SemARFlow: Injecting Semantics into Unsupervised Optical Flow Estimation for Autonomous Driving

Shuai Yuan, Shuzhi Yu, Hannah Kim, and Carlo Tomasi

Duke University

{shuai, shuzhiyu, hannah, tomasi}@cs.duke.edu

Abstract

Unsupervised optical flow estimation is especially hard near occlusions and motion boundaries and in low-texture regions. We show that additional information such as semantics and domain knowledge can help better constrain this problem. We introduce SemARFlow, an unsupervised optical flow network designed for autonomous driving data that takes estimated semantic segmentation masks as additional inputs. This additional information is injected into the encoder and into a learned upsampler that refines the flow output. In addition, a simple yet effective semantic augmentation module provides self-supervision when learning flow and its boundaries for vehicles, poles, and sky. Together, these injections of semantic information improve the KITTI-2015 optical flow test error rate from 11.80% to 8.38%. We also show visible improvements around object boundaries as well as a greater ability to generalize across datasets. Code is available at https://github.com/duke-vision/semantic-unsup-flow-release.

1. Introduction

Optical flow estimation, *i.e.*, pixel-level motion tracking across video frames, has broad applications in many computer vision tasks that include object tracking [61], video editing [14, 31], and autonomous driving [6, 58].

Thanks to the success of deep convolutional neural networks [34, 17, 36] and transformer networks [33, 69, 88, 28] in computer vision, many top-performing supervised optical flow networks have been proposed in recent years [10, 65, 22, 68, 85], in which ground-truth labels supervise training. However, real optical flow is hard to label, so most supervised methods train (or at least pre-train) on synthetic datasets [10, 44, 4, 56] which makes them hard to adapt to real applications due to the significant gap between synthetic and real data [70, 29].

Due to label scarcity, *unsupervised* training of optical
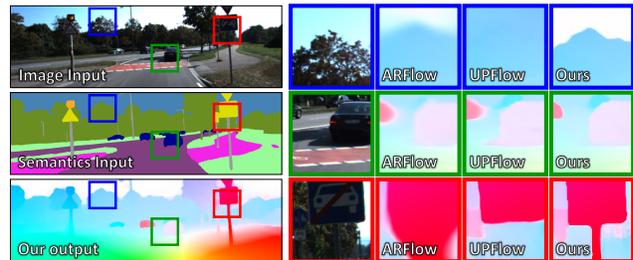

Figure 1. An example on KITTI-2015 test set (sample #63). Only one input frame is shown for conciseness. Our SemARFlow takes additional semantic segmentation inputs (estimated by an off-the-shelf model) and outputs much sharper flow around semantic boundaries.

flow estimators [81] instead uses loss terms based on the assumptions of constant brightness and smooth flow [45, 27, 42]. Some self-supervision techniques have also been studied to enhance model performance [38, 39, 37]. Unsupervised training makes it possible to train flow networks directly on large real datasets from the target domain.

Even so, unsupervised optical flow estimation is a poorly constrained problem. Brightness is not constant in regions with occlusions [72] or on shiny surfaces [43], nor is it smooth across motion boundaries [30, 82]. Moreover, points in regions with poor texture content [59] or in dark shadows [58] are difficult to track as they are easily confused with neighboring points (the so-called aperture problem). This dearth of reliable constraints makes the unsupervised training of flow networks especially challenging.

A natural way to address this issue is to inject additional constraints in the form of semantics and domain knowledge. For example, in autonomous driving applications [84, 6], we have clear expectations about object types and layouts of the scene, as well as prior knowledge of how each type of object typically moves. We focus on this application domain and show that this additional information helps the network achieve better flow results.

To make an autonomous driving system work well in

reality, it is best to train it on real data. However, annotating real driving video with optical flow labels is expensive [83], as it requires careful synchronization and calibration of diverse sensors including cameras, LiDAR, and GPS/IMU [16], aided by some manual annotation and curation based on CAD models of moving objects [46].

In contrast, annotating semantic labels seems much more feasible, and indeed semantic labels are available in most (if not all) existing driving datasets. We consider semantic segmentation because it provides semantics at the pixel level, the same level as optical flow. As one of the most popular and well-studied tasks in modern computer vision, semantic segmentation [40, 47, 87] has been extensively adopted for autonomous driving systems. In this paper, we show that adding semantic segmentation inputs helps improve unsupervised optical flow performance significantly.

Specifically, we first infer semantic segmentation maps using an off-the-shelf model [89], which of course is trained with semantic labels. An encoder with semantic map input is used to aggregate image and semantic features (Sec. 3.1), and a learned upsampler is added into the iterative decoder to refine flow around object boundaries given semantic inputs (Sec. 3.2). We also propose a simple yet effective semantic augmentation module for self-supervision, which provides realistic augmentations specific to the vehicles, poles, and sky classes based on domain knowledge and segmentation maps (Sec. 3.3). An occluder cache is implemented to improve efficiency (Sec. 3.4). Semantic augmentation provides challenging samples for self-supervision, which help train flow better in occluded regions and around foreground object boundaries.

Overall, by injecting semantic segmentation inputs, our SemARFlow network achieves significantly better unsupervised flow results both quantitatively (Sec. 4.3) and qualitatively (Sec. 4.4). Adapted from ARFlow [37], SemARFlow reduces KITTI-2015 [46] test error from 11.80% to 8.38% and out-performs the current unsupervised state-of-the-art UPFlow [42] (9.38%) by a clear margin. The example in Fig. 1 also demonstrates visible improvements around object boundaries. The effectiveness of each module is justified through an extensive ablation study (Sec. 4.5) and improvement analysis (Sec. 4.6). In addition to performance boost, unsupervised flow networks with additional semantic inputs generalize better across different datasets (Sec. 4.7).

Our research is essentially novel compared to previous approaches. Some early work incorporates semantics in traditional energy-minimization methods [41, 18] for optical flow through geometric constraints such as piece-wise rigid motion and planar surface motion [57, 21, 1, 75]. In comparison, to the best of our knowledge, we are the first to inject semantics into the unsupervised training of recent optical flow networks. Some research also trains a network for segmentation and optical flow jointly [9]. In contrast, we leverage existing, separately trained segmentation systems both because they are available and because modularity—separating segmentation from flow estimation—is important for the development of large real-application systems such as autonomous driving.

In summary, our contributions are as follows.

- To the best of our knowledge, we are the first to explore adding semantic inputs to assist the unsupervised training of deep optical flow networks.

- We propose a simple yet effective network called SemARFlow that achieves state-of-the-art results both quantitatively and qualitatively. Our model works well on real-life occlusions and yields sharp motion boundaries around objects.

- We provide full training and inference code as well as trained models to encourage follow-up research.

2. Related work

Supervised optical flow Since the introduction of the pioneering deep optical flow network FlowNet [10], more and more top-performing CNN-based flow estimators have been proposed over the years [23, 49, 65, 20, 22, 68]. Recently, vision transformer networks and attention mechanism have also been applied to this problem, and these have achieved state-of-the-art performance on benchmark datasets [76, 26, 85, 64, 77]. The supervised methods are often pre-trained on large synthetic datasets such as FlyingChairs [10] and FlyingThings3D [44] before fine-tuning on the target dataset. However, there is a clear gap between the artificially generated data and real scenarios.

Unsupervised optical flow Due to the lack of ground-truth labels, unsupervised optical flow estimation uses surrogate losses such as photometric loss and smoothness loss to supervise training [81, 55]. To tackle the issues around occlusion regions, various methods have been proposed including estimated occlusion masks [72], forward-backward consistency [45], and multi-frame fusion [25, 53]. To better upsample the flow in the decoder, UPFlow [42] additionally predicts a confidence map and an interpolation flow to guide flow refinement and has become the state-of-the-art unsupervised flow method.

Some latest research has also explored the use of self-supervision to enhance flow prediction. Early methods apply the knowledge distillation technique to train a two-stage teacher-student network [38, 39]. ARFlow [37] further improves this idea by generating reliable self-supervision signals from data transformations, while merging the two training stages into single-stage training with one added loss term. SimFlow [24] replaces the handcrafted features with deep self-supervised features to measure similarity in the

unsupervised losses. SMURF [63] utilizes a RAFT-like structure [68] and applies multi-frame self-supervised training with many technical improvements. Our SemARFlow also uses self-supervision but with the guidance of semantic segmentation, which is much more realistic than guidance without semantics.

Semantic segmentation Semantic segmentation classifies each pixel of the given image into semantic objects. Fully Convolutional Network (FCN) [40] is one of the early CNN-based segmentation methods. It takes inputs of arbitrary sizes and outputs dense pixel-level predictions, becoming one of the popular backbone architectures for follow-up work. Deconvolution network [47] is also proposed to better recover low-level details of the prediction. One main challenge for these systems is that they lack global scene information. Subsequent work addresses this issue by enlarging the receptive field of the network with global pyramid pooling layers as in PSPNet [86], hybrid dilated convolutions [71], and a fast-down-sampling strategy [80, 79]. Attention modules have also shown to help in semantic segmentation by capturing full-image dependencies of all pixels [19, 87] or the semantic inter-dependencies across spatial and channel dimensions [12].

There also exists extensive work on semantic segmentation in the context of autonomous driving [2, 67, 13, 5, 89], thanks to the publication of large-scale driving datasets [8, 46]. To better train the network on a coarsely labeled dataset like KITTI [46], Zhu *et al*. use a video prediction network SDCNet [52] to synthesize new training samples with relaxed label propagation [89]. Due to their good performance on KITTI [46], we utilize their network models to infer semantic inputs for all our experiments.

Combining semantics and optical flow Though there has been much progress on both semantic segmentation and optical flow estimation, the semantic optical flow problem (how to exploit semantics to help optical flow estimation) has received limited attention in recent years, and the current best results are thus much outdated. Some early methods incorporate semantics through geometric constraints to refine flow on various semantic regions, such as planar regions (using homographies) [57], static regions (using rigid camera motion and epipolar constraints) [21, 75], and rigid objects (estimating rigid motion for each object instance) [1]. However, most methods are traditional flow methods based on energy minimization, where an initial flow estimate is usually needed and semantics is mostly used for refinement. In comparison, we explore adapting latest unsupervised optical flow networks to leverage semantic inputs in one single stage of estimation.

Apart from using semantics to help flow, some research has also explored using optical flow to help seman-

tic segmentation [51, 35], or to train both tasks jointly [9]. There are also studies on exploiting semantics on some other correspondence matching tasks such as stereo matching [74, 78] and 3D scene flow estimation, where some additional depth cues such as stereo camera inputs [54, 11] or point clouds [60] are needed.

3. Method

Our network is adapted from the two-frame version of ARFlow [37], which uses a light-weight PWCNet [65] as its backbone. The inputs are two consecutive frames $I_1, I_2 \in \mathbb{R}^{H \times W \times 3}$ as well as their semantic segmentation maps $S_1, S_2 \in \{0, 1, \cdots, c\}^{H \times W}$, where the number of classes c is 19 as we use the Cityscapes format [8]. A detailed diagram of the structure of our network can be found in the Appendix.

3.1. Semantic encoder

We first inject semantic information into the encoder. As shown in Fig. 2(a), shared by each frame $i \in \{1, 2\}$, separate convolutional layers extract features from image I_i and semantics S_i (one-hot encoded). Features from the two pipelines are concatenated and fed to additional layers. Features at different resolutions $(H/2^l, W/2^l)$ form a pyramid $\{f_i^{(l)} \mid 2 \leq l \leq 6\}$. The semantic information in these features helps delineate objects in dark shadows, where appearance is more homogeneous.

3.2. Iterative decoder with a learned upsampler

Following [22, 37], an iterative residual refinement decoder starts from zero estimate $F_{1 \to 2}^{(7)} = 0$. For iteration $l \in \{6, 5, 4, 3, 2\}$), the decoder refines feature map $F_{1 \to 2}^{(l+1)}$ into $F_{1 \to 2}^{(l)}$ based on features $f_1^{(l)}, f_2^{(l)}$ (Fig. 2(b)).

More specifically, $F_{1 \to 2}^{(l+1)}$ is upsampled to match the resolution and used to warp $f_2^{(l)}$ to yield warped feature $\hat{f}_1^{(l)}$. Correlation volumes are computed between $f_1^{(l)}$ and $\hat{f}_1^{(l)}$. A one-by-one convolutional layer $C^{(l)}$ compresses the number of channels to a fixed number so that the same layer can be reused across all iterations as proposed by [22]. A flow estimator network predicts a flow residual to be added to the current estimate, and a context network then aggregates flow information spatially and refines the current flow again.

A *learned* upsampler network upsamples the final output $F_{1 \to 2}^{(2)}$ in our system. This is different from ARFlow [37], which simply uses four-fold bilinear interpolation, making the final flow boundaries blurry. In contrast, our model learns to sharpen flow boundaries based on the semantic inputs, which have clear boundaries around moving objects. To this end, we add a convex upsampler network similar to the one in RAFT [68]. Different from UPFlow [42],

(a) shared pyramidal encoder ($i \in \{1, 2\}$)

(b) one iteration (at the lth-level) of the iterative decoder ($l \in \{6, 5, 4, 3, 2\}$)
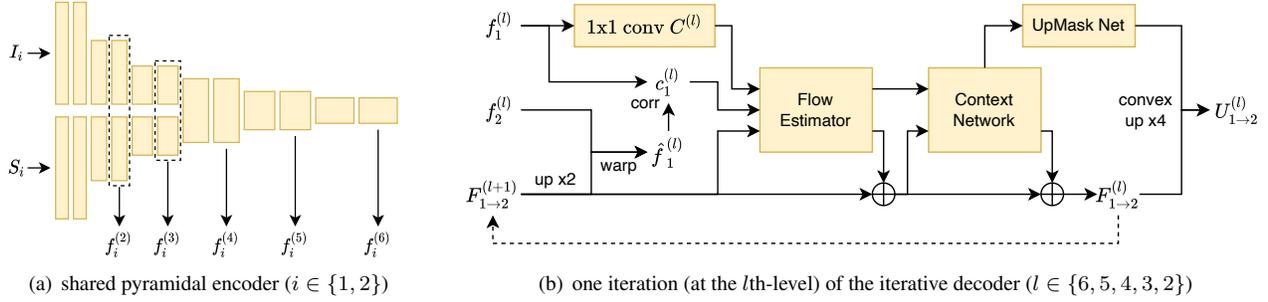
Figure 2. Network structure. See text in Sec. 3.1 and Sec. 3.2 for explanations. More detailed diagrams are in appendix.
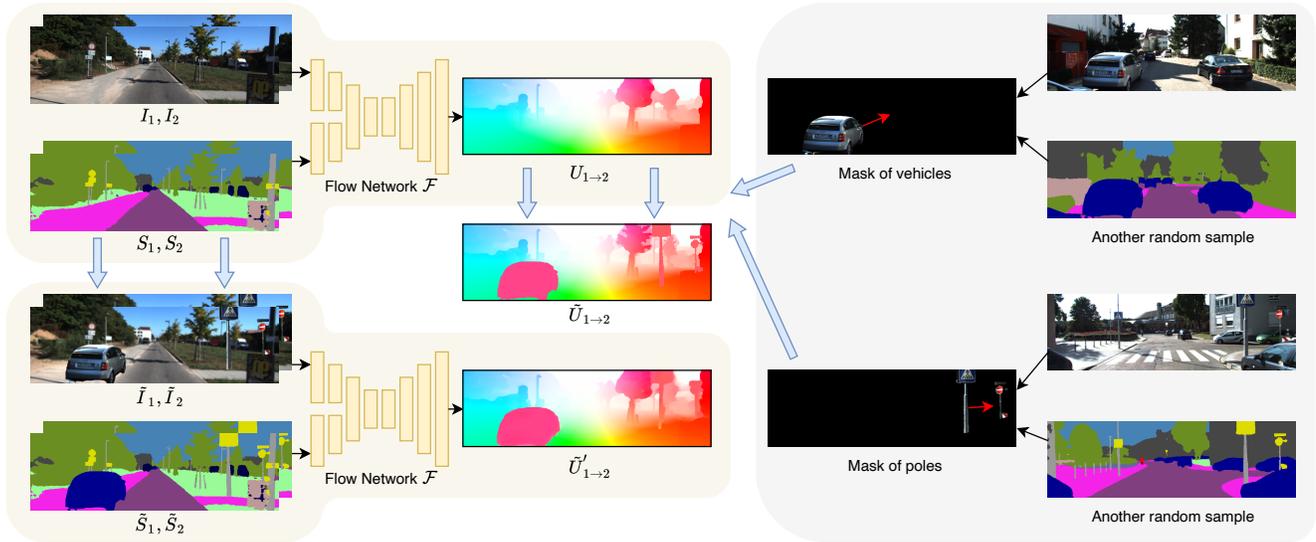

Figure 3. Illustration of semantic augmentation as self-supervision. See text in Sec. 3.3 for details.

our learned upsampler is only used to upsample from internal estimate $F_{1\to2}^{(l)}$ to output flow $U_{1\to2}^{(l)}$, but not when we upsample $F_{1\to2}^{(l+1)}$ at the first step in the decoder iteration. The upsampled $U_{1\to2}^{(l)}$ has resolution $(H/2^{l-2}, W/2^{l-2})$, so $U_{1\to2} = U_{1\to2}^{(2)}$ (with the original resolution) is the final flow prediction of the network.

3.3. Semantic augmentation as self-supervision

ARFlow has a very effective in-network augmentation module that samples random transformations $\mathcal{T}_{\theta_1}, \mathcal{T}_{\theta_2}$ of the flow prediction $U_{1\to2}$ in the first pass of the network and then uses the transformed images $\hat{I}_1 = \mathcal{T}_{\theta_1}(I_1), \hat{I}_2 = \mathcal{T}_{\theta_2}(I_2)$ in a second pass. The prediction $U_{1\to2}$ is also transformed accordingly and used to self-supervise the output of the second pass. See [37] for details.

We retain the augmentation module but make a third pass of the network using semantics-transformed inputs for self-supervision in addition to the ARFlow augmentations of ap-

pearance (*e.g.*, color jitter, random noise) and spatial transformations (*e.g.*, random rotation, random rescaling). The idea behind semantic augmentation is to blend in real object motions across samples.

To this end (see Fig. 3), we carve objects from other samples based on their semantic maps and paste them as moving foreground objects into the current sample I_1, I_2, thereby producing augmented images $\tilde{I}_1, \tilde{I}_2$. The segmentation maps S_1, S_2 and the first pass output flow $U_{1\to2}$ are transformed accordingly, and the transformed $\tilde{U}_{1\to2}$ self-supervises the third-pass network outputs $\tilde{U}'_{1\to2} = \mathcal{F}(\tilde{I}_1, \tilde{I}_2, \tilde{S}_1, \tilde{S}_2)$, where $\mathcal{F}$ is the flow network.

Similar to occlusion hallucination in SelFlow [39], our semantic augmentation also creates new occlusions and uses the reliable non-occluded flow from the first pass to self-supervise the flow on those newly occluded pixels. However, rather than superpixels filled with noise, we create occlusions with real objects, which are much more realistic.

Moreover, since we determine the motion of each oc-

cluder, we can use its known optical flow for motion self-supervision. As illustrated in Fig. 3, the occluder motion can be very different from that of the background. This trains the model to assume less smoothness around highly-dynamic objects and to output sharper motion estimates near motion boundaries. These hallucinated but realistic occluders also provide more examples of actively moving objects, which are relatively scarce in the original datasets.

Vehicles We augment vehicle classes (car, truck, bus, train) as they are major sources of errors in previous models (see Tab. 5 for statistics). We group these four classes because they are easily confused with each other in the semantic input. To carve out a single car instance with high probability, we find connected components of vehicle segmentation regions using OpenCV [3] and accept masks whose bounding boxes are 50-300 pixels wide and 50-150 pixels high. We also drop masks that occupy less than 60% area of their bounding boxes to remove largely occluded vehicles.

Poles We augment pole classes (pole, traffic light, traffic sign) not only because they are common foreground objects that cause occlusions, but also because their motions are usually poorly estimated due to their thin shapes. We use a 200-pixel wide, image-height sliding window to scan the segmentation of these pole classes and find the window with the most pole pixels. If more than 10% pole pixels occur in it, that object mask is stored. This produces groups of nearby poles rather than just individual poles, which may not provide enough augmentation by themselves.

Sky We also augment the sky through prior knowledge that flow in the sky should be very small. We shrink the sky flow estimate by half before using it for self-supervision. As shown in Fig. 3, the middle tree part in $U_{1\to2}$ is very blurry, with a large part of sky mixed in there. After shrinking sky flow by half, the tree in $\tilde{U}_{1\to2}$ now has sharper flow around the boundary between tree and sky. Using this as self-supervision helps the model sharpen the motion boundaries between sky and other regions.

Another way to apply this prior knowledge is to post-process, *i.e.*, shrink the final *output* flow in the sky region, instead of using the shrunk flow to self-supervise. However, the latter approach is preferable because it allows the network to balance prior knowledge (self-supervision) with the current sample observation (photometric loss). Our experiments do show cases where some small objects (such as some overhead electric power lines) are misclassified as part of the sky in the semantic input. In these cases, the network learns to rely less on prior knowledge.

3.4. Occluder cache

During training, we maintain a cache of occluding objects for semantic augmentation. We sample objects from the cache to augment the current batch and then push any new objects found in the batch into the cache, with random replacement if the cache is full.

Specifically, after we load a sample for training, we first use the current model to infer its optical flow and then search within the sample for any cars (or poles) that meet the standards defined in Sec. 3.3 as "cutouts". For each cutout, we first find the flow vectors on all pixels that belong to that cutout and then compute the mean flow for later reference. We store the current sample, the cutout mask, and the mean flow together in the cache.

Later on, when we randomly select a cutout to augment another sample, we also retrieve the stored mean flow of this cutout, which is then augmented by random rescaling (by 0.8-1.5 times) and reversing (with 50% probability). We use the augmented mean flow to translate the entire cutout object as an occluder to generate a new sample. For realism, occluders are pasted to the same location they occupied in their image of origin. Holding occluders in a cache provides a random mixture of samples across multiple batches and makes semantic augmentation very efficient.

3.5. Loss functions

Photometric loss In the first pass, forward and backward flow $U_{1\to2}^{(l)}, U_{2\to1}^{(l)}$ at each level are predicted and occlusion masks $O_{1\to2}^{(l)}, O_{2\to1}^{(l)}$ are computed by forward-backward consistency [45]. Frames are warped by

$$I_i'^{(l)}(\boldsymbol{p}) = I_j^{(l)}(\boldsymbol{p} + U_{i\to j}^{(l)}(\boldsymbol{p})), \quad (i,j \in \{1,2\})$$

where $I_i^{(l)}$ is I_i down-sampled to the l-th scale, and $\boldsymbol{p}$ denotes pixel coordinates at that scale. Following [37], three measures, namely L$_1$-distance (ρ_1), structural similarity (SSIM) [73] (ρ_2), and census loss [45] (ρ_3) are linearly combined to measure photometric differences between $I_i^{(l)}$ and $I_i'^{(l)}$. Occlusion regions are masked out and both forward and backward directions are taken into account at each level. The final photometric loss is

$$\ell_{\text{ph}} = \frac{1}{2} \sum_{(i,j)\in\{(1,2),(2,1)\}} \sum_{l=2}^{6} a_l \sum_{k=1}^{3} c_k \rho_k(I_i^{(l)}, I_i'^{(l)}, O_{i\to j}^{(l)}). \tag{1}$$

Smoothness loss Unlike most previous work, we do not include a smoothness loss because we find it to conflict with our learned upsampler. See ablation study in Sec. 4.5.

Augmentation loss As in ARFlow [37], our second forward pass computes the L$_1$-distance between the trans-

	Method	Train		Test					
		2012	2015	2012		2015			
		EPE	EPE	Fl-noc	EPE	Fl-all	Fl-noc	Fl-bg	Fl-fg
supervised	PWC-Net+ [66]	-	(1.50)	3.36	1.4	7.72	4.91	7.69	7.88
	IRR-PWC [22]	-	(1.63)	3.21	1.6	7.65	4.86	7.68	7.52
	RAFT [68]	-	(0.63)	-	-	5.10	3.07	4.74	6.87
	Separable Flow [85]	-	(0.69)	-	-	4.53	2.78	4.25	5.92
unsupervised	SelFlow [39]	1.69	4.84	4.31	2.2	14.19	9.65	12.68	21.74
	SimFlow [24]	-	5.19	-	-	13.38	8.21	12.60	17.27
	ARFlow [37]	1.44	2.85	-	1.8	11.80	-	-	-
	UFlow [27]	1.68	2.71	4.26	1.9	11.13	8.41	9.78	17.87
	UPFlow [42]	**1.27**	2.45	-	**1.4**	9.38	-	-	-
	Ours (baseline)	1.39	2.61	4.30	1.7	9.89	6.98	8.82	15.21
	Ours (+enc)†	1.29	2.42	3.97	1.5	8.99	3.97	8.19	13.01
	Ours (+enc +aug)†	1.28	**2.18**	**3.90**	1.5	**8.38**	**3.90**	**7.48**	**12.91**

Table 1. KITTI benchmark results (EPE/px and Fl/%). Metrics evaluated at 'all' (all pixels, default for EPE), 'noc' (non-occlusions), 'bg' (background), and 'fg' (foreground). Key metrics (used in official ranking) are underlined. Our 'baseline' is an adapted ARFlow with added learned upsampler and no smoothness loss. '+enc' means adding semantic encoder; '+aug' means adding semantic augmentation. '()' means evaluation data used in training. '-' means unavailable. † denotes models with semantic inputs. For all metrics, lower is better.

formed flow $\hat{U}_{1\to2}$ and the second pass output $\hat{U}'_{1\to2} = \mathcal{F}(\hat{I}_1, \hat{I}_2, \hat{S}_1, \hat{S}_2)$ for each pixel $\boldsymbol{p}$ as

$$\hat{D}(\boldsymbol{p}) = \|\hat{U}_{1\to2}(\boldsymbol{p}) - \hat{U}'_{1\to2}(\boldsymbol{p})\|_1.$$

This loss is then averaged over the transformed non-occluded region where self-supervision $\hat{U}_{1\to2}$ is accurate

$$\ell_{\mathrm{ar}} = \frac{\sum_{\boldsymbol{p}}(1 - \hat{O}_{1\to2}(\boldsymbol{p}))\hat{D}(\boldsymbol{p})}{\sum_{\boldsymbol{p}}(1 - \hat{O}_{1\to2}(\boldsymbol{p}))}. \quad (2)$$

For semantic augmentation (third pass), since we are also doing self-supervision with a new generated sample, we use a similar loss definition

$$\ell_{\mathrm{aug}} = \frac{\sum_{\boldsymbol{p}}(1 - \tilde{O}_{1\to2}(\boldsymbol{p}))\tilde{D}(\boldsymbol{p})}{\sum_{\boldsymbol{p}}(1 - \tilde{O}_{1\to2}(\boldsymbol{p}))}, \quad (3)$$

where $\tilde{D}(\boldsymbol{p})$ is the distance between the semantic-augmented flow $\tilde{U}_{1\to2}$ and the third-pass output $\tilde{U}'_{1\to2}$

$$\tilde{D}(\boldsymbol{p}) = \|\tilde{U}_{1\to2}(\boldsymbol{p}) - \tilde{U}'_{1\to2}(\boldsymbol{p})\|_1,$$

and the augmented mask $\tilde{O}_{1\to2}$ is now computed by

$$\tilde{O}_{1\to2} = 1 - \max(1 - O_{1\to2}, M)$$

because we penalize on both the originally non-occluded region $1 - O_{1\to2}$ and the pasted foreground object mask M (of which we know the true motion).

Final loss (with $\lambda = 0.02$ as in ARFlow [37]):

$$\ell = \ell_{\mathrm{ph}} + \lambda(\ell_{\mathrm{ar}} + \ell_{\mathrm{aug}}). \quad (4)$$

4. Experiments

4.1. Datasets

We mainly use KITTI [15, 46] and Cityscapes [8] datasests for experiments. Following [37], we train our model first on KITTI raw sequences [46] (55.7k samples) and then fine-tune on KITTI-2015 multi-view extension [46] (11.8k samples). We validate our model using KITTI-2015 train [46] (200 samples) and KITTI-2012 train set [15] (194 samples) since they are the only sets that have optical flow labels. We also train on Cityscapes [8] sequences (83.3k samples) to test model generalization ability, and we sample every other frame to match its frame rate with KITTI. All semantic segmentation inputs are estimated using an off-the-shelf model [89] with a DeepLabV3Plus [7] backbone, which achieves 83.45% mean IoU on Cityscapes and 72.82% mean IoU on KITTI.

4.2. Implementation details

We implement the model in PyTorch [48] [1]. We use the Adam optimizer [32] with $\beta_1 = 0.9, \beta_2 = 0.999$ and batch size 4. We first train on KITTI raw sequences [46] for 100k iterations with a fixed learning rate 0.0002 and then train on KITTI multi-view extension set [15, 46] for another 100k iterations using OneCycleLR schedule [62] with maximum learning rate 0.0004 and linear annealing.

For data augmentation, we include random horizontal flipping and swapping of the input frames. We resize the inputs to 256×832 before feeding into the network. The photometric loss weight for each scale a_l $(2 \leq l \leq 6)$ in

---

[1]Code and instructions are available at https://github.com/duke-vision/semantic-unsup-flow-release.

Method	Fl-all	Fl-noc	Fl-bg	Fl-fg
JFS [21]	17.07	9.81	15.9	22.92
SOF [57]	16.81	10.86	14.63	27.73
MRFlow [75]	12.19	8.86	10.13	22.52
Bai *et al.* [1]	11.62	8.75	8.61	26.69
Ours (final)	**8.38**	**3.90**	**7.48**	**12.91**

Table 2. KITTI-2015 test results (Fl/%) compared with other semantic optical flow methods. Metrics evaluated at 'all' (all pixels), 'noc' (non-occlusions), 'bg' (background), and 'fg' (foreground).

Eq. (1) are set as 1, 1, 1, 1, 0. The weights for three photometric distance measures ρ_k ($1 \leq k \leq 3$) in Eq. (1) are set as 0.15, 0.85, 0 for the first 50k iterations and 0, 0, 1 afterwards. We start the appearance and spatial augmentation (second pass as in ARFlow [37]) at 50k iterations, and semantic augmentation (third pass) after 150k iterations.

4.3. Benchmark testing results

As common practice, we evaluate optical flow predictions based on two error measurements, Fl (error rate) and EPE (mean L_2 distance). When computing Fl, the estimate of each pixel is considered correct if the error is smaller than 3 pixels or 5% of the magnitude of ground-truth flow [46].

As shown in Tab. 1, our semantic modules (both semantic encoder and semantic augmentation) improve performance on both KITTI-2012 and KITTI-2015 sets on all metrics. Our final model achieves 8.38% Fl-all error rate on KITTI-2015 test set, which is significantly better than ARFlow [37] (11.80%), from which we adapt. We also outperform the current state-of-the-art UPFlow [42] (9.38%) by a clear margin. All these results strongly suggest that adding semantic inputs can help improve unsupervised flow estimation significantly.

In addition, Tab. 2 shows that our test results significantly outperform all previous semantic optical flow methods, most of which are based on traditional energy minimization. We are the first to apply semantic inputs to recent unsupervised flow networks, so we are able to push the state-of-the-art by a clear margin.

4.4. Qualitative results

Some qualitative results are shown in Fig. 4. We can see that our flow outputs have very sharp boundaries and very clean object motion, especially around cars and poles. Our model successfully learns to adapt flow estimation to the semantic map input. Moreover, our model is able to handle very challenging samples where the foreground car motion is drastically different from the background due to the challenging samples we create in semantic augmentation.

up	no sm	enc	aug	Fl-all	EPE-all	EPE-noc	EPE-occ
				10.36	2.90	2.07	6.97
		1		9.92	2.69	1.89	6.57
		2		9.83	2.65	1.86	6.40
		3		9.75	2.61	1.85	6.54
		4		9.75	2.64	1.85	6.55
✓				10.22	2.83	1.97	6.75
✓	✓			8.87	2.61	1.85	6.04
✓	✓	3		8.26	2.42	1.73	5.79
✓	✓		✓	8.80	2.48	1.60	6.64
✓	✓	3	✓	**7.79**	**2.18**	**1.40**	**5.64**

Table 3. Ablation study on KITTI-2015 train set (EPE/px and Fl/%). 'up' means the learned upsampler; 'no sm' means turning off smoothness loss; 'enc' means the level at which we merge semantic features and image features in the encoder; 'aug' means semantic augmentation.

Options of aug	Fl-all	EPE-all	EPE-noc	EPE-occ
Ours (final)	**7.79**	**2.18**	**1.40**	5.64
start from 100k	7.92	2.19	**1.40**	**5.41**
vehicles only	7.94	2.21	1.44	5.78
loss on new occ	8.15	2.27	1.42	5.62

Table 4. Ablation study of different semantic augmentation options on KITTI-2015 train (EPE/px and Fl/%). See text for explanations.

4.5. Ablation study

We also do ablation study to show the effectiveness of each of our proposed modules. We can see in Tab. 3 that adding a semantic encoder to the vanilla ARFlow does help, and the optimal number of encoder layer groups added is 3. Also, our learned upsampler improves results significantly once we turn off the smoothness loss. This makes sense because the upsampler learns to adapt to boundaries, which can be distracted by the smoothness loss. Moreover, both our semantic encoder and semantic augmentation module help improve the results further, which suggests the effectiveness of adding semantic inputs.

Another ablation study is on different options for the semantic augmentation (Tab. 4). We try starting augmentation earlier from 100k iterations, which works slightly worse. There are mainly two reasons: (1) we use the first forward output to self-supervise our augmented output, so if we start early, the model will use poor output to self-supervise the augmented pass; and (2) our semantic augmentation creates very challenging samples with objects moving very differently from the background, so exposing these hard samples to the model too early may make it hard to train. Apart from this, we also try only using vehicles to augment as well as focusing loss on the newly occluded region, which are both inferior to our final version.
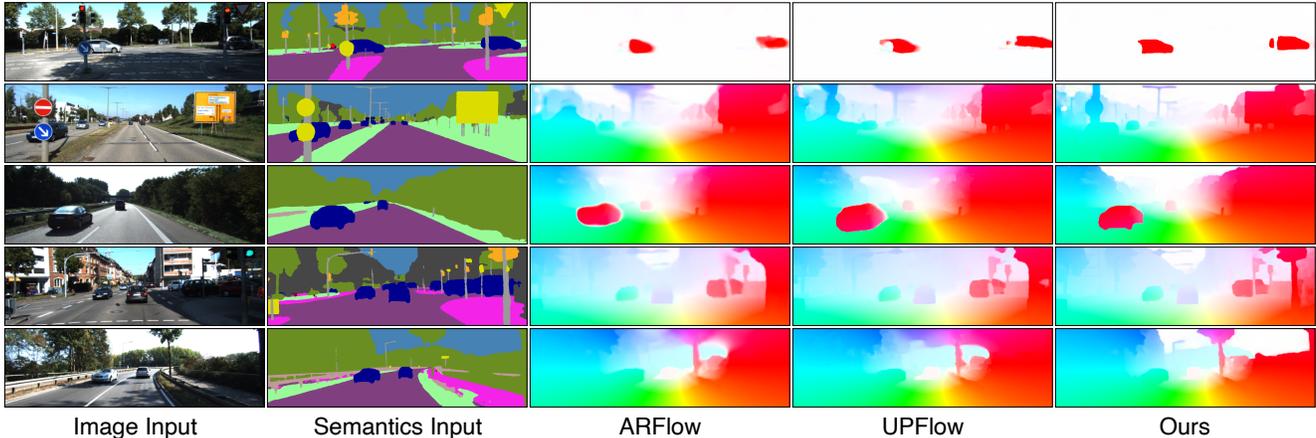
Figure 4. Qualitative results on KITTI test set (sample #7, 20, 38, 112, 183) compared with ARFlow [37] and UPFlow [42]. We only show first frame input for conciseness. Our outputs have much clearer boundaries around objects. For more qualitative examples, see Appendix.

	road	car	terrain	vegetation	sidewalk	building	wall	pole
Proportion	42.4%	17.6%	14.0%	11.6%	6.2%	4.1%	1.3%	1.1%
ARFlow [37]	4.57	15.79	9.42	15.34	4.61	6.00	16.34	10.75
Ours (final)	3.67	10.17	8.74	13.47	3.09	4.27	11.18	9.44
Relative improvement	19.6%	35.6%	7.3%	12.2%	33.0%	28.7%	31.6%	12.2%
Reweighed contribution	19.0%	49.4%	4.8%	10.9%	4.7%	3.5%	3.4%	0.7%

Table 5. KITTI-2015 flow error (Fl-all/%) for each semantic class. The first row shows the proportion of each class among evaluated pixels (KITTI flow does not evaluate on every pixel). We only show classes that account for at least 1% of the evaluated pixels.

4.6. Improvement analysis

It may be interesting to understand where our improvements come from in terms of semantic classes, so we compute the KITTI-2015 train set error for each class. As shown in Tab. 5, our model improves the flow on every class. After reweighing the absolute improvement of each class based on their proportion in the evaluated pixels, we find that car flow accounts for nearly half of our improvement on Fl-all, indicating the effectiveness of augmenting vehicle objects.

4.7. Generalization ability

We also investigate the generalization ability of our semantic-aided flow models. We train flow models on Cityscapes and directly test them on KITTI-2015 train set without fine-tuning. Following [50], we crop the bottom 25% of the frame to remove the car logo and resize the frame to size 256×704. All other experiment settings are exactly the same as for KITTI. As we can see from Tab. 6, adding semantic inputs significantly helps our unsupervised flow model generalize and adapt better across datasets.

5. Conclusion and future work

In this paper, we show that adding semantic segmentation inputs can help significantly improve the performance

Method	Fl-all	EPE-all	EPE-noc	EPE-occ
ARFlow (our impl.)	13.21	4.08	2.88	9.40
Ours (baseline)	12.27	3.81	2.43	9.91
Ours (+enc)†	11.28	3.33	2.12	8.75
Ours (+enc +aug)†	**10.32**	**2.64**	**1.56**	**7.12**

Table 6. Generalization results (train on Cityscapes, and test on KITTI-2015 train). We use our implementation of ARFlow [37] to run this experiment. Our 'baseline' is an adapted ARFlow with added learned upsampler and no smoothness loss. '+enc' means adding semantic encoder; '+aug' means adding semantic augmentation. † denotes models with semantic inputs.

of unsupervised optical flow networks on autonomous driving datasets. We propose a novel network model called SemARFlow, a semantic adaptation of ARFlow [37], with a semantic encoder, a learned upsampler, and a semantic augmentation module, where some domain-specific motion prior has been taken into account. Our network better predicts flow at occlusion regions and effectively sharpens flow estimates around object boundaries, even for very challenging samples. The additional semantic inputs also make our network generalize better across datasets.

One direction for future work is motivated by the example illustrated in Fig. 5. Our model does output sharp flow
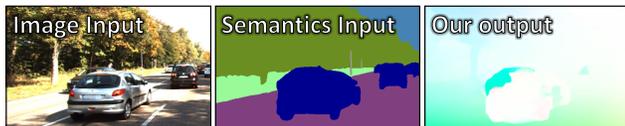
Figure 5. An example on KITTI-2015 train set (sample #164). See the discussion in Sec. 5 for details.

boundaries between the cars and the background, thanks to good boundary information from the semantic maps. However, the boundary between these two fast-moving cars is inferred from image information only, because their semantic maps merge into one. As a result, that part of the flow boundary is less accurate. An interesting direction for future work is to use instance-level semantics as input for more detailed object masks.

References

[1] Min Bai, Wenjie Luo, Kaustav Kundu, and Raquel Urtasun. Exploiting semantic information and deep matching for optical flow. In *Proceedings of the European Conference on Computer Vision*, pages 154–170. Springer, 2016. 2, 3, 7

[2] Shubhankar Borse, Ying Wang, Yizhe Zhang, and Fatih Porikli. Inverseform: A loss function for structured boundary-aware segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5901–5911, 2021. 3

[3] Gary Bradski. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000. 5

[4] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *Proceedings of the European Conference on Computer Vision*, pages 611–625. Springer, 2012. 1

[5] Yingfeng Cai, Lei Dai, Hai Wang, and Zhixiong Li. Multi-target pan-class intrinsic relevance driven model for improving semantic segmentation in autonomous driving. *IEEE Transactions on Image Processing*, 30:9069–9084, 2021. 3

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

[7] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 801–818, 2018. 6

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 3, 6

[9] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10713–10720, 2020. 2, 3

[10] Alexey Dosovitskiy, Philipp Fischery, 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 *Proceedings of the IEEE International Conference on Computer Vision*, ICCV '15, pages 2758–2766, Washington, DC, USA, 2015. IEEE Computer Society. 1, 2

[11] Cheng Feng, Long Ma, Congxuan Zhang, Zhen Chen, Liyue Ge, and Shaofeng Jiang. Ss-sf: Piecewise 3d scene flow estimation with semantic segmentation. *IEEE Access*, 9:22745–22759, 2021. 3

[12] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019. 3

[13] Aditya Ganeshan, Alexis Vallet, Yasunori Kudo, Shin-ichi Maeda, Tommi Kerola, Rares Ambrus, Dennis Park, and Adrien Gaidon. Warp-refine propagation: Semi-supervised auto-labeling via cycle-consistency. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 15499–15509, 2021. 3

[14] Chen Gao, Ayush Saraf, Jia-Bin Huang, and Johannes Kopf. Flow-edge guided video completion. In *Proceedings of the European Conference on Computer Vision*, pages 713–729. Springer, 2020. 1

[15] 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. 6

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

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1

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

[19] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 603–612, 2019. 3

[20] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *Proceedings of the IEEE Conference*

on Computer Vision and Pattern Recognition, pages 8981–8989, 2018. 2

[21] Junhwa Hur and Stefan Roth. Joint optical flow and temporally consistent semantic segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 163–177. Springer, 2016. 2, 3, 7

[22] Junhwa Hur and Stefan Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5754–5763, 2019. 1, 2, 3, 6

[23] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1647–1655, 2017. 2

[24] Woobin Im, Tae-Kyun Kim, and Sung-Eui Yoon. Unsupervised learning of optical flow with deep feature similarity. In *Proceedings of the European Conference on Computer Vision*, pages 172–188. Springer, 2020. 2, 6

[25] Joel Janai, Fatma Guney, Anurag Ranjan, Michael Black, and Andreas Geiger. Unsupervised learning of multi-frame optical flow with occlusions. In *Proceedings of the European Conference on Computer Vision*, pages 690–706, 2018. 2

[26] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9772–9781, 2021. 2

[27] Rico Jonschkowski, Austin Stone, Jonathan T Barron, Ariel Gordon, Kurt Konolige, and Anelia Angelova. What matters in unsupervised optical flow. In *Proceedings of the European Conference on Computer Vision*, pages 557–572. Springer, 2020. 1, 6

[28] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys*, 54(10s):1–41, 2022. 1

[29] Hannah H Kim, Celia Cintas, Girmaw Abebe Tadesse, and Skyler Speakman. Spatially constrained adversarial attack detection and localization in the representation space of optical flow networks. In *International Joint Conference on Artificial Intelligence*, 2023. 1

[30] Hannah Halin Kim, Shuzhi Yu, and Carlo Tomasi. Joint detection of motion boundaries and occlusions. In *Proceedings of the British Machine Vision Conference*, 2021. 1

[31] Hannah Halin Kim, Shuzhi Yu, Shuai Yuan, and Carlo Tomasi. Cross-attention transformer for video interpolation. In *Proceedings of the Asian Conference on Computer Vision Workshops*, pages 320–337, 2022. 1

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Proceedings of the International Conference on Learning Representations*, 2014. 6

[33] Alexander Kolesnikov, Alexey Dosovitskiy, Dirk Weissenborn, Georg Heigold, Jakob Uszkoreit, Lucas Beyer, Matthias Minderer, Mostafa Dehghani, Neil Houlsby, Sylvain Gelly, Thomas Unterthiner, and Xiaohua Zhai. An image is worth 16x16 words: Transformers for image recogni-

tion at scale. In *Proceedings of the International Conference on Learning Representations*, 2021. 1

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 1

[35] Xiangtai Li, Jiangang Bai, Kuiyuan Yang, and Yunhai Tong. Flow2seg: Motion-aided semantic segmentation. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 225–237. Springer, 2019. 3

[36] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, 2021. 1

[37] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6489–6498, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[38] Pengpeng Liu, Irwin King, Michael R Lyu, and Jia Xu. Ddflow: Learning optical flow with unlabeled data distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8770–8777, 2019. 1, 2

[39] Pengpeng Liu, Michael Lyu, Irwin King, and Jia Xu. Selflow: Self-supervised learning of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4571–4580, 2019. 1, 2, 4, 6

[40] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 2, 3

[41] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, volume 2, pages 674–679, 1981. 2

[42] Kunming Luo, Chuan Wang, Shuaicheng Liu, Haoqiang Fan, Jue Wang, and Jian Sun. Upflow: Upsampling pyramid for unsupervised optical flow learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1045–1054, 2021. 1, 2, 3, 6, 7, 8

[43] Rémi Marsal, Florian Chabot, Angélique Loesch, and Hichem Sahbi. Brightflow: Brightness-change-aware unsupervised learning of optical flow. In *Proceedings or the IEEE winter conference on applications of computer vision*, pages 2061–2070, 2023. 1

[44] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4040–4048, 2016. 1, 2

[45] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 1, 2, 5

[46] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Proceedings of the IEEE Conference*

on Computer Vision and Pattern Recognition, 2015. 2, 3, 6, 7

[47] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015. 2, 3

[48] 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 *Advances in Neural Information Processing Systems*, pages 8024–8035. Curran Associates, Inc., 2019. 6

[49] Anurag Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2720–2729, 2017. 2

[50] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12240–12249, 2019. 8

[51] Hazem Rashed, Senthil Yogamani, Ahmad El-Sallab, Pavel Krizek, and Mohamed El-Helw. Optical flow augmented semantic segmentation networks for automated driving. In *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 165–172, 2019. 3

[52] Fitsum A Reda, Guilin Liu, Kevin J Shih, Robert Kirby, Jon Barker, David Tarjan, Andrew Tao, and Bryan Catanzaro. Sdc-net: Video prediction using spatially-displaced convolution. In *Proceedings of the European Conference on Computer Vision*, pages 718–733, 2018. 3

[53] Zhile Ren, Orazio Gallo, Deqing Sun, Ming-Hsuan Yang, Erik B Sudderth, and Jan Kautz. A fusion approach for multi-frame optical flow estimation. In *Proceedings or the IEEE winter conference on applications of computer vision*, pages 2077–2086. IEEE, 2019. 2

[54] Zhile Ren, Deqing Sun, Jan Kautz, and Erik Sudderth. Cascaded scene flow prediction using semantic segmentation. In *Proceedings of the International Conference on 3D Vision*, pages 225–233. IEEE, 2017. 3

[55] Zhe Ren, Junchi Yan, Bingbing Ni, Bin Liu, Xiaokang Yang, and Hongyuan Zha. Unsupervised deep learning for optical flow estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017. 2

[56] Stephan R. Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2232–2241, 2017. 1

[57] Laura Sevilla-Lara, Deqing Sun, Varun Jampani, and Michael J Black. Optical flow with semantic segmentation and localized layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3889–3898, 2016. 2, 3, 7

[58] Shihao Shen, Louis Kerofsky, and Senthil Yogamani. Optical flow for autonomous driving: Applications, challenges and improvements. *arXiv preprint arXiv:2301.04422*, 2023. 1

[59] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994. 1

[60] Yukang Shi and Kaisheng Ma. Safit: Segmentation-aware scene flow with improved transformer. In *Proceedings of the International Conference on Robotics and Automation*, pages 10648–10655. IEEE, 2022. 3

[61] Jeongho Shin, Sangjin Kim, Sangkyu Kang, Seong-Won Lee, Joonki Paik, Besma Abidi, and Mongi Abidi. Optical flow-based real-time object tracking using non-prior training active feature model. *Real-time imaging*, 11(3):204–218, 2005. 1

[62] 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*, volume 11006, pages 369–386. SPIE, 2019. 6

[63] Austin Stone, Daniel Maurer, Alper Ayvaci, Anelia Angelova, and Rico Jonschkowski. Smurf: Self-teaching multi-frame unsupervised raft with full-image warping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3887–3896, 2021. 3

[64] Xiuchao Sui, Shaohua Li, Xue Geng, Yan Wu, Xinxing Xu, Yong Liu, Rick Goh, and Hongyuan Zhu. Craft: Cross-attentional flow transformer for robust optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17602–17611, 2022. 2

[65] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 1, 2, 3

[66] 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 transactions on pattern analysis and machine intelligence*, 42(6):1408–1423, 2019. 6

[67] Andrew Tao, Karan Sapra, and Bryan Catanzaro. Hierarchical multi-scale attention for semantic segmentation. *arXiv preprint arXiv:2005.10821*, 2020. 3

[68] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision*, pages 402–419. Springer, 2020. 1, 2, 3, 6

[69] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 1

[70] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE Con-*

ference on Computer Vision and Pattern Recognition Workshops, pages 969–977, 2018. 1

[71] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. Understanding convolution for semantic segmentation. In *Proceedings or the IEEE winter conference on applications of computer vision*, pages 1451–1460. Ieee, 2018. 3

[72] Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, Peng Wang, and Wei Xu. Occlusion aware unsupervised learning of optical flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4884–4893, 2018. 1, 2

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

[74] Zhenyao Wu, Xinyi Wu, Xiaoping Zhang, Song Wang, and Lili Ju. Semantic stereo matching with pyramid cost volumes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7484–7493, 2019. 3

[75] Jonas Wulff, Laura Sevilla-Lara, and Michael J Black. Optical flow in mostly rigid scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4671–4680, 2017. 2, 3, 7

[76] Haofei Xu, Jiaolong Yang, Jianfei Cai, Juyong Zhang, and Xin Tong. High-resolution optical flow from 1d attention and correlation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10498–10507, 2021. 2

[77] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, and Dacheng Tao. Gmflow: Learning optical flow via global matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8121–8130, 2022. 2

[78] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. Segstereo: Exploiting semantic information for disparity estimation. In *Proceedings of the European Conference on Computer Vision*, pages 636–651, 2018. 3

[79] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129:3051–3068, 2021. 3

[80] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 325–341, 2018. 3

[81] 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 Gang Hua and Hervé Jégou, editors, *Proceedings of the European Conference on Computer Vision Workshops*, pages 3–10, Cham, 2016. Springer International Publishing. 1, 2

[82] Shuzhi Yu, Hannah H Kim, Shuai Yuan, and Carlo Tomasi. Unsupervised flow refinement near motion boundaries. In

[83] Shuai Yuan, Xian Sun, Hannah Kim, Shuzhi Yu, and Carlo Tomasi. Optical flow training under limited label budget via active learning. In *Proceedings of the European Conference on Computer Vision*, pages 410–427, 2022. 2

[84] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020. 1

[85] Feihu Zhang, Oliver J Woodford, Victor Adrian Prisacariu, and Philip HS Torr. Separable flow: Learning motion cost volumes for optical flow estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10807–10817, 2021. 1, 2, 6

[86] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2881–2890, 2017. 3

[87] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6881–6890, 2021. 2, 3

[88] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *Proceedings of the International Conference on Learning Representations*, 2021. 1

[89] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8856–8865, 2019. 2, 3, 6

Proceedings of the British Machine Vision Conference. BMVA Press, 2022. 1