Probabilistic Pixel-Adaptive Refinement Networks
– Supplemental Material –

Anne S. Wannenwetsch[1,2*] Stefan Roth[2]
[1]Amazon, Germany [2] TU Darmstadt, Germany

In this supplemental material, we give further implementation details of the different types of refinement networks and provide results for a more comprehensive comparison on optical flow benchmarks. Moreover, we present an analysis considering the PPAC improvements on unreliable pixels as well as additional visualizations of PPAC-refined optical flow fields and segmentation maps.

A. Additional Implementation Details

A.1. Learning procedure

To train our networks, we use the Adam optimizer [36] with default parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and without weight decay. PPAC refinement networks are trained with a learning rate of 1×10^{-3} for networks on Sintel and a learning rate of 5×10^{-3} on KITTI. For semantic segmentation on Pascal VOC 2012, we use a learning rate of 1×10^{-4} for guidance and probability branches and 1×10^{-5} for the remaining PPAC parameters. The image inputs to all networks are normalized while estimates and log-probabilities remain unchanged. For faster training of all refinement networks, we save the outputs of the underlying backbone networks (*i.e*. HD3 or DeepLabv3+), and only propagate through the refinement step.

A.2. Network architectures

Tables 6 – 8 show the network structures used for PPAC, PAC, and our baseline simple refinement network, respectively. Here, 'C' represents standard convolution layers, 'P' layers with non-probabilistic PACs, and 'PP' layers with our PPACs. The networks for optical flow and semantic segmentation differ mainly by the number of input and output channels (2 or 21, respectively). For optical flow, the guidance branch uses only the first image as input since the flow fields should be aligned w.r.t. the objects in this image. All standard as well as PAC and PPAC-convolutions pad the inputs with zeros to preserve the feature size and use a stride of one. Moreover, a bias term is learned for all types of convolutions. The output of guidance and, if applicable, probability branches is split equally by the number of PAC or PPACs such that individual guidance is used for the components of the combination branch. For the simple setup, we equally experimented with networks with two convolutions and thus more channels but found the given one with three convolutions to perform better.

*This work was done at TU Darmstadt prior to Anne S. Wannenwetsch joining Amazon.

Table 6. Network structure of our PPAC networks for optical flow/semantic segmentation with ∼12.3k/14.3k parameters.

	layer type	kernel size	non-linearity	shared weights
guidance branch	C: $3 \to 15$	5×5	ReLU	✗
	C: $15 \to 15$	5×5	ReLU	✗
	C: $15 \to 10$	5×5	✗	✗
probability branch	C: $5/21 \to 5$	5×5	ReLU	✗
	C: $5 \to 5$	5×5	ReLU	✗
	C: $5 \to 2$	5×5	Sigmoid	✗
combination branch	PP: $2/21 \to 2/21$	7×7	✗	✓
	PP: $2/21 \to 2/21$	7×7	✗	✓

Table 7. Network structure of our PAC baseline networks for optical flow/semantic segmentation with ∼12.6k/15.5k parameters.

	layer type	kernel size	non-linearity	shared weights
guidance branch	C: $8/24 \to 15/13$	5×5	ReLU	✗
	C: $15/13 \to 15/13$	5×5	ReLU	✗
	C: $15/13 \to 10$	5×5	✗	✗
combination branch	P: $2/21 \to 2/21$	7×7	✗	✓
	P: $2/21 \to 2/21$	7×7	✗	✓

Table 8. Network structure of our simple baseline network for optical flow with a total of ∼12.4k parameters.

	layer type	kernel size	non-linearity	shared weights
simple branch	C: $10 \to 11$	7×7	ReLU	✗
	C: $11 \to 11$	7×7	ReLU	✗
	C: $11 \to 2$	7×7	✗	✗

Table 9. Average end-point error *(AEE)* of top-ranked two-frame optical flow methods on Sintel train and test. ⋆Re-evaluated for comparability.

	train		test	
	clean	final	clean	final
VCN [67]	(1.66)	(2.24)	**2.81**	**4.40**
IRR-PWC [26]	(1.92)	(2.51)	3.84	4.58
PWC-Net+ [64]	(1.71)	(2.34)	3.45	4.60
PPAC-HD3 *(ours)*	**(1.54)**	**(1.05)**	4.59	4.60
HD3 [60]	(1.68)⋆	(1.15)⋆	4.79	4.67

Table 10. Average end-point error *(AEE)* and 3-pixel outlier rate on non-occluded/all pixels *(Out-Noc/all)* of top-ranked optical flow methods on KITTI 2012 train and test. Results in parentheses indicate that data was used in training. †Methods use left and right stereo images. ⋆Re-evaluated for comparability.

	train	test		
	AEE	AEE	Out-Noc	Out-all
PPAC-HD3 *(ours)*	**(0.71)**	1.2	**2.01** %	5.09 %
HD3 [60]	(0.81)⋆	1.4	2.26 %	5.41 %
PRSM† [65]	–	**1.0**	2.46 %	**4.23** %
LiteFlowNet2 [62]	–	1.4	2.63 %	6.16 %
SPS-StFl† [66]	–	1.3	2.82 %	5.61 %

Table 11. Average end-point error *(AEE)*, 3-pixel outlier rate on all pixels *(Out-all)*, and runtimes *(time)* of top-ranked methods on KITTI 2015 train and test. Results in parentheses indicate that data was used in training. †Methods use left and right stereo images. ⋆Re-evaluated for comparability.

	train		test	
	AEE	Out-all	Out-all	time
UberATG-DRISF† [63]	–	–	**4.73** %	0.75 s
PPAC-HD3 *(ours)*	(1.20)	**(3.56** %)	6.06 %	0.19 s
ISF† [61]	–	–	6.22 %	600 s
VCN [67]	**(1.16)**	(4.1 %)	6.30 %	0.18 s
HD3 [60]	(1.40)⋆	(4.39 %)⋆	6.55 %	**0.11 s**⋆

B. Detailed Comparison on Optical Flow Benchmarks

For completeness, we give a more detailed comparison on benchmarks for optical flow, including the training results of PPAC-HD3 as well as the results of related work.

Table 9 shows results on Sintel clean and final. For comparability, we re-evaluated the flow fields of HD3 on the training splits, taking into account the available invalid masks. Our proposed PPAC-HD3 ranks 4th w.r.t. to the AEE on Sintel final.

Tables 10 and 11 summarize results for the best-ranked published methods on KITTI 2012 and 2015. For completeness, we also include scene flow methods. Note, however, that such approaches are not fully comparable as they leverage additional stereo images to compute flow. On both datasets, PPAC-HD3 ranks 1st among optical flow methods and 2nd over all published approaches on KITTI 2015. As we used the publically available checkpoint for HD3, which differs slightly from the one used in [60], we report re-evaluated results on the training sets. Moreover, we provide HD3 runtimes evaluated on the same GTX 1080 Ti GPU as PPAC-HD3 for fair comparison.

Table 12. Relative improvement of average end-point error *(AEE)*, evaluated on the 10% most unreliable and the remaining pixels of our Sintel and KITTI test splits.

	Sintel		KITTI
	clean	final	
Most unreliable pixels	9.86 %	8.93 %	4.28 %
Remaining pixels	4.11 %	3.01 %	9.42 %

C. Improvement of Unreliable Pixels

In the main paper, we argue that probabilistic pixel-adaptive refinement allows to propagate correct estimates into unreliable regions. Here, we examine the influence of PPACs on unreliable pixels in more detail. We evaluate the refinement of optical flow by computing the AEE on the 10% most unreliable pixels of each flow field and comparing it to the AEE of the remaining pixels. To assess the reliability of a pixel estimate, we upsample the probabilities of the last output scale and use nearest neighbor interpolation if the estimated residuals fall outside the probability grid. We found these reliabilities to correlate better with the optical flow errors than the ones obtained from the composed full matching probability distribution proposed in [60]. Moreover, we use the same PPAC refinement networks as trained for the experiments in Table 2.

Table 12 shows the relative improvement on unreliable and remaining pixels evaluated on our test splits of Sintel and KITTI. On Sintel clean and final, we clearly observe a more significant improvement on the unreliable pixels, justifying the conclusion that PPACs allow to replace pixels of low reliability. In contrast, our evaluation on KITTI shows a larger improvement for the remaining pixels. This correlates well with the fact that we found the output probabilities of [60] to be less well calibrated on KITTI, judging by the comparatively larger benefit of oracle confidences, *c.f.* Sec. 6.1. However, when comparing the relative improvements of PPACs to the ones obtained by PACs (8.87% for more reliable and 2.72% for uncertain pixels), we observe that PPACs nevertheless allow for better handling of unreliable regions even if the reliability estimates are not completely accurate themselves.

Figure 7. Examples of ground truth *(top)*, HD3 optical flow [60] *(middle)*, and our PPAC-refined optical flow *(bottom)* on Sintel final. *Best viewed on screen.*

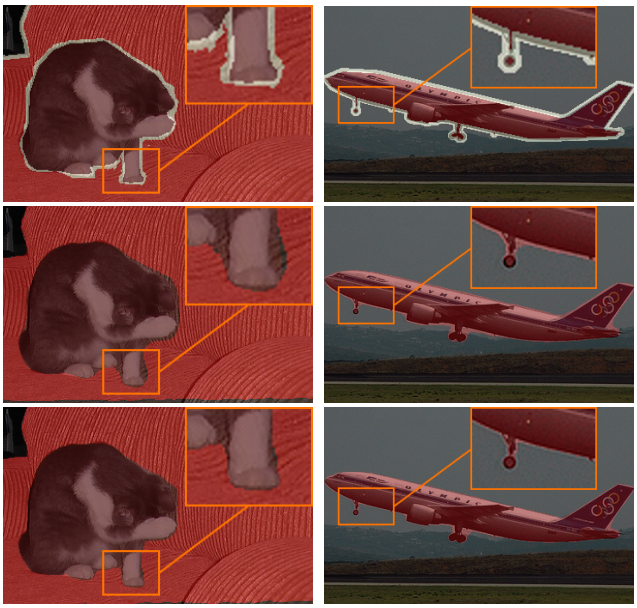

Figure 8. Additional examples of cropped ground truth *(top)*, DeepLabv3+ [7] *(middle)*, and PPAC-refined segmentation maps *(bottom)* on Pascal VOC 2012. *Best viewed on screen.*

D. Additional Visualizations

Fig. 7 shows additional visualizations of refined optical flow fields on our own validation and test splits of Sintel final. As such, none of these flow fields was presented to the PPAC refinement network during training. We clearly observe improved motion boundaries but also the ability of our approach to correctly propagate estimates into erroneous regions, *e.g.* the bird wings on the leftmost example.

In Fig. 8, we provide additional visualizations of refined segmentation maps on Pascal VOC 2012. PPAC refinement leads to a clear reduction of errors near object boundaries, *e.g.* by considerably minimizing the segmentation margin visible at the cat paw.

References

[61] Aseem Behl, Omid Hosseini Jafari, Siva Karthik Mustikovela, Hassan Abu Alhaija, Carsten Rother, and Andreas Geiger. Bounding boxes, segmentations and object coordinates: How important is recognition for 3D scene flow estimation in autonomous driving scenarios? In *ICCV*, pages 2574–2583, 2017.

[62] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. A lightweight optical flow CNN – Revisiting data fidelity and regularization. *arXiv:1903.07414 [cs.CV]*, 2019.

[63] Wei-Chiu Ma, Shenlong Wang, Rui Hu, Yuwen Xiong, and Raquel Urtasun. Deep rigid instance scene flow. In *CVPR*, pages 3614–3622, 2019.

[64] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Models matter, so does training: An empirical study of CNNs for optical flow estimation. *arXiv:1809.05571 [cs.CV]*, 2018.

[65] Christoph Vogel, Konrad Schindler, and Stefan Roth. 3D scene flow estimation with a piecewise rigid scene model. *Int. J. Comput. Vision*, 115(1):1–28, 2015.

[66] Koichiro Yamaguchi, David McAllester, and Raquel Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *ECCV*, volume 5, pages 756–771, 2014.

[67] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *NeurIPS*2019*, pages 784–805.