining hierarchical concepts for learning-based optical flow estimation. In *Proc. IEEE International Conference on Image Processing (ICIP)*, pages 1236–1240, 2022.
- [6] Azin Jahedi, Maximilian Luz, Marc Rivinius, and Andrés Bruhn. CCMR: High resolution optical flow estimation via coarse-to-fine context-guided motion reasoning. In *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6899–6908, 2024.
- [7] Azin Jahedi, Maximilian Luz, Marc Rivinius, Lukas Mehl, and Andrés Bruhn. MS-RAFT+: High resolution multi-scale RAFT. *International Journal of Computer Vision (IJCV)*, 132(5):1835–1856, 2024.
- [8] Joel Janai, Fatma Guney, Anurag Ranjan, Michael Black, and Andreas Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *Proc. European Conference on Computer Vision (ECCV)*, pages 690–706, 2018.
- [9] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. In *Proc. European Conference on Computer Vision (ECCV)*, pages 557–572, 2020.
- [10] 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.
- [11] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyu Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6488–6497, 2020.
- [12] Pengpeng Liu, Irwin King, Michael R. Lyu, and Jia Xu. DDFlow: learning optical flow with unlabeled data distillation. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, pages 8770–8777, 2019.
- [13] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. SelfFlow: self-supervised learning of optical flow. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4571–4580, 2019.
- [14] Libo Long and Jochen Lang. Regularization for unsupervised learning of optical flow. *Sensors*, 23(8):4080:1–4080:18, 2023.
- [15] Libo Long, Tianran Liu, Robert Laganière, and Jochen Lang. Robust unsupervised optical flow under low-visibility conditions. In *Proc. European Conference on Computer Vision Workshops (ECCVW)*, 2024.
- [16] Kunming Luo, Chuan Wang, Shuaicheng Liu, Haoqiang Fan, Jue Wang, and Jian Sun. UPFlow: upsampling pyramid for unsupervised optical flow learning. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1045–1054, 2021.
- [17] Rémi Marsal, Florian Chabot, Angélique Loesch, and Hichem Sahbi. BrightFlow: brightness-change-aware unsupervised learning of optical flow. In *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2060–2069, 2023.
- [18] Daniel Maurer and Andrés Bruhn. ProFlow: learning to predict optical flow. In *Proc. British Machine Vision Conference (BMVC)*, 2018.
- [19] 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 *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016.
- [20] 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 *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4981–4991, 2023.
- [21] Simon Meister, Junhwa Hur, and Stefan Roth. UnFlow: unsupervised learning of optical flow with a bidirectional census loss. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, pages 7251–7259, 2018.
- [22] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3070, 2015.
- [23] Sergiu Mocanu, Alan R. Moody, and April Khademi. FlowReg: fast deformable unsupervised medical image registration using optical flow. *Journal of Machine Learning for Biomedical Imaging (MELBA)*, 15:1–40, 2021.
- [24] Markus Philipp, Neal Bacher, Stefan Saur, Franziska Mathis-Ullrich, and Andrés Bruhn. From chairs to brains: Customizing optical flow for surgical activity localization. In *IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 1–5. IEEE, 2022.
- [25] Claudio S Ravasio, Theodoros Pissas, Edward Bloch, Blanca Flores, Sepehr Jalali, Danail Stoyanov, Jorge M Cardoso, Lyndon Da Cruz, and Christos Bergeles. Learned optical flow for intra-operative tracking of the retinal fundus. *International journal of computer assisted radiology and surgery*, 15:827–836, 2020.

- [26] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Zha Hongyuan. Unsupervised deep learning for optical flow estimation. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, 2017.
- [27] Saurabh Saxena, Charles Herrmann, Junhwa Hur, Abhishek Kar, Mohammad Norouzi, Deqing Sun, and David J Fleet. The surprising effectiveness of diffusion models for optical flow and monocular depth estimation. In *Proc. Conference on Neural Information Processing Systems (NeurIPS)*, pages 39443–39469, 2023.
- [28] Mohammadreza Alipour Sormoli, Mehrdad Dianati, Sajjad Mozaffari, and Roger Woodman. Optical flow based detection and tracking of moving objects for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems (TITS)*, 25(9):12587–12590, 2024.
- [29] Austin Stone, Daniel Maurer, Alper Ayvaci, Anelia Angelova, and Rico Jonschkowski. Smurf: Self-teaching multi-frame unsupervised raft with full-image warping. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3887–3896, 2021.
- [30] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2018.
- [31] Xunpei Sun, Gang Chen, and Zuoxun Hou. M2flow: A motion information fusion framework for enhanced unsupervised optical flow estimation in autonomous driving. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, pages 7140–7148, 2025.
- [32] Zitang Sun, Zhengbo Luo, and Shin'ya Nishida. Decoupled spatiotemporal adaptive fusion network for self-supervised motion estimation. *Neurocomputing*, 534:133–146, 2023.
- [33] Chuanbo Tang, Xihua Sheng, Zhuoyuan Li, Haotian Zhang, Li Li, and Dong Liu. Offline and online optical flow enhancement for deep video compression. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, pages 5118–5126, 2024.
- [34] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *Proc. European Conference on Computer Vision (ECCV)*, pages 402–419, 2020.
- [35] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 839–846, 1998.
- [36] Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, Peng Wang, and Wei Xu. Occlusion aware unsupervised learning of optical flow. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4884–4893, 2018.
- [37] Jeongbeen Yoon, Sanghyun Kim, Suha Kwak, and Minsu Cho. Optical flow domain adaptation via target style transfer. In *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2111–2121, 2024.
- [38] Jason J. Yu, Adam W. Harley, and Konstantinos G. Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. In *Proc. European Conference on Computer Vision Workshops (ECCVW)*, pages 3–10, 2016.
- [39] Shuai Yuan, Shuzhi Yu, Hannah Kim, and Carlo Tomasi. SemARFlow: injecting semantics into unsupervised optical flow estimation for autonomous driving. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9532–9543, 2023.
- [40] Shuai Yuan, Lei Luo, Zhuo Hui, Can Pu, Xiaoyu Xiang, Rakesh Ranjan, and Denis Demandolx. UnSAMFlow: unsupervised optical flow guided by segment anything model. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19027–19037, 2024.
- [41] Rui Zhao, Ruiqin Xiong, Ziluo Ding, Xiaopeng Fan, Jian Zhang, and Tiejun Huang. MRDFlow: unsupervised optical flow estimation network with multi-scale recurrent decoder. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 32(7):4639–4652, 2022.
- [42] Yiran Zhong, Pan Ji, Jianyuan Wang, Yuchao Dai, and Hongdong Li. Unsupervised deep epipolar flow for stationary or dynamic scenes. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12095–12104, 2019.
- [43] Hanyu Zhou, Yi Chang, Wending Yan, and Luxin Yan. Unsupervised cumulative domain adaptation for foggy scene optical flow. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9569–9578, 2023.