SUPPLEMENTARY MATERIAL FOR

ScopeFlow: Dynamic Scene Scoping for Optical Flow

A. Introduction

With this supplementary package, we would like to provide more details on our training pipeline and framework, as well as more visualizations of the improved flow and occlusion estimation.

The ScopeFlow approach provides an improved training pipeline for optical flow models, which reduces the bias in sampling different regions of the input images while keeping the power of the regularization provided by fixed-size partial crops. Due to the sizable impact on performance that can be achieved by the improved training pipeline, we created a generic, easy to configure, training package, in order to encourage others to train state of the art models with our improved pipeline, as described in Sec. C.

B. Dynamic scoping

The common pipeline of batch sampling and augmentation in optical flow training includes four stages: (i) sampling images, (ii) applying random photometric changes, (iii) applying a random affine transformation, and (iv) cropping a fixed-size randomly located patch. We propose changes for stages (iii) and (iv), by choosing the zooming parameters more carefully along with the training, and incorporating a new randomized cropping scheme, presented and extensively tested in our paper.

Fig. 1 provides a demonstration of the ScopeFlow pipeline, which enlarges the variety of scopes presented during the data-driven process while reducing the bias towards specific categories.

C. ScopeFlow software package

In order to simplify the applicability of our approach, we created a small and easy to use package, which supports YAML configurations of a multi-stage optical flow model training and evaluation. We found this approach very helpful when running experiments for finding the best scoping augmentation approach.

Our code is attached to this supplemental, and our models (too large to be included within the limits of the supplementary material) would be made public. Furthermore,

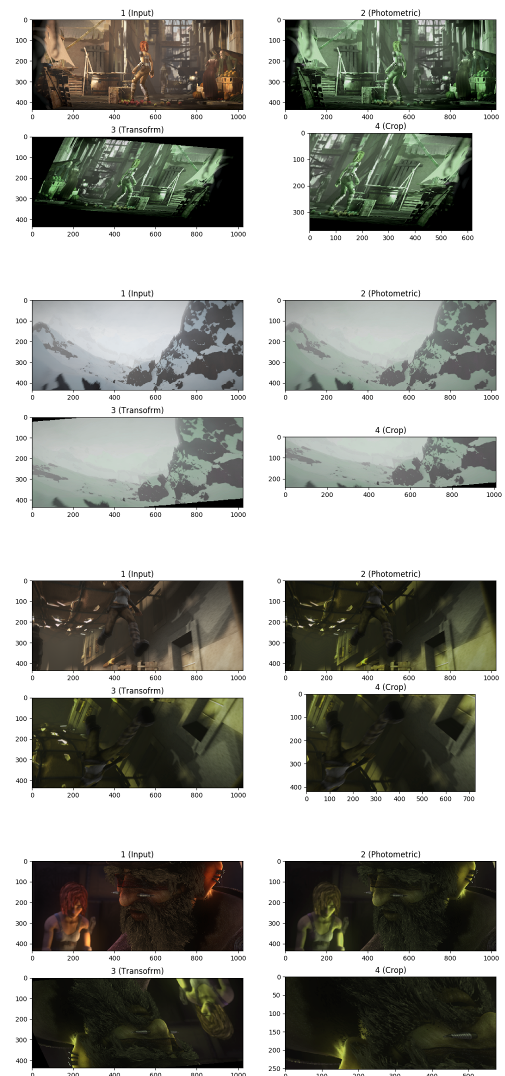

Figure 1. Randomized scoping within $[r_{min}, r_{max}] = [0.5, 1]$. Training with ScopeFlow online-processing approach leads to the learning of richer features and reduces the error in challenging motion categories, such as fast speed and occluded pixels.

we provide easy visualization of our online augmentation pipeline, as described in the README of our package.

D. Comparison to the IRR baseline

In our experiments, we use the IRR [3] variant, of the popular PWC-Net architecture, to evaluate our method. This variant has shown to provide excellent results, while keeping a low number of parameters. To emphasize the improvements, we give here a thorough comparison, of all the public results obtained in the main three benchmarks, for our method and the IRR baseline.

D.1. MPI Sintel

Other than leading the MPI Sintel [1] table, as can be seen in Tab. 6 and Tab. 7 in Sec. G, we improve the baseline IRR models by a large margin in all metrics, and in particular the challenging metrics of occlusions (14.7%) and fast pixels (18.4%). The only metric that did not improve is the metrics of low-speed pixels (< 40), which should not be a surprise, since our method reduces the bias between the fast and slow pixels, as shown in our paper.

D.2. KITTI 2012

We uploaded our results to the KITTI 2012 [2] benchmark. As can be seen in Tab. 1 and Tab. 2, training the IRR model with ScopeFlow pipeline improves the mean EPE by more than 20%. Moreover, the improvement is achieved for all thresholds of outliers and for all metrics.

IRR on KITTI 2012:

Error	Out-Noc	Out-All	Avg-Noc	Avg-All
2 pixels	5.34 %	9.81 %	0.9 px	1.6 px
3 pixels	3.21 %	6.70 %	0.9 px	1.6 px
4 pixels	2.33 %	5.16 %	0.9 px	1.6 px
5 pixels	1.86 %	4.25 %	0.9 px	1.6 px

Table 1. IRR results on KITTI 2012

ScopeFlow on KITTI 2012:

Error	Out-Noc	Out-All	Avg-Noc	Avg-All
2 pixels	4.36 %	8.30 %	0.7 px	1.3 px
3 pixels	2.68 %	5.66 %	0.7 px	1.3 px
4 pixels	1.96 %	4.39 %	0.7 px	1.3 px
5 pixels	1.56 %	3.60 %	0.7 px	1.3 px

Table 2. ScopeFlow results on KITTI 2012

In addition, Fig. 2 provides a qualitative comparison to the baseline IRR model on the KITTI 2012 benchmark. As can be seen, most of the improvement provided by the ScopeFlow pipeline is in the challenging occluded and marginal pixels.

D.3. KITTI 2015

We uploaded our results to the KITTI 2015 [5] benchmark. As can be seen in Tab. 3 and Tab. 4, training the IRR model with ScopeFlow pipeline improves the mean EPE by more than 12%, in the default category of 3 pixels. Moreover, the improvement is achieved for all thresholds of outliers and for all metrics.

IRR on KITTI 2015:

Error	Fl-bg	Fl-fg	Fl-all
All / All	7.68 %	7.52 %	7.65 %
Noc / All	4.92 %	4.62 %	4.86 %

Table 3. IRR results on KITTI 2015

ScopeFlow on KITTI 2015:

Error	Fl-bg	Fl-fg	Fl-all
All / All	6.72 %	7.36 %	6.82 %
Noc / All	4.44 %	4.49 %	4.45 %

Table 4. ScopeFlow results on KITTI 2015

In addition, Fig. 3 provides a qualitative comparison to the leading VCN model on the KITTI 2015 benchmark, showing a clear improvement for handling non-background challenging objects. Our results are leading the category of non-background pixels, which belong to faster foreground objects.

E. Ablation visualization

Fig. 4 provides a demonstration of the contribution of different training changes, composing the ScopeFlow pipeline presented in our paper, to the improvement of the final flow. As expected, most of the improvements are in the marginal image area. Our method improves, in particular, the moving objects, which have many occluded and fast-moving pixels.

F. Occlusions comparison

In order to provide a qualitative demonstration of our improved occlusion estimation, we compared our results to the methods with the highest reported occlusion estimation. We provide a layered view of false positive, false negative and true positive predictions. All occlusion estimations created using the pre-trained models, published by the authors, and sampled from the Sintel final pass dataset. Fig. 5 shows that the model trained with the ScopeFlow pipeline improves occlusion estimation in the marginal image area and mainly for foreground objects. We used the F1 metric, with an average approach of 'micro' (the same trend presented by all averaging approaches).

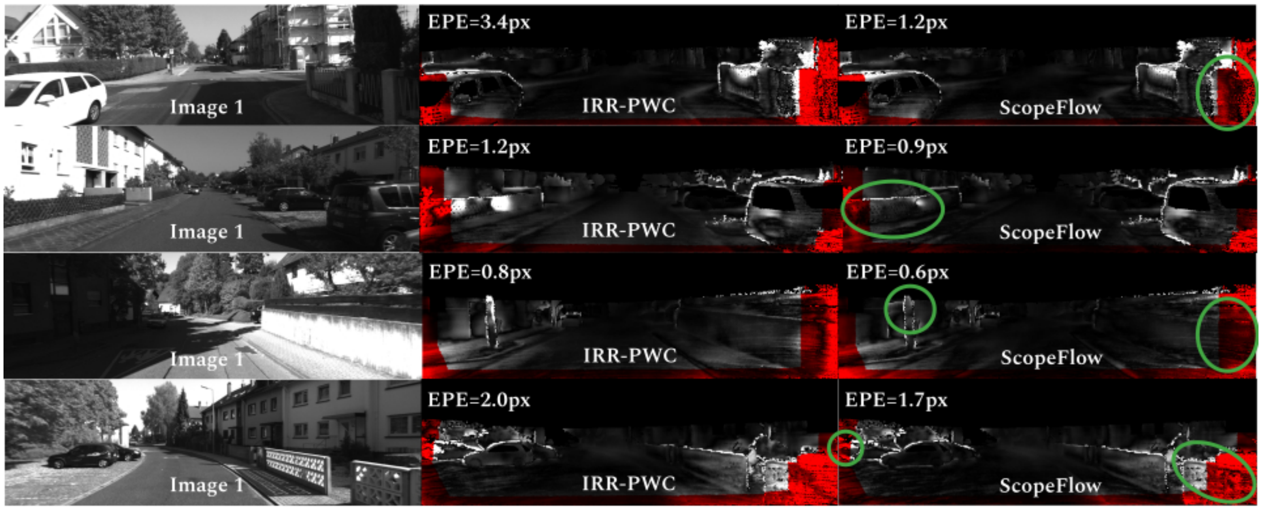

Figure 2. Qualitative comparison to the IRR baseline on KITTI 2012 benchmark. Our improved training pipeline got the lowest AEPE on KITTI 2012 among all other two-frame methods, using a low parameters off-the-shelf model architecture, which has an inferior performance on the KITTI benchmarks. Occluded regions are marked in red, erroneous regions with a higher intensity. Most of the improvement provided by ScopeFlow is in these challenging marginal pixels.

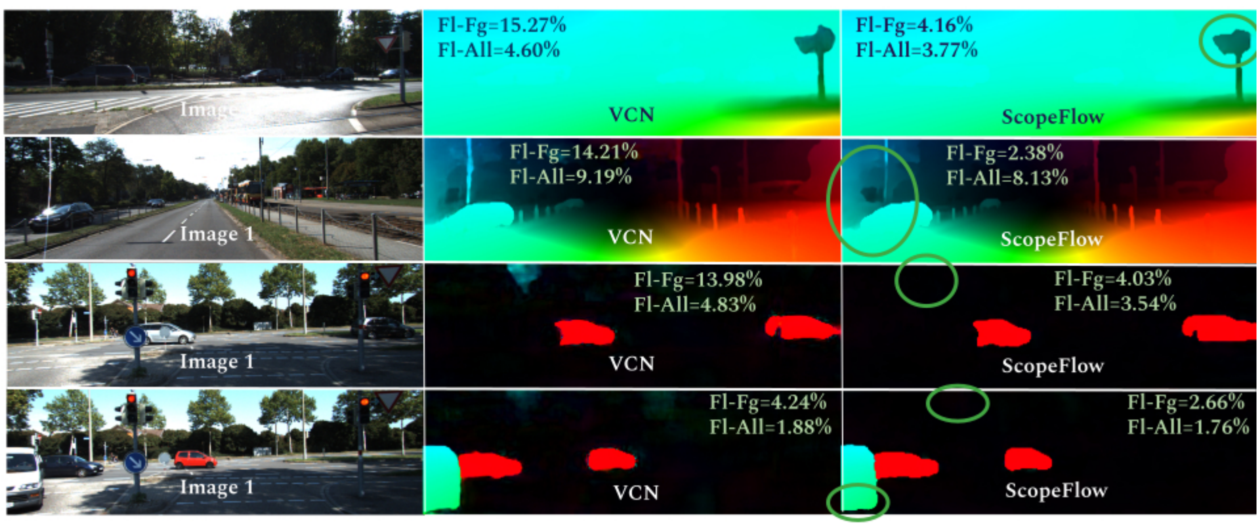

Figure 3. Qualitative comparison to the VCN [6] method KITTI 2015 benchmark. Although the VCN architecture gets the best outlier percentage among all pixels, we have a better handling for non-background objects among all other two-frame methods.

G. Public tables

We uploaded our results to the two main optical flow benchmarks: MPI Sintel and KITTI (2012 & 2015). In the subsections below, we provide the screenshots that capture the sizable improvements achieved by using our pipeline for training an optical flow model, with an off-the-shelf, low parameters model. Since our method can support almost any architecture, we plan, as future work, to apply it to other architectures as well.

G.1. MPI Sintel

We add here two screenshots of the public table: (i) the table on the day of upload (14.10.19), and (ii) the table after the official submission deadline for CVPR 2020. As shown in Fig. 6, our method ranks first on MPI Sintel since 14.10.19, surpassing all other methods, and leading the categories of: (i) matchable pixels, (ii) pixels far more than $10px$ from the nearest occlusion boundary, and (iii) fast-moving pixels ($>$ 40 pixels per frame). We also provide a screenshot taken after the official CVPR paper submission

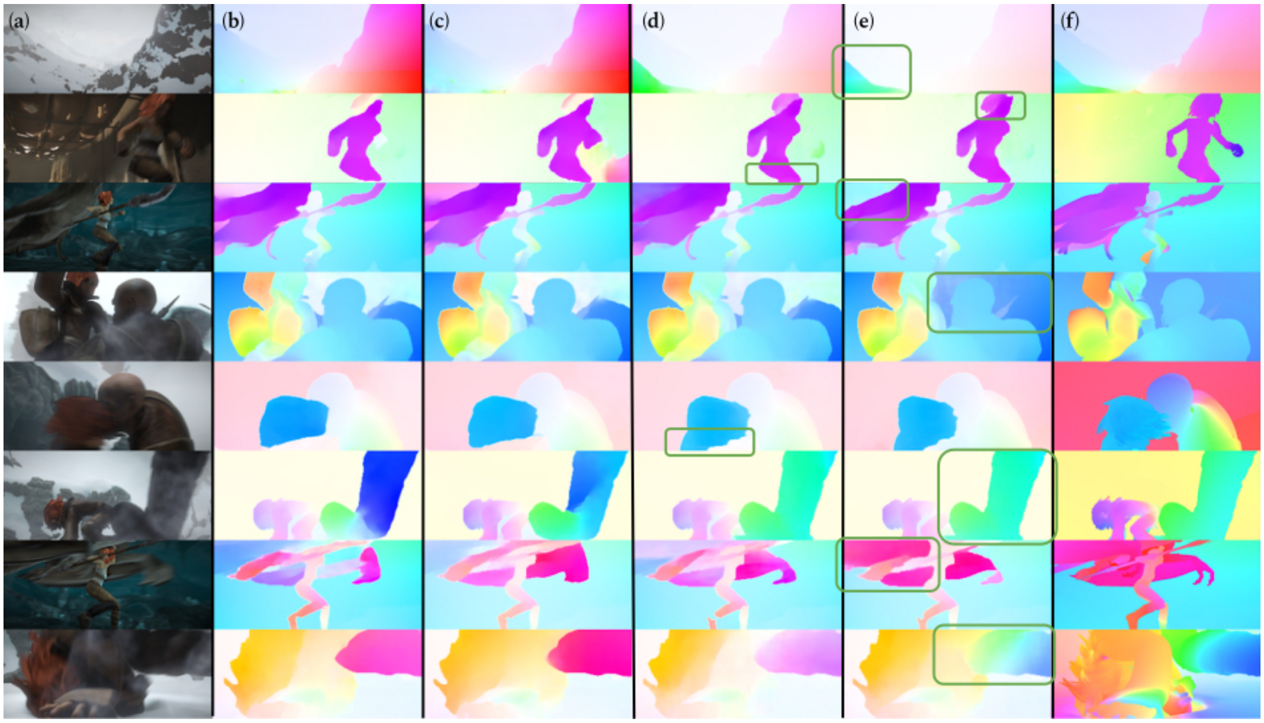

Figure 4. Ablation visualization on the MPI Sintel training set. (a) First image, (b) IRR-PWC baseline, (c) ScopeFlowR (reduced regularization), (d) ScopeFlowZ (zooming schedule), (e) ScopeFlow (final model), (f) Ground Truth flow.

deadline, in Fig. 7, showing our method still leading the Sintel benchmark. We changed our method's name after the initial upload (on 14.10.19) from OFBoost to ScopeFlow.

G.2. KITTI 2012

Fig. 8 shows a screenshot of the KITTI 2012 flow table, with the lowest outlier threshold (of 2%), taken on the CVPR paper submission deadline. Our method provides the lowest average EPE among all published two-frame methods, lower by 23% from the IRR-PWC baseline results.

G.3. KITTI 2015

Fig. 9 shows a screenshot of the KITTI 2015 flow table, taken on the CVPR paper submission deadline. Our method provides the lowest percentage of outliers, averaged over foreground regions, among all published two-frame methods. Moreover, the percentage of outliers, averaged over all ground truth pixels, is lower by more than 12% from the IRR-PWC baseline results.

References

[1] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012. 2

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

[3] Junhwa Hur and Stefan Roth. Iterative residual refinement for joint optical flow and occlusion estimation. In *CVPR*, 2019. 2, 5

[4] E. Ilg, T. Saikia, M. Keuper, and T. Brox. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *European Conference on Computer Vision (ECCV)*, 2018. 5

[5] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[6] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *NeurIPS*, 2019. 3

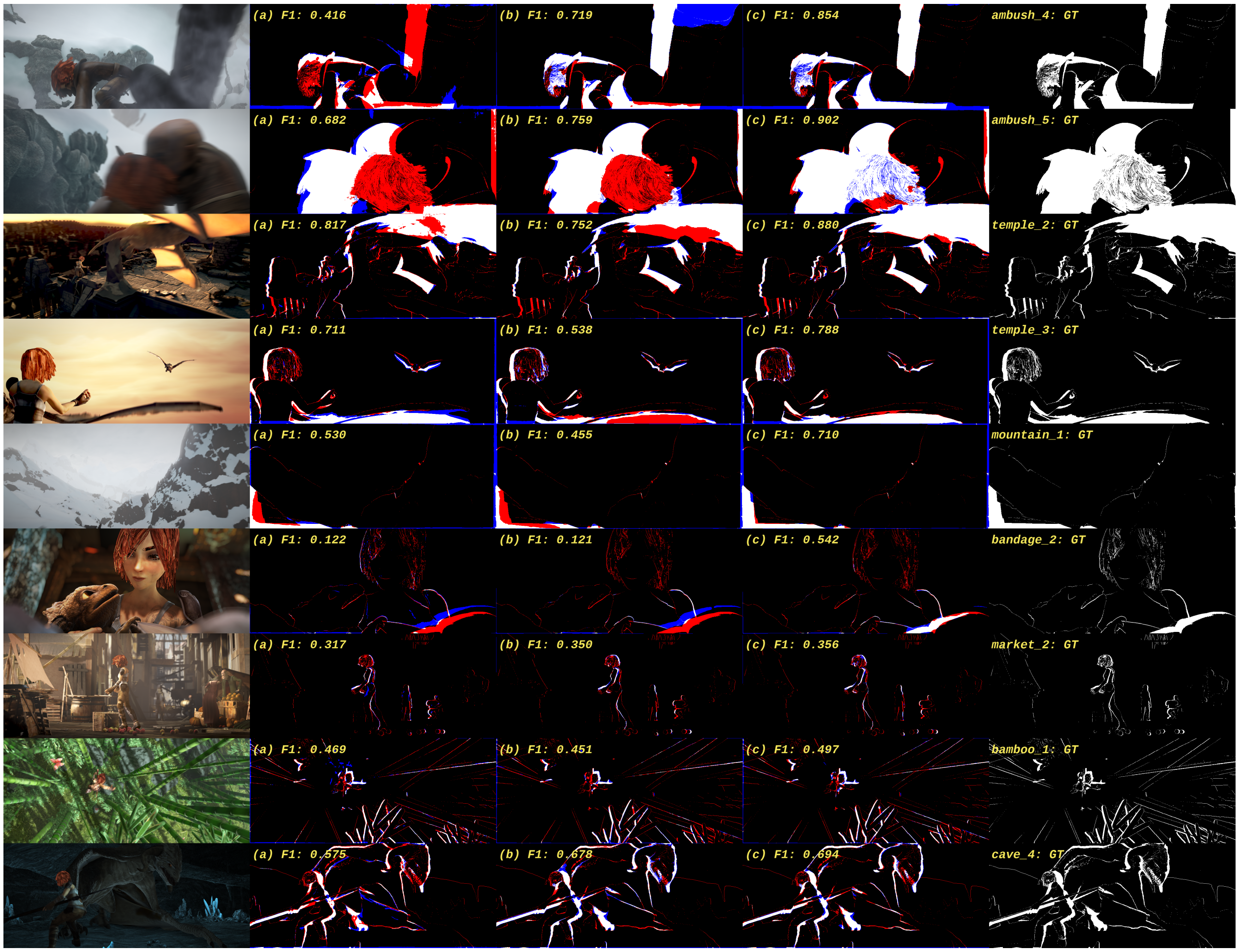

Figure 5. **Occlusion comparison over Sintel final pass**. Comparison of occlusion estimations created by: (a) FlowNet-CSSR-ft-sd [4], (b) IRR-PWC [3] baseline, and (c) ScopeFlow (ours). First frame on the left column and ground truth flow on the right column. For each occlusion map: false positive are in blue, false negative in red, and true positive in white. All occlusion maps estimated using Sintel Final samples and the original models published by the authors. Our improvements are mainly for foreground objects on the image margins.

Final | Clean

		EPE all	EPE matched	EPE unmatched	d0-10	d10-60	d60-140	s0-10	s10-40	s40+	
	GroundTruth [1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	Visualize Results
ScopeFlow →	OFBoost_v3 [2]	4.098	1.999	21.214	4.028	1.689	1.180	0.725	2.589	24.477	Visualize Results
	SelFlow [3]	4.262	2.040	22.369	4.083	1.715	1.287	0.582	2.343	27.154	Visualize Results
ScopeFlowZ →	OFBoost_v2 [4]	4.317	2.086	22.511	4.018	1.728	1.311	0.739	2.600	26.218	Visualize Results
ScopeFlowR →	OFBoost [5]	4.503	2.160	23.607	4.124	1.884	1.292	0.706	2.645	27.831	Visualize Results
	VCN [6]	4.520	2.195	23.478	4.423	1.802	1.357	0.934	2.816	26.434	Visualize Results
	ContinualFlow_ROB [7]	4.528	2.723	19.248	5.050	2.573	1.713	0.872	3.114	26.063	Visualize Results

Figure 6. Public Sintel table on the day of upload (taken on 14.10.19). Our method is leading the challenging final pass of the MPI Sintel benchmark. We renamed our method for clarity from OFBoost to ScopeFlow. ScopeFlowR is our method with regularization changes, ScopeFlowZ is our version with zooming changes. ScopeFlow is our final version with dynamic scoping.

	EPE all	EPE matched	EPE unmatched	d0-10	d10-60	d60-140	s0-10	s10-40	s40+	
GroundTruth [1]	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	Visualize Results
ScopeFlow [2]	4.098	1.999	21.214	4.028	1.689	1.180	0.725	2.589	24.477	Visualize Results
MaskFlownet [3]	4.172	2.048	21.494	3.783	1.745	1.310	0.592	2.389	26.253	Visualize Results
SelFlow [4]	4.262	2.040	22.369	4.083	1.715	1.287	0.582	2.343	27.154	Visualize Results
ScopeFlowZ [5]	4.317	2.086	22.511	4.018	1.728	1.311	0.739	2.600	26.218	Visualize Results
MaskFlownet-S [6]	4.384	2.120	22.840	3.905	1.821	1.359	0.645	2.526	27.429	Visualize Results
VCN [7]	4.404	2.216	22.238	4.381	1.782	1.423	0.955	2.725	25.570	Visualize Results
LiteFlowNet3 [8]	4.448	2.089	23.681	3.873	1.755	1.344	0.754	2.503	27.471	Visualize Results
ScopeFlowR [9]	4.503	2.160	23.607	4.124	1.884	1.292	0.706	2.645	27.831	Visualize Results
ContinualFlow_ROB [10]	4.528	2.723	19.248	5.050	2.573	1.713	0.872	3.114	26.063	Visualize Results
MFF [11]	4.566	2.216	23.732	4.664	2.017	1.222	0.893	2.902	26.810	Visualize Results
IRR-PWC [12]	4.579	2.154	24.355	4.165	1.843	1.292	0.709	2.423	28.998	Visualize Results
PWC-Net+ [13]	4.596	2.254	23.696	4.781	2.045	1.234	0.945	2.978	26.620	Visualize Results
PPAC-HD3 [14]	4.599	2.116	24.852	3.521	1.702	1.637	0.617	2.083	30.457	Visualize Results
CompactFlow [15]	4.626	2.099	25.253	4.192	1.825	1.233	0.845	2.677	28.120	Visualize Results
PCF-F [16]	4.630	2.197	24.465	3.410	1.737	1.744	0.603	2.131	30.652	Visualize Results
HD3-Flow [17]	4.666	2.174	24.994	3.786	1.719	1.647	0.657	2.182	30.579	Visualize Results
CompactFlow-woscv [18]	4.858	2.213	26.439	4.220	1.867	1.453	0.906	2.701	29.709	Visualize Results
SENSE [19]	4.860	2.301	25.732	4.121	1.991	1.493	0.812	2.606	30.402	Visualize Results
STC-Flow [20]	4.868	2.439	24.669	4.711	2.188	1.459	0.882	3.116	28.866	Visualize Results

Figure 7. Public Sintel table after CVPR papers submission deadline (taken on 18.11.19). Our method is still leading the main Sintel table after the addition of many new methods.

Error threshold 2 pixels **Evaluation area** All pixels

	Method	Setting	Code	Out-Noc	Out-All	Avg-Noc	Avg-All	Density	Runtime	Environment	Compare
1	DM-Net-i2		code	**0.00 %**	**0.00 %**	**0.0 px**	0.0 px	0.00 %	0.90 s	1 core @ 2.5 Ghz (C/C++)	
2	Anonym			0.00 %	0.00 %	0.0 px	0.0 px	0.00 %	TBD s	1 core @ 2.5 Ghz (Python)	
3	PCF-F			3.55 %	7.92 %	0.6 px	1.2 px	100.00 %	0.08 s	GPU @ 2.5 Ghz (Python)	
4	PPAC-HD3			3.57 %	7.59 %	0.6 px	1.2 px	100.00 %	0.14 s	GPU @ 2.5 Ghz (Python)	
5	HD^3-Flow		code	3.79 %	7.89 %	0.7 px	1.4 px	100.00 %	0.10 s	NVIDIA Pascal Titan XP	
Z. Yin, T. Darrell and F. Yu: Hierarchical Discrete Distribution Decomposition for Match Density Estimation. CVPR 2019.											
6	PRSM		code	4.10 %	6.92 %	0.7 px	1.0 px	100.00 %	300 s	1 core @ 2.5 Ghz (Matlab + C/C++)	
C. Vogel, K. Schindler and S. Roth: 3D Scene Flow Estimation with a Piecewise Rigid Scene Model. ijcv 2015.											
7	FF			4.28 %	8.82 %	0.7 px	1.4 px	100.00 %	0.058 s	NVIDIA GTX 1080	
8	cvpr-304			4.29 %	8.29 %	0.7 px	1.3 px	100.00 %	-1 s	1 core @ 2.5 Ghz (C/C++)	
9	ScopeFlow			4.36 %	8.30 %	0.7 px	1.3 px	100.00 %	-1 s	1 core @ 2.5 Ghz (Python)	
10	LiteFlowNet2		code	4.38 %	8.92 %	0.7 px	1.4 px	100.00 %	0.0486 s	GTX 1080 (slower than Pascal Titan X)	
T. Hui, X. Tang and C. Loy: A Lightweight Optical Flow CNN - Revisiting Data Fidelity and Regularization. arXiv preprint arXiv:1903.07414 2019.											
11	VC-SF			4.47 %	7.65 %	0.8 px	1.3 px	100.00 %	300 s	1 core @ 2.5 Ghz (Matlab + C/C++)	
C. Vogel, S. Roth and K. Schindler: View-Consistent 3D Scene Flow Estimation over Multiple Frames. Proceedings of European Conference on Computer Vision. Lecture Notes in, Computer Science 2014.											
12	PMC-PWC			4.47 %	8.41 %	0.7 px	1.4 px	100.00 %	-1 s	Nvidia 1080Ti (Python)	
13	HTC			4.52 %	11.24 %	0.8 px	1.6 px	100.00 %	0.03 s	1 core @ 2.5 Ghz (C/C++)	

Figure 8. Public KITTI 2012 flow table (with the lowest outlier threshold of 2%) on the CVPR paper submission deadline (taken on 15.11.19). Our method is with the lowest AEPE among all published two-frame methods, lower by 23% from the IRR-PWC baseline.

Evaluation ground truth All pixels **Evaluation area** All pixels

	Method	Setting	Code	Fl-bg	Fl-fg	Fl-all	Density	Runtime	Environment	Compare
1	UberATG-DRISF			**3.59 %**	10.40 %	**4.73 %**	100.00 %	0.75 s	CPU+GPU @ 2.5 Ghz (Python)	
W. Ma, S. Wang, R. Hu, Y. Xiong and R. Urtasun: Deep Rigid Instance Scene Flow. CVPR 2019.										
2	DH-SF			4.12 %	12.07 %	5.45 %	100.00 %	350 s	1 core @ 2.5 Ghz (Matlab + C/C++)	
3	PCF-F			6.05 %	**5.99 %**	6.04 %	100.00 %	0.08 s	GPU @ 2.5 Ghz (Python)	
4	ISF			5.40 %	10.29 %	6.22 %	100.00 %	10 min	1 core @ 3 Ghz (C/C++)	
A. Behl, O. Jafari, S. Mustikovela, H. Alhaija, C. Rother and A. Geiger: Bounding Boxes, Segmentations and Object Coordinates: How Important is Recognition for 3D Scene Flow Estimation in Autonomous Driving Scenarios?. International Conference on Computer Vision (ICCV) 2017.										
5	VCN			5.83 %	8.66 %	6.30 %	100.00 %	0.2 s	Titan X Pascal	
G. Yang and D. Ramanan: Volumetric Correspondence Networks for Optical Flow. NeurIPS 2019.										
6	HD^3-Flow		code	6.05 %	9.02 %	6.55 %	100.00 %	0.10 s	NVIDIA Pascal Titan XP	
Z. Yin, T. Darrell and F. Yu: Hierarchical Discrete Distribution Decomposition for Match Density Estimation. CVPR 2019.										
7	PRSM		code	5.33 %	13.40 %	6.68 %	100.00 %	300 s	1 core @ 2.5 Ghz (C/C++)	
C. Vogel, K. Schindler and S. Roth: 3D Scene Flow Estimation with a Piecewise Rigid Scene Model. ijcv 2015.										
8	cvpr-304			6.62 %	6.98 %	6.68 %	100.00 %	-1 s	1 core @ 2.5 Ghz (C/C++)	
ERROR: Wrong syntax in BIBTEX file.										
9	ScopeFlow			6.72 %	7.36 %	6.82 %	100.00 %	-1 s	GPU @ 2.5 Ghz (Python)	
10	OSF+TC			5.76 %	13.31 %	7.02 %	100.00 %	50 min	1 core @ 2.5 Ghz (C/C++)	
M. Neoral and J. Šochman: Object Scene Flow with Temporal Consistency. 22nd Computer Vision Winter Workshop (CVWW) 2017.										
11	SSF			5.63 %	14.71 %	7.14 %	100.00 %	5 min	1 core @ 2.5 Ghz (Matlab + C/C++)	
Z. Ren, D. Sun, J. Kautz and E. Sudderth: Cascaded Scene Flow Prediction using Semantic Segmentation. International Conference on 3D Vision (3DV) 2017.										
12	SPOSF			5.41 %	15.96 %	7.16 %	100.00 %	10 min	1 core @ 3.5 Ghz (Matlab + C/C++)	

Figure 9. Public KITTI 2015 flow table on the CVPR paper submission deadline (taken on 15.11.19). Our method is with the lowest percentage of foreground (objects) outliers among all published two-frame methods.