UnSAMFlow: Unsupervised Optical Flow Guided by Segment Anything Model

Shuai Yuan*, Lei Luo, Zhuo Hui, Can Pu, Xiaoyu Xiang, Rakesh Ranjan, and Denis Demandolx
Meta Reality Labs
{shuaiyuan,luoleyouluole,harryhui,cpu,xiangxiaoyu,rakeshr,denisd}@meta.com

Abstract

Traditional unsupervised optical flow methods are vulnerable to occlusions and motion boundaries due to lack of object-level information. Therefore, we propose UnSAMFlow, an unsupervised flow network that also leverages object information from the latest foundation model Segment Anything Model (SAM). We first include a self-supervised semantic augmentation module tailored to SAM masks. We also analyze the poor gradient landscapes of traditional smoothness losses and propose a new smoothness definition based on homography instead. A simple yet effective mask feature module has also been added to further aggregate features on the object level. With all these adaptations, our method produces clear optical flow estimation with sharp boundaries around objects, which outperforms state-of-the-art methods on both KITTI and Sintel datasets. Our method also generalizes well across domains and runs very efficiently.

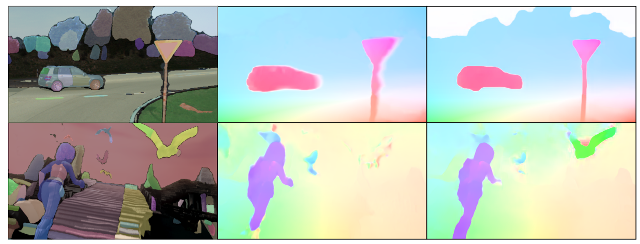


Figure 1. Our UnSAMFlow utilizes object-level information from SAM [30] to generate clear optical flow with sharp boundaries.

1. Introduction

Optical flow estimation [18, 36] involves finding pixel-level correspondences between video frames, which has broad applications such as video understanding [63], video editing [12, 28], and autonomous driving [13, 69].

Following the latest trend of deep learning in computer vision [10, 16, 31], most recent methods have modeled the optical flow problem under the supervised learning framework [9, 20, 22, 25, 27, 37, 49, 52, 55, 71], where ground-truth labels are used to train the networks. However, obtaining such labels for real-life videos is especially difficult since it usually requires precise calibrations across multiple sensors, leading to prohibitively high annotation costs [68]. This drawback makes these supervised techniques hard to be applied to large-scale real applications.

Due to the high annotation costs, much recent work has focused on the *unsupervised* training of optical flow [64]. Instead of ground-truth labels, unsupervised flow networks rely on two key principles to define losses [26, 32, 38, 41, 46, 51, 64]. Firstly, brightness constancy assumes that the corresponding points across frames should maintain similar local appearances. Secondly, the optical flow field should be spatially smooth. However, these assumptions are compromised at occlusion regions [24, 57], where foreground objects cover background appearances, and around motion boundaries [61, 65], where the motion is cut off abruptly. These issues are pervasive in real applications and have posted great challenges to unsupervised optical flow [66].

Fundamentally, the issues with occlusions and motion boundaries both stem from the *low-level* nature of optical flow, where *object-level* information is generally missing. To better handle occlusions, it is important to understand the spatial relationships and interactions between objects. Also, optical flow should be smooth only within the same continuous object region, while sharp motion boundaries are allowed near object edges. Thus, object-level information could play a key role in refining unsupervised optical flow.

Indeed, some previous methods have explored aggregating object information, using semantic segmentation to help optical flow [21, 47, 60, 69]. However, though convenient, the use of semantic segmentation is not precise because it does not distinguish different instances of the same semantic class, which may have drastically different motions. It is also constrained by the limited number of classes defined and may not recognize novel objects in the open world.

In comparison, the latest Segment Anything Model [30] (SAM) may be a better option. SAM is a general-purpose

*Affiliated with Meta at the time of this work.

image segmentation model pre-trained on very large and diverse datasets. It can separate different instances and has shown impressive zero-shot performances on objects not seen in training. In addition, SAM detects objects of various scales and levels, segmenting small object parts (such as hands and arms) as well. This can reduce complexity and help differentiate motions of object parts separately.

So motivated, we integrate SAM as additional object-level information to enhance unsupervised optical flow, which can be achieved through three novel adaptations. We first adapt the semantic augmentation module from SemARFlow [69] to enable self-supervision based on SAM masks (Sec. 3.3). Moreover, we enforce smooth motion within each SAM segment using a new regional smoothness loss based on homography (Sec. 3.4). This approach effectively rectifies numerous inconsistent flow outliers. Lastly, we design a mask feature module to aggregate features from the same SAM mask for robustness (Sec. 3.5).

Our method significantly outperforms previous methods on both KITTI [14, 42] and Sintel [2] benchmarks through both quantitative (Sec. 4.3) and qualitative (Sec. 4.4) evaluations. Notably, our network achieves 7.83% test error on KITTI-2015 [42], outperforming the state-of-the-art UPFlow [38] (9.38%) and SemARFlow [69] (8.38%) by a clear margin. As the examples show in Fig. 1, our method produces much clearer and sharper motion that is consistent with the SAM masks. Extensive ablation studies also justify the effectiveness of each proposed adaptation (Sec. 4.5). Further analysis shows that our method generalizes well across domains (Sec. 4.6) and runs efficiently (Sec. 4.7).

In summary, our contributions are as follows.

- To the best of our knowledge, we are the first to effectively combine SAM [30] with unsupervised optical flow estimation, which helps learning optical flow for wide-range real-world videos without ground-truth labels.
- We analyze the issues of previous smoothness losses with visualizations and propose a new smoothness loss definition based on homography and SAM as a solution.
- We show how SAM masks can be processed, represented, and aggregated into neural networks, which can be directly extended to other tasks using SAM.

2. Related work

Unsupervised optical flow Traditional methods [18, 36, 70] optimize optical flow based on brightness constancy and local smoothness. These constraints have been transformed to photometric and smoothness losses in early unsupervised networks [46, 64]. Since then, more specific modules have been proposed, including occlusion-aware adjustments [24, 35, 41, 57, 67], iterative refinement [22], learned upsampler [37], self-supervision [32–34, 51], dataset learning [15, 19, 54]. The latest SMURF [51] based on RAFT [55] has also achieved outstanding performances.

Segment Anything Model (SAM) Segment Anything Model (SAM) [30] is a recent general-purpose foundation model for image segmentation tasks. The model is trained on enormous high-quality annotated images and is designed to accept prompts (points, boxes, *etc.*) to retrieve object masks. Its trained model can transfer well zero-shot to new data distributions and thus has been applied to many vision tasks and applications such as object tracking [7, 62], video segmentation [45, 72], neural rendering [3, 5, 48], and medical imaging [17, 39, 40, 59]

Combining optical flow and object information To aggregate object-level information, many previous methods have combined semantic or instance segmentation models off-the-shelf as object cues to help refine optical flow [1, 47, 56, 60, 69]. Given semantics, most methods reason the rigid motions of objects and conduct refinement based on geometric techniques such as homography [47], epipolar geometry [1], and SfM [47, 60]. SemARFlow [69] is a latest neural network that incorporates semantics on the feature level and through self-supervision, which we follow closely. Besides, joint training has also been explored to benefit both optical flow and segmentation tasks based on semantic consistency and occlusion reasoning [6, 8, 21].

One concurrent work, SAMFlow [73], also combines SAM with optical flow. However, we focus on *unsupervised* flow as opposed to their *supervised* flow. Furthermore, our method imports SAM *outputs* instead of SAM *features*, so our trained network can automatically be reused by any segmentation model as long as it generates SAM-style object masks. These discrepancies make our method more flexible and feasible in real applications without needing labels.

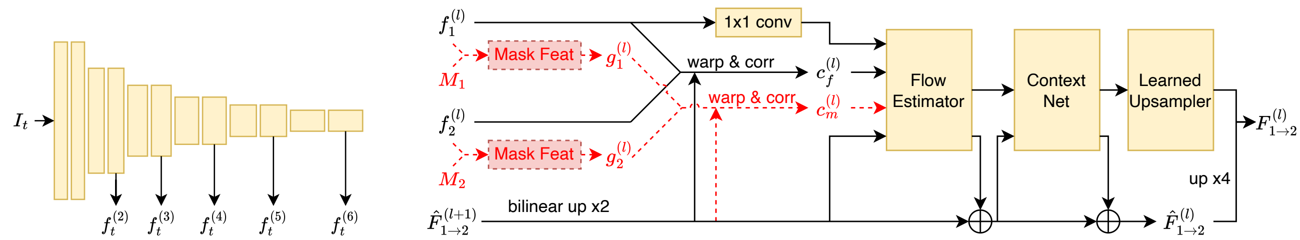
3. Method

3.1. Problem formulation

Unsupervised optical flow Given two consecutive RGB frames $I_1, I_2 \in \mathbb{R}^{H\times W\times 3}$, unsupervised optical flow estimation aims at estimating the dense optical flow field $F_{1\to 2} \in \mathbb{R}^{H\times W\times 2}$ without using ground-truth labels.

SAM mask detection For each input frame I_t ($t \in \{1,2\}$), we can compute its SAM [30] masks $M_t = \{0,1\}^{n_t\times H\times W}$, which is composed of the binary masks of the n_t objects found in I_t. The masks are generated by prompting SAM using a grid of around 1k points along with post-processing such as NMS [43] to drop redundant masks.

To utilize SAM masks in optical flow training, one key question is how to effectively process, represent, and aggregate these masks in the network. Different from semantic or instance segmentation, SAM masks do not identify the semantic classes of each object, so one-hot representations are not applicable. Also, the number of detected masks



(a) Encoder for each image I_t ($t \in \{1, 2\}$)

(b) One iteration of the iterative decoder at the l-th level ($2 \le l \le 6$).

Figure 2. Our network structure. The red part highlights our mask feature adaptation ("+mf"), which is only applied in our second setting where the SAM masks, M_1 and M_2, are used as additional inputs to the network. See more detailed network structures in Appendix A.1.

Figure 3. Examples of object crops selected from KITTI [42] and Sintel [2] using SAM [30] for semantic augmentation (Sec. 3.3)

n_t may vary from sample to sample. Moreover, the relationship between these masks and all pixels is not strictly one-to-one. Instead, a pixel can belong to multiple different masks, for instance, in the case of embedded objects like cars and wheels. Conversely, some pixels may not be assigned to any mask at all. These technicalities have posted great challenges, and we will show how we tackle these issues in later chapters.

Two problem settings For the sake of practicality, we consider two settings. In the first setting, we do *not* use SAM masks as additional inputs to the network. Instead, we only apply SAM during training to generate better loss signals, so SAM is not needed at inference time once training is completed. In addition, for the second setting, we also input SAM masks to the network to generate object-level mask features. This setting should yield better performances, albeit with the trade-off of extra overhead during the generation of SAM masks at inference time.

In this paper, we mainly propose three adaptations to utilize SAM information in the flow network, namely the semantic augmentation (Sec. 3.3), the homography smoothness loss (Sec. 3.4), and the mask feature module (Sec. 3.5). These three adaptions can be plugged in independently. The former two only need SAM during training, so they can be applied to both problem settings. The last one involves adding SAM inputs and new learnable weights to the network, so we only apply it in the second setting.

3.2. Network structure and loss

We first build our baseline network from ARFlow [32], with some simple adaptations suggested by SemARFlow [69] such as adding the learned upsampler network. Our network structure is shown in Fig. 2.

Encoder We use a simple fully convolutional encoder (Fig. 2a) to extract a feature pyramid $\{f_t^{(2)}, f_t^{(3)}, \cdots, f_t^{(6)}\}$ for each input image I_t ($t \in \{1, 2\}$), where the l-th level feature $f_t^{(l)}$ has resolution $(H/2^l, W/2^l)$.

Decoder We adopt the iterative decoder used in previous work [32, 69] as our decoder. The decoder starts from a coarse level zero estimate $\hat{F}_{1\to2}^{(7)} = 0$ and iteratively refines the estimate to finer levels. Fig. 2b illustrates one iteration that refines from estimate $\hat{F}_{1\to2}^{(l+1)}$ to the finer $\hat{F}_{1\to2}^{(l)}$, which has resolution $(H/2^l, W/2^l)$. A learned upsampler network (similar to the one in RAFT [55]) is applied to upsample $\hat{F}_{1\to2}^{(2)}$ by 4 times to generate our final flow estimate $F_{1\to2} = F_{1\to2}^{(2)}$ on the original resolution (H, W).

In Fig. 2b, we also highlight in red the optional mask feature module (to be discussed in Sec. 3.5), which requires the SAM masks M_1, M_2 as additional inputs to the decoder and is thus only included in our second problem setting mentioned in Sec. 3.1. See more details in Appendix A.2.

Loss We adopt the same photometric loss ℓ_{ph} as in ARFlow [32], which is a linear combination of three distance measures (L_1, SSIM [58], and Census loss [41]) between input frames and the frames warped by $F_{1\to2}$ and $F_{2\to1}$. Occluded regions estimated by bidirectional consistency check [41] are disregarded when computing ℓ_{ph}.

In addition, we also combine a semantic augmentation loss ℓ_{aug} (Sec. 3.3) and a homography smoothness loss ℓ_{hg} (Sec. 3.4), so our final loss is

$$\ell = \ell_{\text{ph}} + w_{\text{aug}} \ell_{\text{aug}} + w_{\text{hg}} \ell_{\text{hg}}, \tag{1}$$

where $w_{\text{aug}} = w_{\text{hg}} = 0.1$ are the balancing weights.

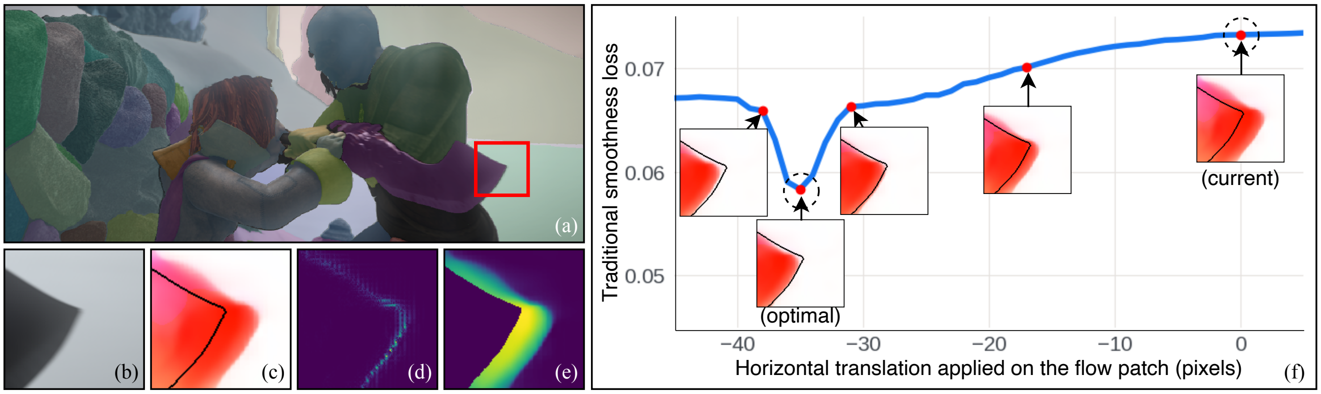


Figure 4. An example of why traditional boundary-aware smoothness loss works poorly. Sample from Sintel [2] final (ambush_5, frame #11). (a) Original image superimposed with SAM full segmentation; (b) Image patch; (c) Optical flow estimate from our baseline model superimposed with the SAM boundary (black); (d) Gradients of the traditional boundary-aware smoothness loss; (e) Gradients of our proposed homography smoothness loss; (f) Illustration of the poor landscape of traditional smoothness loss. Note that for both gradients in (d)(e), we use loss definitions based on L_2 norm for better visualizations. See Sec. 3.4 and Appendix A.6 for explanations.

3.3. Semantic augmentation as self-supervision

Inspired by SemARFlow [69], we adopt a similar semantic augmentation process to improve our network by self-supervision during training. However, we extract semantics from SAM [30] instead of semantic segmentation [4, 74].

Overview After estimating the flow $F_{1\to 2}$ for inputs I_1, I_2, we manually apply some transformations $\mathcal{T}_1$, $\mathcal{T}_2$ to I_1, I_2, respectively, to obtain the augmented $\tilde{I}_1$, $\tilde{I}_2$. The new flow after transformation $\tilde{F}_{1\to 2}$ can also be generated at the same time since all transformation parameters are known. We then run another forward pass of the network to infer flow for the augmented inputs $\tilde{I}_1$, $\tilde{I}_2$, which is then self-supervised by $\tilde{F}_{1\to 2}$ using L_1 loss (*i.e.*, the ℓ_{aug} in Eq. (1)).

The transformations $\mathcal{T}_1$, $\mathcal{T}_2$ mentioned above not only include appearance transformations (on brightness, contrast, random noise, *etc.*), 2D affine transformations (translation, rotation, scaling), and occlusion augmentation (cropping), as proposed by ARFlow [32], but also contain a special semantic augmentation that involves input semantics, as proposed by SemARFlow [69], which we discuss next.

Semantic augmentation During semantic augmentation, new objects are copied and pasted across samples. For example, we may crop out a car object from another random sample and paste it into the current sample used for training. An augmented simple motion is also applied to the cropped objects. This transformation utilizes the semantic knowledge and creates realistic samples with new occlusions.

In contrast to SemARFlow [69], which picks object crops of specific classes such as cars and poles using semantic segmentation, our method utilizes SAM masks without class labels. Consequently, we select key objects among the SAM masks by finding those masks that overlap the most with other masks. This is based on the heuristic that key objects typically consist of multiple object parts that can also be detected by SAM. Some example key objects selected are shown in Fig. 3. See more details in Appendix A.3.

3.4. Homography smoothness loss

Our second adaptation comes from the motivation that object segmentation can be used to formulate a more precise smoothness constraint to better regularize the optical flow field. We first analyze the issues of previous traditional smoothness losses and then show how we resolve those issues with the help of SAM [30].

Issues of previous smoothness losses Most previous networks define their smoothness loss based on the second-order derivatives of the flow field [64]. Optical flow field $F_{1\to 2}$ is a two-dimensional function of point $\boldsymbol{p} = (x, y)$. Previous smoothness losses are typically in the form of

$$\ell_s = \sum_{\boldsymbol{p}} \left(w_x(\boldsymbol{p}) \left\| \frac{\partial^2 F_{1\to 2}(\boldsymbol{p})}{\partial x^2} \right\| + w_y(\boldsymbol{p}) \left\| \frac{\partial^2 F_{1\to 2}(\boldsymbol{p})}{\partial y^2} \right\| \right), \tag{2}$$

in which w_x, w_y are the edge-aware weights to avoid penalties across object boundaries, where motion is not necessarily continuous. Such weights are usually derived from image edges, which often coincide with object boundaries [57]. In our case, we can obtain more accurate boundaries from SAM masks. However, we find that these boundary-aware smoothness definitions work poorly.

One example is shown in Fig. 4. The patch (Fig. 4b) exhibits a rightward motion of the blade, occluding the nearby snow background, which barely moves. We show our baseline flow estimate, as well as the object boundary, in Fig. 4c.

We can see that the estimated flow is not consistent with the object boundary due to occlusion (part of the snow regarded as moving together with the blade). In this case, the smoothness loss mostly comes from around the flow boundary, and so does its gradient (Fig. 4d). This gradient signal is very weak since the boundary only takes up a very smaller region, so its update is confined within only the small local neighborhood around the flow boundary.

Furthermore, we show that the landscape of the broadly used boundary-aware smoothness loss is problematic. We examine the smoothness loss of the patch while gradually translating the flow patch horizontally until it roughly fits the object boundary provided by SAM. The results are visualized in Fig. 4f. We can see that the optimal solution indeed finds the flow that is most consistent with the object boundary since we do not penalize across object boundaries. However, such solution lies in a very steep local minimum of the loss, while in contrast, the landscape around our current estimate is rather flat, meaning that any local change around the current estimate makes little difference to the loss. This vividly explains why traditional boundary-aware smoothness losses are very hard to optimize in training.

Regional smoothness based on homography Traditional boundary-aware smoothness (Eq. (2)) works poorly since its definition and gradient are too local. To resolve this issue, our idea is to define smoothness based on object *regions* instead of object *boundaries*.

Specifically, the inaccurate flow values in Fig. 4c can be understood as outliers in the same object region (snow). Thus, parametric models, such as homography, can be used to fix these outliers. For each object region of interest (found though occlusion estimation [41, 68]), we first estimate its homography with RANSAC [11] using the reliable correspondences provided by the current flow estimate. We define criteria to reject the estimated homography with low RANSAC inlier rate (see details in Appendix A.4). A refined flow can then be generated for that object using homography. We compute the L_1 distance between our current estimate and the refined flow as our homography smoothness loss ℓ_{hg} in Eq. (1). Our homography smoothness loss results in non-local gradients (Fig. 4e), which strongly enforces smoothness by regions.

One alternative, though, is to directly use the refined flow as our output, so the homography works through post-processing instead of loss signals [47]. Nevertheless, we still prefer defining losses because in that case, homography and SAM are only needed during training. Empirically, we do not see big differences between their performances.

3.5. Mask feature and correlation

Full segmentation representation of SAM masks To better use masks in the network, we need to transform SAM masks to a full segmentation representation, where every pixel is assigned to exactly one mask.

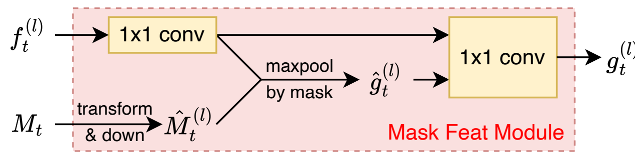


Figure 5. Our proposed mask feature module (Sec. 3.5)

Specifically, we first sort all current object masks by the size of their area. For pixels that belong to multiple masks, we assign it to the one that has the smallest area. For pixels that do not belong to any mask, we create a new "background" object mask to cover all these pixels.

Mask feature module Our mask feature computations are highlighted in Fig. 2b. Given the mask M_t and image feature $f_t^{(l)}$, we use a mask feature module to generate $g_t^{(l)}$. Similar to the warping and correlation of image features $f_t^{(l)}$ in the original ARFlow [32], we also warp mask features $g_t^{(l)}$ and compute their correlations, which are then concatenated into the input of flow estimator network in Fig. 2b.

The detailed structure of our mask feature module is depicted in Fig. 5. We first transform the SAM masks M_t to full segmentation representation and downsize it to the l-level resolution. We then compute a new feature from $f_t^{(l)}$ and apply max-pooling by segmentation. Specifically, for each object segmentation, we apply max-pooling among all features of that object and copy the pooled feature to all those pixels, yielding a pooled feature map $\hat{g}_t^{(l)}$. Finally, we concatenate the feature before max-pooling with $\hat{g}_t^{(l)}$, and extract our mask feature $g_t^{(l)}$. This module aggregates features on the higher object level and can, thus, compensate the original pixel-level image features. See more details in Appendix A.5.

4. Experiments

4.1. Datasets

We conduct experiments on KITTI [14, 42] and Sintel [2] datasets and follow the same training data schedules from previous methods [32, 34, 37]. For KITTI [14, 42], we first train on raw sequences (55.7k samples) and then fine-tune on the multi-view extension subset (5.9k samples). For Sintel [2], we first train on raw frames (12.5k samples) provided by ARFlow [32] and then fine-tune on clean and final passes together (2.1k samples). We also adopt the Sintel sub-splits used in ARFlow [32], which divide the original dataset by scenes into two subsets of 1k samples, for two-fold cross validation. Images from the test scenes have been excluded from the raw sequences for both datasets.

	Method	Train 2012 EPE	Train 2015 EPE	Test 2012 Fl-noc	Test 2012 EPE	Test 2015 Fl-all	Test 2015 Fl-noc	Test 2015 Fl-bg	Test 2015 Fl-fg	Param.
Supervised	PWC-Net+ [53]	-	(1.50)	3.36	1.4	7.72	4.91	7.69	7.88	8.8M
	IRR-PWC [22]	-	(1.63)	3.21	1.6	7.65	4.86	7.68	7.52	6.4M
	RAFT [55]	-	(0.63)	-	-	5.10	3.07	4.74	6.87	5.3M
	FlowFormer [20]	-	(0.53)	-	-	4.68	**2.69**	**4.37**	**6.18**	18.2M
	SAMFlow [73]*†	-	-	-	-	**4.49**	-	-	-	-
Unsupervised	UnFlow-CSS [41]	3.29	8.10	-	-	23.27	-	-	-	116.6M
	DDFlow [33]	2.35	5.72	4.57	3.0	14.29	9.55	13.08	20.40	4.3M
	SelFlow [34]	1.69	4.84	4.31	2.2	14.19	9.65	12.68	21.74	4.8M
	SimFlow [23]	-	5.19	-	-	13.38	8.21	12.60	17.27	-
	ARFlow [32]	1.44	2.85	5.02	1.8	11.80	8.91	10.30	19.32	2.2M
	UFlow [26]	1.68	(2.71)	4.26	1.9	11.13	8.41	9.78	17.87	-
	UPFlow [38]	1.27	2.45	-	**1.4**	9.38	-	-	-	3.5M
	Ours (baseline)	1.32	2.44	4.05	1.6	9.60	6.77	8.74	**13.89**	2.5M
	Ours (+aug)*	1.33	2.26	4.15	1.6	9.05	6.46	7.96	14.55	2.5M
	Ours (+aug +hg)*	1.27	2.11	3.89	1.5	8.18	6.04	6.67	15.72	2.5M
	Ours (+aug +hg +mf)*†	**1.26**	**2.01**	**3.79**	**1.4**	**7.83**	**5.67**	**6.40**	14.98	2.6M

Table 1. KITTI benchmark errors (EPE/px and Fl/%). Metrics evaluated at "all" (all pixels), "noc" (non-occlusions), "bg" (background), and "fg" (foreground). "+aug": semantic augmentation module; "+hg": homography smoothness loss; "+mf": mask feature module. "*": SAM used in training; "†": SAM used in inference. Numbers with parentheses indicate that the same evaluation data were used in training.

	Method	Train Clean all	Train Final all	Test Clean all	Test Clean noc	Test Clean occ	Test Final all	Test Final noc	Test Final occ	Param.
Supervised	PWC-Net+ [53]	(1.71)	(2.34)	3.45	1.41	20.12	4.60	2.25	23.70	8.8M
	IRR-PWC [22]	(1.92)	(2.51)	3.84	1.47	23.22	4.58	2.15	24.36	6.4M
	RAFT [55]	(0.77)	(1.27)	1.61	0.62	9.65	2.86	1.41	14.68	5.3M
	FlowFormer [20]	(0.48)	(0.74)	1.16	0.42	7.16	2.09	**0.96**	11.30	18.2M
	SAMFlow [73]*†	-	-	**1.00**	**0.38**	**5.97**	**2.08**	1.04	**10.60**	-
Unsupervised	UnFlow-CSS [41]	-	7.91	9.38	5.37	42.11	10.22	6.06	44.11	116.6M
	DDFlow [33]	(2.92)	(3.98)	6.18	2.27	38.05	7.40	3.41	39.94	4.3M
	SelFlow [34]	(2.88)	(3.87)	6.56	2.67	38.30	6.57	3.12	34.72	4.8M
	SimFlow [23]	(2.86)	(3.57)	5.93	2.16	36.66	6.92	3.02	38.70	-
	ARFlow [32]	(2.79)	(3.73)	4.78	1.91	28.26	5.89	2.73	31.60	2.2M
	UFlow [26]	(2.50)	(3.39)	5.21	2.04	31.06	6.50	3.08	34.40	-
	UPFlow [38]	(2.33)	(2.67)	4.68	1.71	28.95	5.32	**2.42**	28.93	3.5M
	Ours (baseline)	(2.67)	(3.63)	4.29	1.64	25.96	5.81	2.76	30.60	2.5M
	Ours (+aug)*	(2.35)	(3.33)	4.00	**1.58**	23.76	5.33	2.53	28.17	2.5M
	Ours (+aug +hg)*	(2.25)	(3.10)	4.00	1.76	22.36	5.22	2.62	**26.40**	2.5M
	Ours (+aug +hg +mf)*†	(2.21)	(3.07)	**3.93**	1.67	**22.34**	**5.20**	2.56	26.75	2.6M

Table 2. Sintel benchmark errors (EPE/px). Metrics evaluated at "all" (all pixels), "noc" (non-occlusions), and "occ" (occlusions). "+aug": semantic augmentation module; "+hg": homography smoothness loss; "+mf": mask feature module. "*": SAM used in training; "†": SAM used in inference. Numbers with parentheses indicate that the same evaluation data were used in training.

4.2. Implementation details

The model is implemented in PyTorch [44]. We train the network using the Adam optimizer [29] ($\beta_1 = 0.9, \beta_2 = 0.999$) with batch size 8. For both datasets, we first train on raw data using a constant learning rate 2e-4 for 100k iterations and then fine-tune on the original dataset using the OneCycleLR scheduler [50] with maximum learning rate 4e-4 for another 100k iterations. Similar to SemARFlow [69], we only turn on the semantic augmentation

Method	Semantics	KITTI	
		2015	2012
JFS [21]	Sem. Seg.	16.47	-
SOF [47]	Sem. Seg.	15.99	-
MRFlow [60]	Sem. Seg.	12.19	-
SDF[1]	Ins. Seg.	11.01	7.69
SemARFlow [69]	Sem. Seg.	8.38	7.35
Ours (+aug +hg +mf)	SAM	**7.83**	**7.05**

Table 3. KITTI test errors (Fl-all/%) compared with other unsupervised semantic optical flow methods. "-": data not available.

w_{aug}	w_{hg}	Sintel		KITTI	
		Final	Clean	2015	2012
0.1*	0.1*	**3.35**	**2.53**	2.11	**1.27**
0.1	0	3.53	2.54	2.26	1.33
0.1	0.2	3.45	2.63	**2.09**	1.28
0.2	0.2	3.45	2.60	2.15	1.29
0.2	0.1	3.50	2.59	2.13	1.29
0	0.1	3.61	2.79	2.16	1.30

Table 4. Ablation Study on Ours (+aug +hg): Balancing weights between w_{aug} and w_{hg} in Eq. (1). * indicates our final setting.

Smoothness Loss	Sintel		KITTI	
	Final	Clean	2015	2012
Homography*	**3.35**	**2.53**	**2.11**	**1.27**
Image-edge-aware	3.63	2.56	2.35	1.35
SAM-boundary-aware	3.65	2.62	2.33	1.36

Table 5. Ablation Study on Ours (+aug +hg): Different smoothness loss definition (Sec. 3.4). * indicates our current setting.

Mask Feature Module	Sintel		KITTI	
	Final	Clean	2015	2012
Concat*	**3.29**	**2.43**	**2.01**	**1.26**
Residual	3.31	2.47	2.05	1.28
Concat + Residual	3.34	2.52	2.06	1.27
No +mf	3.35	2.53	2.11	1.27

Table 6. Ablation Study on Ours (+aug +hg +mf): Different mask feature modules (Sec. 3.5). * indicates our final setting.

and homography smoothness modules after 150k iterations.

In terms of Segment Anything Model [30], we use the off-the-shelf default ViT-H pretrained model, which generates an average of 63.7 object masks for each KITTI sample [42] and around 82.9 masks for each Sintel sample [2].

For data augmentation, we follow ARFlow [32] and include appearance transformations (brightness, contrast, saturation, hue, gaussian blur, *etc*.), random horizontal flipping, and random swapping of input images. We resize the inputs to dimension 256×832 for KITTI and 448×1024 for Sintel before feeding into the network.

4.3. Benchmark testing

Our benchmark testing results are shown in Tabs. 1 and 2. Our final models with all three adaptations significantly outperform state-of-the-art unsupervised methods on both KITTI [14, 42] and Sintel [2] datasets on almost all evaluation metrics. Our final model achieves 7.84% error rate on KITTI-2015 test set, which is much better than UPFlow [37] (9.38%) and ARFlow [32] (11.80%, the backbone network that we adapt from). All these results show the benefits of utilizing SAM models in unsupervised optical flow training.

In Tabs. 1 and 2, we can also see that our errors decline progressively as we incrementally add each proposed module to the network. This justifies the effectiveness of all our proposed adaptations. Also, for the setting that we do not use SAM masks as network inputs ("+aug +hg"), our model also outperforms state-of-the-art methods on both datasets. This implies that our approach has the potential to enhance unsupervised flow networks solely by optimizing training, guided by SAM, without introducing any additional computational overhead during inference.

Notably, our networks exhibit substantial improvements from SAM particularly on real datasets, such as KITTI [14, 42], compared with animation images in Sintel [2]. This is because SAM is mostly trained on real-life images, so it produces masks of higher quality for KITTI than for Sintel.

Tab. 3 shows the comparison among current semantics-guided optical flow methods. Our model guided by SAM outshines all previous methods guided by semantic or instance segmentation, even though SAM is not trained on KITTI [42]. This indicates the great potential of SAM as a zero-shot general-purpose semantic model that could be used directly in other tasks such as optical flow estimation.

4.4. Qualitative results

Figs. 6 and 7 show some qualitative examples of our final model, compared with previous state-of-the-art methods. We can see that our network outputs better flow around objects with much sharper boundaries, which are consistent with the SAM mask inputs. Our method can also handle different lighting conditions (dark shadows, bright reflections) better thanks to the robust masks provided by SAM.

4.5. Ablation studies

We do extensive ablation studies to analyze the effectiveness of our proposed adaptations and their detailed settings. For KITTI [14, 42], we compute validation errors on the original train set images. For Sintel [2], we apply two-fold cross validation and report the average validation errors.

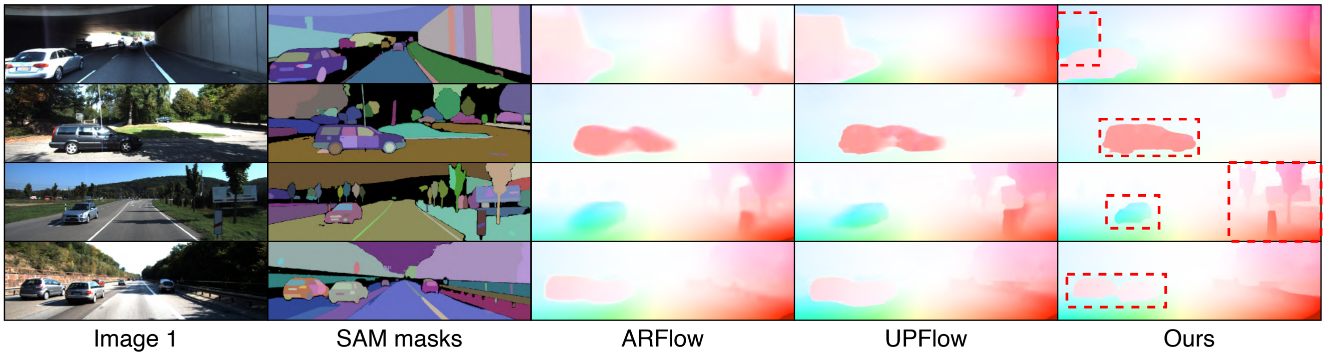


Figure 6. KITTI-2015 test qualitative examples (sample frame #48, 72, 190, 196). See more examples in Appendix B.2

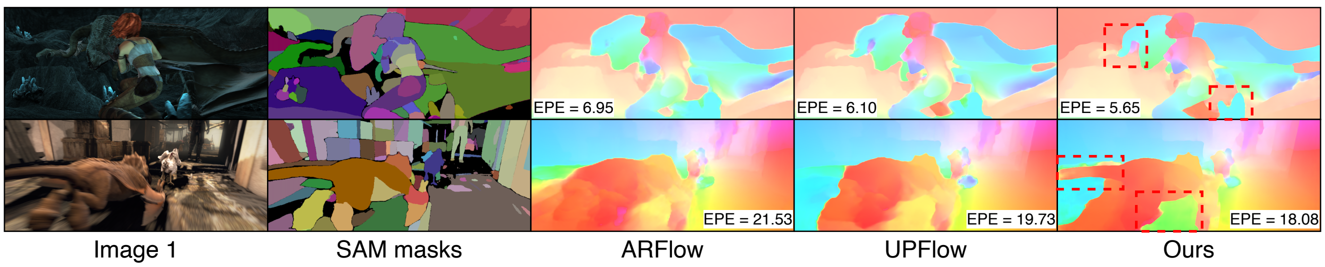


Figure 7. Sintel (final pass) test qualitative examples (sample: cave_3 frame 16; market_4 frame 47). See more examples in Appendix B.2

Loss weights We tune the loss weights w_{aug} and w_{hg} in Tab. 4. We also compare with the settings where either w_{aug} or w_{hg} equals zero, which means we turn off the semantic augmentation or homography smoothness module. The results show that our current setting works the best, and the ablation of each module results in loss of performance.

Smoothness definitions In Tab. 5, we can see that our regional smoothness loss based on homography works significantly better than the traditionally used edge-aware smoothness loss based on image edges or SAM boundaries. These results are consistent with our analysis in Sec. 3.4.

Mask feature modules We also experiment some other ways of aggregating mask features and image features in Tab. 6. "Residual" refers to adding the processed mask features to image features as a residual connection. The results show that our current setting works slightly better than using residual connections.

4.6. Generalization ability

We show that our flow network guided by SAM exhibits great generalization ability across dataset domains in Tab. 7. Specifically, we train on one of the datasets (KITTI [42] or Sintel [2]) and then test directly on the other without fine-tuning. Our final model guided by SAM obtains clearly better results than our baseline model without SAM.

Method	KITTI→Sintel		Sintel→KITTI	
	Final	Clean	2015	2012
Ours (w/ SAM)	**5.75**	**4.90**	**7.58**	2.99
Ours (w/o SAM)	7.02	6.39	8.59	**2.93**

Table 7. Generalization ability. Training on one dataset and testing directly on the other dataset. We show Sintel/KITTI train set EPEs.

4.7. Time efficiency

Our network operates very efficiently in real time. For each RGB sample of dimension 376×1242, our network inference takes $0.0334(\pm 0.0038)$ second on one Tesla P100 GPU, excluding the time for computing SAM masks.

5. Conclusion

We propose UnSAMFlow, an unsupervised optical flow network guided by object information from Segment Anything Model (SAM), with three novel adaptations, namely semantic augmentation, homography smoothness, and mask feature correlation. Our method achieves state-of-the-art results and exhibits visible improvements.

Limitations Our performance relies on the accuracy of SAM masks, which may be undermined for samples with serious lighting issues, artifacts, or motion blur. The lack of semantic classes in the SAM output also makes its object information incomplete, awaiting future improvements.

References

[1] Min Bai, Wenjie Luo, Kaustav Kundu, and Raquel Urtasun. Exploiting semantic information and deep matching for optical flow. In *ECCV*, pages 154–170. Springer, 2016. 2, 7

[2] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, pages 611–625. Springer-Verlag, 2012. 2, 3, 4, 5, 7, 8

[3] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Wei Shen, Lingxi Xie, Xiaopeng Zhang, and Qi Tian. Segment anything in 3d with nerfs. *arXiv preprint arXiv:2304.12308*, 2023. 2

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pages 801–818, 2018. 4

[5] Xiaokang Chen, Jiaxiang Tang, Diwen Wan, Jingbo Wang, and Gang Zeng. Interactive segment anything nerf with feature imitation. *arXiv preprint arXiv:2305.16233*, 2023. 2

[6] Jingchun Cheng, Yi-Hsuan Tsai, Shengjin Wang, and Ming-Hsuan Yang. Segflow: Joint learning for video object segmentation and optical flow. In *ICCV*, pages 686–695, 2017. 2

[7] Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang. Segment and track anything. *arXiv preprint arXiv:2305.06558*, 2023. 2

[8] Mingyu Ding, Zhe Wang, Bolei Zhou, Jianping Shi, Zhiwu Lu, and Ping Luo. Every frame counts: Joint learning of video segmentation and optical flow. In *AAAI*, pages 10713–10720, 2020. 2

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

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1

[11] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 5

[12] Chen Gao, Ayush Saraf, Jia-Bin Huang, and Johannes Kopf. Flow-edge guided video completion. In *ECCV*, pages 713–729. Springer, 2020. 1

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361. IEEE, 2012. 1

[14] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2, 5, 7

[15] Yunhui Han, Kunming Luo, Ao Luo, Jiangyu Liu, Haoqiang Fan, Guiming Luo, and Shuaicheng Liu. Realflow: Em-based realistic optical flow dataset generation from videos. In *ECCV*, pages 288–305. Springer, 2022. 2

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645. Springer, 2016. 1

[17] Sheng He, Rina Bao, Jingpeng Li, P Ellen Grant, and Yangming Ou. Accuracy of segment-anything model (sam) in medical image segmentation tasks. *arXiv preprint arXiv:2304.09324*, 2023. 2

[18] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981. 1, 2

[19] Hsin-Ping Huang, Charles Herrmann, Junhwa Hur, Erika Lu, Kyle Sargent, Austin Stone, Ming-Hsuan Yang, and Deqing Sun. Self-supervised autoflow. In *CVPR*, pages 11412–11421, 2023. 2

[20] Zhaoyang Huang, Xiaoyu Shi, Chao Zhang, Qiang Wang, Ka Chun Cheung, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer: A transformer architecture for optical flow. In *ECCV*, pages 668–685. Springer, 2022. 1, 6

[21] Junhwa Hur and Stefan Roth. Joint optical flow and temporally consistent semantic segmentation. In *ECCV*, pages 163–177. Springer, 2016. 1, 2, 7

[22] Junhwa Hur and Stefan Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *CVPR*, pages 5754–5763, 2019. 1, 2, 6

[23] Woobin Im, Tae-Kyun Kim, and Sung-Eui Yoon. Unsupervised learning of optical flow with deep feature similarity. In *ECCV*, pages 172–188. Springer, 2020. 6

[24] Joel Janai, Fatma Guney, Anurag Ranjan, Michael Black, and Andreas Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *ECCV*, pages 690–706, 2018. 1, 2

[25] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *ICCV*, pages 9772–9781, 2021. 1

[26] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. In *ECCV*, pages 557–572. Springer, 2020. 1, 6

[27] Hyunyoung Jung, Zhuo Hui, Lei Luo, Haitao Yang, Feng Liu, Sungjoo Yoo, Rakesh Ranjan, and Denis Demandolx. Anyflow: Arbitrary scale optical flow with implicit neural representation. In *CVPR*, pages 5455–5465, 2023. 1

[28] Hannah Halin Kim, Shuzhi Yu, Shuai Yuan, and Carlo Tomasi. Cross-attention transformer for video interpolation. In *ACCVW*, pages 320–337, 2022. 1

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2014. 6

[30] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *ICCV*, pages 4015–4026, 2023. 1, 2, 3, 4, 7

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 25, 2012. 1

[32] Liang Liu, Jiangning Zhang, Ruifei He, Yong Liu, Yabiao Wang, Ying Tai, Donghao Luo, Chengjie Wang, Jilin Li, and Feiyue Huang. Learning by analogy: Reliable supervision from transformations for unsupervised optical flow estimation. In *CVPR*, pages 6489–6498, 2020. 1, 2, 3, 4, 5, 6, 7

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

[34] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *CVPR*, pages 4571–4580, 2019. 2, 5, 6

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

[36] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, pages 674–679, 1981. 1, 2

[37] Ao Luo, Fan Yang, Kunming Luo, Xin Li, Haoqiang Fan, and Shuaicheng Liu. Learning optical flow with adaptive graph reasoning. In *AAAI*, pages 1890–1898, 2022. 1, 2, 5, 7

[38] Kunming Luo, Chuan Wang, Shuaicheng Liu, Haoqiang Fan, Jue Wang, and Jian Sun. Upflow: Upsampling pyramid for unsupervised optical flow learning. In *CVPR*, pages 1045–1054, 2021. 1, 2, 6

[39] Jun Ma and Bo Wang. Segment anything in medical images. *arXiv preprint arXiv:2304.12306*, 2023. 2

[40] Maciej A Mazurowski, Haoyu Dong, Hanxue Gu, Jichen Yang, Nicholas Konz, and Yixin Zhang. Segment anything model for medical image analysis: an experimental study. *Medical Image Analysis*, 89:102918, 2023. 2

[41] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI*, 2018. 1, 2, 3, 5, 6

[42] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015. 2, 3, 5, 7, 8

[43] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *International Conference on Pattern Recognition*, pages 850–855. IEEE, 2006. 2

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035. Curran Associates, Inc., 2019. 6

[45] Frano Rajič, Lei Ke, Yu-Wing Tai, Chi-Keung Tang, Martin Danelljan, and Fisher Yu. Segment anything meets point tracking. *arXiv preprint arXiv:2307.01197*, 2023. 2

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

[47] Laura Sevilla-Lara, Deqing Sun, Varun Jampani, and Michael J Black. Optical flow with semantic segmentation and localized layers. In *CVPR*, pages 3889–3898, 2016. 1, 2, 5, 7

[48] Qiuhong Shen, Xingyi Yang, and Xinchao Wang. Anything-3d: Towards single-view anything reconstruction in the wild. *arXiv preprint arXiv:2304.10261*, 2023. 2

[49] Xiaoyu Shi, Zhaoyang Huang, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer++: Masked cost volume autoencoding for pretraining optical flow estimation. In *CVPR*, pages 1599–1610, 2023. 1

[50] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial Intelligence and Machine Learning for Multi-domain Operations Applications*, pages 369–386. SPIE, 2019. 6

[51] Austin Stone, Daniel Maurer, Alper Ayvaci, Anelia Angelova, and Rico Jonschkowski. Smurf: Self-teaching multi-frame unsupervised raft with full-image warping. In *CVPR*, pages 3887–3896, 2021. 1, 2

[52] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, pages 8934–8943, 2018. 1

[53] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Models matter, so does training: An empirical study of cnns for optical flow estimation. *IEEE TPAMI*, 42(6):1408–1423, 2019. 6

[54] Deqing Sun, Daniel Vlasic, Charles Herrmann, Varun Jampani, Michael Krainin, Huiwen Chang, Ramin Zabih, William T Freeman, and Ce Liu. Autoflow: Learning a better training set for optical flow. In *CVPR*, pages 10093–10102, 2021. 2

[55] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419. Springer, 2020. 1, 2, 3, 6

[56] Xianshun Wang, Dongchen Zhu, Yanqing Liu, Xiaoqing Ye, Jiamao Li, and Xiaolin Zhang. Semflow: Semantic-driven interpolation for large displacement optical flow. *IEEE Access*, 7:51589–51597, 2019. 2

[57] Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, Peng Wang, and Wei Xu. Occlusion aware unsupervised learning of optical flow. In *CVPR*, pages 4884–4893, 2018. 1, 2, 4

[58] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 3

[59] Junde Wu, Rao Fu, Huihui Fang, Yuanpei Liu, Zhaowei Wang, Yanwu Xu, Yueming Jin, and Tal Arbel. Medical sam adapter: Adapting segment anything model for medical image segmentation. *arXiv preprint arXiv:2304.12620*, 2023. 2

[60] Jonas Wulff, Laura Sevilla-Lara, and Michael J Black. Optical flow in mostly rigid scenes. In *CVPR*, pages 4671–4680, 2017. 1, 2, 7

[61] Huizong Yang and Anthony Yezzi. Decomposing the tangent of occluding boundaries according to curvatures and torsions. In *European Conference on Computer Vision*, pages 123–138. Springer, 2022. 1

[62] Jinyu Yang, Mingqi Gao, Zhe Li, Shang Gao, Fangjing Wang, and Feng Zheng. Track anything: Segment anything meets videos. *arXiv preprint arXiv:2304.11968*, 2023. 2

[63] Vickie Ye, Zhengqi Li, Richard Tucker, Angjoo Kanazawa, and Noah Snavely. Deformable sprites for unsupervised video decomposition. In *CVPR*, pages 2657–2666, 2022. 1

[64] 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 *ECCVW*, pages 3–10. Springer, 2016. 1, 2, 4

[65] Shuzhi Yu, Hannah H Kim, Shuai Yuan, and Carlo Tomasi. Unsupervised flow refinement near motion boundaries. In *BMVC*. BMVA Press, 2022. 1

[66] Shuai Yuan. *Assisting Unsupervised Optical Flow Estimation With External Information*. PhD thesis, Duke University, 2023. 1

[67] Shuai Yuan and Carlo Tomasi. Ufd-prime: Unsupervised joint learning of optical flow and stereo depth through pixel-level rigid motion estimation. *arXiv preprint arXiv:2310.04712*, 2023. 2

[68] Shuai Yuan, Xian Sun, Hannah Kim, Shuzhi Yu, and Carlo Tomasi. Optical flow training under limited label budget via active learning. In *ECCV*, pages 410–427. Springer Nature Switzerland, 2022. 1, 5

[69] Shuai Yuan, Shuzhi Yu, Hannah Kim, and Carlo Tomasi. Semarflow: Injecting semantics into unsupervised optical flow estimation for autonomous driving. In *ICCV*, pages 9566–9577, 2023. 1, 2, 3, 4, 6, 7

[70] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Pattern Recognition*, pages 214–223. Springer, 2007. 2

[71] Feihu Zhang, Oliver J Woodford, Victor Adrian Prisacariu, and Philip HS Torr. Separable flow: Learning motion cost volumes for optical flow estimation. In *ICCV*, pages 10807–10817, 2021. 1

[72] Zhenghao Zhang, Zhichao Wei, Shengfan Zhang, Zuozhuo Dai, and Siyu Zhu. Uvosam: A mask-free paradigm for unsupervised video object segmentation via segment anything model. *arXiv preprint arXiv:2305.12659*, 2023. 2

[73] Shili Zhou, Ruian He, Weimin Tan, and Bo Yan. Samflow: Eliminating any fragmentation in optical flow with segment anything model. *arXiv preprint arXiv:2307.16586*, 2023. 2, 6

[74] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *CVPR*, pages 8856–8865, 2019. 4