Supplementary Material for Disentangling Architecture and Training for Optical Flow

Deqing Sun*,† Charles Herrmann* Fitsum Reda
Michael Rubinstein David J. Fleet William T. Freeman

Google Research

The main paper includes only several examples on the Davis datasets due to space limits. Here we provide more visual examples to more comprehensively evaluate these models visually. We also include screenshots that indicate how our method does on public benchmarks and detailed results on these benchmarks. Throughout the document, we add "-it" to each method to denote our newly trained model, where "it" stands for improved training.

1 Screenshots of Public Benchmarks

Evaluation ground truth: All pixels   Evaluation area: All pixels

	Method	Setting	Code	Fl-bg	Fl-fg	Fl-all	Density	Runtime	Environment	Compare
1	CamLiFlow		code	**2.31 %**	7.04 %	**3.10 %**	100.00 %	1.2 s	GPU @ 2.5 Ghz (Python + C/C++)	
H. Liu, T. Lu, Y. Xu, J. Liu, W. Li and L. Chen: CamLiFlow: Bidirectional Camera-LiDAR Fusion for Joint Optical Flow and Scene Flow Estimation. CVPR 2022.										
2	RigidMask+ISF		code	2.63 %	7.85 %	3.50 %	100.00 %	3.3 s	GPU @ 2.5 Ghz (Python)	
G. Yang and D. Ramanan: Learning to Segment Rigid Motions from Two Frames. CVPR 2021.										
3	DRPC			3.17 %	8.79 %	4.11 %	100.00 %	2.7 s	GPU @ >3.5 Ghz (Python)	
4	DIP			3.86 %	5.96 %	4.21 %	100.00 %	0.15 s	1 core @ 2.5 Ghz (Python)	
5	RAFT-3D			3.39 %	8.79 %	4.29 %	100.00 %	2 s	GPU @ 2.5 Ghz (Python + C/C++)	
Z. Teed and J. Deng: RAFT-3D: Scene Flow using Rigid-Motion Embeddings. arXiv preprint arXiv:2012.00726 2020.										
6	LPSF			3.18 %	9.92 %	4.31 %	100.00 %	60 s	1 core @ 2.5 Ghz (C/C++)	
7	RAFT-it			4.11 %	**5.34 %**	4.31 %	100.00 %	0.1 s	GPU @ 2.5 Ghz (Python)	
8	SeparableFlow		code	4.25 %	5.92 %	4.53 %	100.00 %	0.5 s	GPU	
F. Zhang, O. Woodford, V. Prisacariu and P. Torr: Separable Flow: Learning Motion Cost Volumes for Optical Flow Estimation. Proceedings of the IEEE/CVF International Conference on Computer Vision 2021.										
9	MetaFlow			4.11 %	6.77 %	4.55 %	100.00 %	0.2 s	1 core @ 2.5 Ghz (Python)	
10	KPA-Flow			4.17 %	6.77 %	4.60 %	100.00 %	0.2 s	1 core @ 2.5 Ghz (Python)	
11	RealFlow			4.20 %	6.76 %	4.63 %	100.00 %	0.2 s	8 cores @ 2.5 Ghz (Python)	
12	FCTR			4.45 %	5.63 %	4.65 %	100.00 %	0.2 s	GPU @ 2.5 Ghz (Python)	
13	FlowNAS-RAFT-K			4.36 %	6.25 %	4.67 %	100.00 %	0.19 s	GPU @ 2.5 Ghz (Python)	
14	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.										
15	RAFT-A		code	4.54 %	5.99 %	4.78 %	100.00 %	0.7 s	GPU @ 2.5 Ghz (Python + C/C++)	
D. Sun, D. Vlasic, C. Herrmann, V. Jampani, M. Krainin, H. Chang, R. Zabih, W. Freeman and C. Liu: AutoFlow: Learning a Better Training Set for Optical Flow. CVPR 2021.										

Fig. 1. Screenshot of KITTI 2015 public benchmark. We name our newly trained RAFT as RAFT-it, "it" stands for improved training.

Table 1 summarizes the detailed results by previously published and our newly trained models. The newly trained models are more accurate than previ-

*Equal technical contribution, †project lead.

RAFT-it [17]	1.554	0.612	9.242	1.664	0.514	0.273	0.287	0.971	9.261	Visualize Results
RAFTwarm+OBS [18]	1.593	0.600	9.692	1.532	0.507	0.309	0.300	0.989	9.470	Visualize Results
RAFTv2-OER-warm-start [19]	1.594	0.625	9.487	1.567	0.512	0.339	0.328	1.014	9.271	Visualize Results
RAFT [20]	1.609	0.623	9.647	1.621	0.518	0.301	0.341	1.036	9.288	Visualize Results
NASFlow-RAFT [21]	1.613	0.503	10.664	1.339	0.405	0.238	0.298	0.892	9.883	Visualize Results
CSFlow-2-view [22]	1.626	0.584	10.123	1.527	0.492	0.254	0.330	1.015	9.539	Visualize Results
NASFlow [23]	1.629	0.639	9.708	1.616	0.540	0.334	0.306	1.001	9.718	Visualize Results
L2L-Flow-ext-warm [24]	1.648	0.622	10.017	1.641	0.516	0.282	0.342	1.018	9.657	Visualize Results
RAFT+NCUP [25]	1.661	0.678	9.666	1.872	0.541	0.302	0.371	1.102	9.402	Visualize Results

Fig. 2. Screenshot of Sintel clean public benchmark. We name our newly trained RAFT as RAFT-it, "it" stands for improved training.

SCAR [26]	2.882	1.391	15.038	3.101	1.145	0.773	0.651	1.759	16.665	Visualize Results
C1 [27]	2.884	1.436	14.696	3.050	1.199	0.821	0.608	1.786	16.833	Visualize Results
RAFT-it [28]	2.896	1.407	15.027	2.811	1.157	0.882	0.510	1.701	17.622	Visualize Results
RFPM [29]	2.901	1.331	15.698	2.732	1.063	0.811	0.535	1.602	17.779	Visualize Results
L2L-Flow-ext [30]	2.954	1.392	15.684	3.059	1.158	0.822	0.649	1.823	17.125	Visualize Results
FCTR [31]	2.979	1.323	16.489	2.963	1.103	0.760	0.664	1.815	17.290	Visualize Results
MF2C [32]	2.980	1.484	15.191	3.187	1.281	0.978	0.692	2.060	16.560	Visualize Results
CSFlow-2-view [33]	3.025	1.445	15.914	3.061	1.125	0.877	0.622	1.881	17.720	Visualize Results
MFFC [34]	3.029	1.517	15.363	3.135	1.189	0.916	0.621	1.812	17.929	Visualize Results
RAFT+OBS [35]	3.104	1.487	16.286	3.107	1.153	0.964	0.657	1.940	18.061	Visualize Results
RAFT-A [36]	3.137	1.590	15.762	3.153	1.270	1.032	0.534	1.956	18.912	Visualize Results

Fig. 3. Screenshot of Sintel final public benchmark. We name our newly trained RAFT as RAFT-it, "it" stands for improved training. RAFT-it is only slightly worse than SeparableFlow on Sintel.final among all published two-frame methods.

Optical flow evaluation results
Show images: below table above table in window

Statistics: Average SD R0.5 R1.0 R2.0 A50 A75 A95
Error type: endpoint angle interpolation normalized interpolation

Average endpoint error	avg. rank	Army (Hidden texture) GT im0 im1			Mequon (Hidden texture) GT im0 im1			Schefflera (Hidden texture) GT im0 im1			Wooden (Hidden texture) GT im0 im1			Grove (Synthetic) GT im0 im1			Urban (Synthetic) GT im0 im1			Yosemite (Synthetic) GT im0 im1			Teddy (Stereo) GT im0 im1		
		all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext	all	disc	untext
RAFT-it [194]	1.7	**0.07** 1	0.21 5	**0.05** 1	0.15 2	0.49 3	**0.11** 1	**0.17** 1	**0.32** 1	0.14 2	**0.05** 1	0.24 2	**0.03** 1	0.46 3	0.68 3	0.23 4	**0.09** 1	**0.31** 1	**0.07** 1	**0.06** 1	0.11 2	**0.07** 1	**0.28** 1	**0.61** 1	**0.20** 1
NNF-Local [75]	5.8	**0.07** 1	0.20 2	**0.05** 1	0.15 2	0.51 6	0.12 7	0.18 3	0.37 3	0.14 2	0.10 4	0.49 10	0.06 4	**0.41** 1	**0.61** 1	0.21 2	0.23 5	0.66 6	0.19 5	0.10 16	0.12 15	0.17 28	0.34 3	0.80 8	0.23 3
PMMST [112]	13.3	0.09 36	0.21 5	0.07 22	0.18 15	0.51 6	0.16 37	0.21 10	0.42 9	0.17 23	0.10 4	0.33 4	0.08 16	0.51 6	0.74 5	0.28 9	0.24 6	0.65 5	0.20 8	0.11 33	0.12 15	0.17 28	0.37 6	0.74 4	0.35 7
RAFT-TF_RVC [179]	13.3	0.10 53	0.30 78	**0.05** 1	0.18 15	0.55 13	0.14 25	0.21 10	0.43 10	0.19 47	0.06 2	**0.23** 1	0.04 2	0.51 6	0.75 7	0.25 5	0.14 2	0.42 2	0.11 2	0.07 3	0.12 15	0.08 2	0.37 6	0.80 8	0.27 4
OFLAF [78]	14.5	0.08 10	0.21 5	0.06 8	0.16 8	0.53 8	0.12 7	0.19 4	0.37 3	0.14 2	0.14 15	0.77 42	0.07 9	0.51 6	0.78 9	0.25 5	0.31 17	0.76 7	0.25 23	0.11 33	0.12 15	0.21 63	0.42 14	0.78 6	0.63 30
MDP-Flow2 [68]	15.0	0.08 10	0.21 5	0.07 22	0.15 2	**0.48** 1	**0.11** 1	0.20 6	0.40 6	0.14 2	0.15 32	0.80 51	0.08 16	0.63 24	0.93 24	0.43 27	0.26 9	0.76 7	0.23 14	0.11 33	0.12 15	0.17 28	0.38 9	0.79 7	0.44 10
NN-field [71]	16.4	0.08 10	0.22 17	**0.05** 1	0.17 12	0.55 13	0.13 15	0.19 4	0.39 5	0.15 8	0.09 3	0.48 9	0.05 3	**0.41** 1	**0.61** 1	**0.20** 1	0.52 74	0.64 4	0.26 28	0.13 63	0.13 47	0.20 57	0.35 4	0.83 11	0.21 2
ComponentFusion [94]	19.6	**0.07** 1	0.21 5	**0.05** 1	0.16 8	0.55 13	0.12 7	0.20 6	0.44 11	0.15 8	0.11 8	0.65 15	0.06 4	0.71 46	1.07 51	0.53 50	0.32 21	1.06 37	0.28 31	0.11 33	0.13 47	0.15 20	0.41 12	0.88 17	0.54 18
CoT-AMFlow [174]	22.4	0.08 10	0.22 17	0.07 22	0.15 2	**0.48** 1	0.12 7	0.21 10	0.45 12	0.15 8	0.16 46	0.86 65	0.08 16	0.67 36	0.96 30	0.56 55	0.27 11	0.82 12	0.24 19	0.12 49	0.12 15	0.18 44	0.42 14	0.85 12	0.60 25
TC/T-Flow [77]	26.4	**0.07** 1	0.21 5	**0.05** 1	0.19 24	0.68 40	0.12 7	0.28 33	0.66 39	0.14 2	0.14 15	0.86 65	0.07 9	0.67 36	0.98 37	0.49 43	0.22 4	0.82 12	0.19 5	0.11 33	0.11 2	0.30 113	0.50 36	1.02 38	0.64 33
PRAFlow_RVC [177]	27.2	0.11 66	0.27 59	0.08 53	0.24 70	0.64 31	0.19 64	0.28 33	0.61 29	0.23 76	0.12 12	0.62 14	0.06 4	0.60 16	0.87 15	0.38 16	0.18 3	0.50 3	0.16 3	0.07 3	0.12 15	0.08 2	0.49 30	0.92 22	0.51 14

Fig. 4. Screenshot of Middlebury public benchmark (AEPE). We name our newly trained RAFT as RAFT-it, "it" stands for improved training. RAFT-it sets a new state of the art on Middlebury.

ously models regardless of occlusions (unmatched), distance to motion boundaries, and speed.

Table 2 summarizes the detailed results on KITTI for the previously best published and the newly trained models. The newly trained models are generally better than the previously trained models. The only exception is the foreground regions for IRR-PWC. Note that the original IRR-PWC implementation computes bidirectional flow, reasons about occlusions, and uses a bilateral refinement, which may help the foreground objects. Our newly trained IRR-PWC

Model	all	match	unmatch	d0-10	d10-60	d60-140	s0-10	s10-40	s40+
PWC-Net	4.60	2.25	23.70	4.78	2.05	1.23	0.95	2.98	26.62
PWC-Net-it (ours)	3.68	1.82	18.87	3.47	1.39	1.18	0.62	1.96	23.07
IRR-PWC	4.58	2.15	24.36	4.17	1.84	1.29	0.71	2.42	29.00
IRR-PWC-it (ours)	3.56	1.83	17.54	3.67	1.40	1.16	0.63	2.04	21.63
RAFT [1]	3.14	1.59	15.76	3.15	1.27	1.03	0.53	1.96	18.91
RAFT-it (ours)	2.90	1.41	15.03	2.81	1.16	0.88	0.51	1.70	17.62

Table 1. Detailed analysis of AEPE on Sintel test set. "it" stands for improved training.

is a straightforward modification of PWC-Net and is more lightweight without these sophisticated modules.

Model	All			Occ		
	Fl-bg	**Fl-fg**	**Fl-all**	**Fl-bg**	**Fl-fg**	**Fl-all**
PWC-Net [2]	9.66 %	9.31 %	9.60 %	6.14 %	5.98 %	6.12 %
PWC-Net [3]	7.69 %	7.88 %	7.72 %	4.91 %	4.88 %	4.91%
PWC-Net-it (ours)	5.18 %	7.36 %	5.54 %	3.41 %	4.90 %	3.68 %
IRR-PWC	7.68 %	7.52 %	7.65 %	4.92 %	4.62 %	4.86 %
IRR-PWC-it (ours)	5.12 %	8.82 %	5.73 %	3.47 %	5.95 %	3.92 %
RAFT [4]	4.74 %	6.87 %	5.10 %	2.87 %	3.98 %	3.07 %
RAFT [1]	4.54 %	5.99 %	4.78 %	3.01 %	3.17 %	3.04%
RAFT-it (ours)	4.11 %	5.34 %	4.31 %	2.68 %	2.77 %	2.70 %

Table 2. Detailed performance on KITTI 2015 test set. "it" stands for improved training.

2 More Visual Comparisons

2.1 PWC-it, IRR-it, and RAFT-it on Davis 2K

In this subsection, we include 4 examples of our improved training models on Davis 2K images: Figures 5, 6, 7, and 8.

2.2 PWC-it, IRR-it, and RAFT-it (down-up) on Davis 4K

In this subsection, we include 4 examples of our improved training models on Davis 4K images: Figures 9, 10, 11, and 12. Note that due to memory constraints, RAFT-it requires the input to be downsampled and then the output flow to be upsampled to 4K.

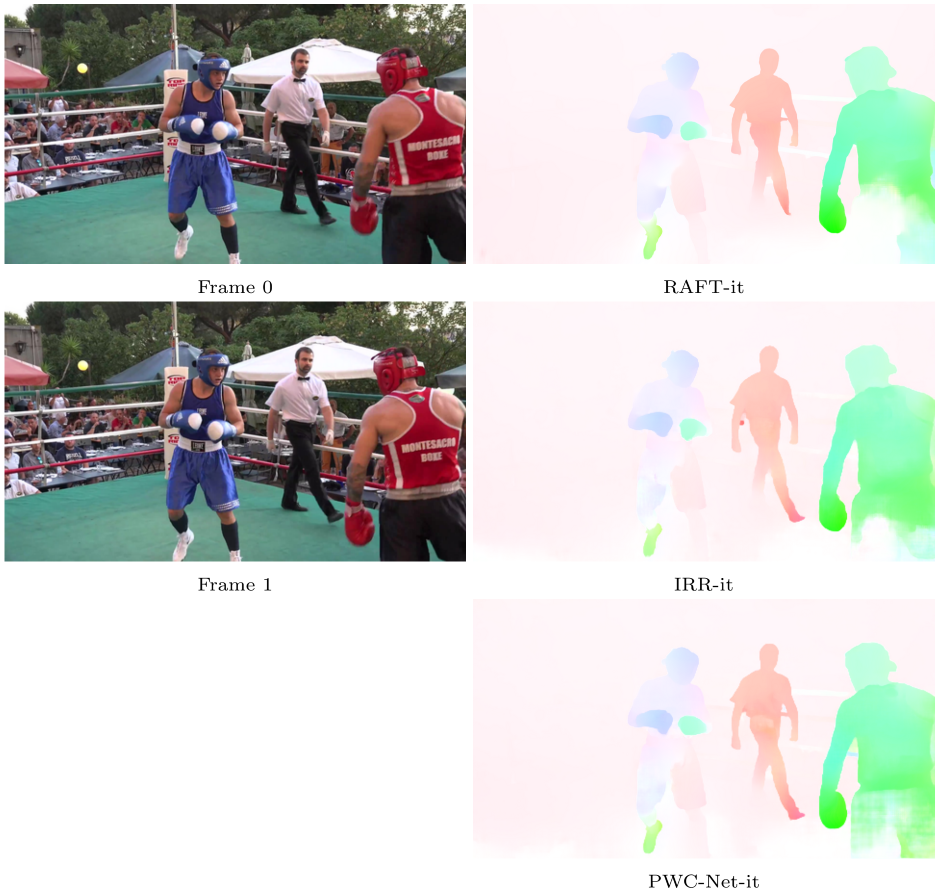

Fig. 5. PWC-it, IRR-it, and RAFT-it on Davis 2K images.

2.3 Old vs New for PWC and RAFT on Davis 448x864

In this subsection, we include 8 examples of two-frame optical flow from PWC-original, RAFT-original, PWC-it, and RAFT-it evaluated on Davis images with resolution 448 by 864: Figures 13.

2.4 Old vs New for PWC and RAFT on Viper 1080x1920

In this subsection, we include 3 examples of two-frame optical flow from PWC-original, RAFT-original, PWC-it, and RAFT-it evaluated on Viper validation images with resolution 1080 by 1920: Figures 14.

Fig. 6. PWC-it, IRR-it, and RAFT-it on Davis 2K images.

References

1. Sun, D., Vlasic, D., Herrmann, C., Jampani, V., Krainin, M., Chang, H., Zabih, R., Freeman, W.T., Liu, C.: Autoflow: Learning a better training set for optical flow. In: CVPR (2021)
2. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: CVPR (June 2018)
3. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Models matter, so does training: An empirical study of cnns for optical flow estimation. IEEE TPAMI (2019)
4. Teed, Z., Deng, J.: RAFT: Recurrent all-pairs field transforms for optical flow. In: Proc. ECCV (2020)

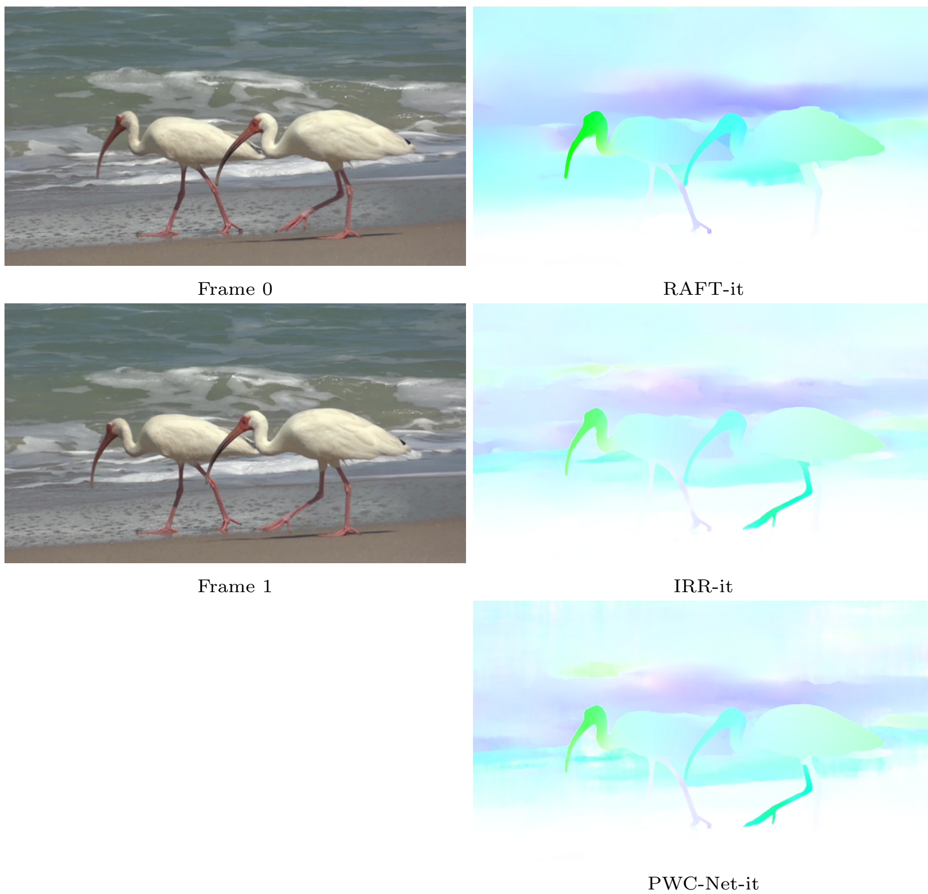

Fig. 7. PWC-it, IRR-it, and RAFT-it on Davis 2K images.

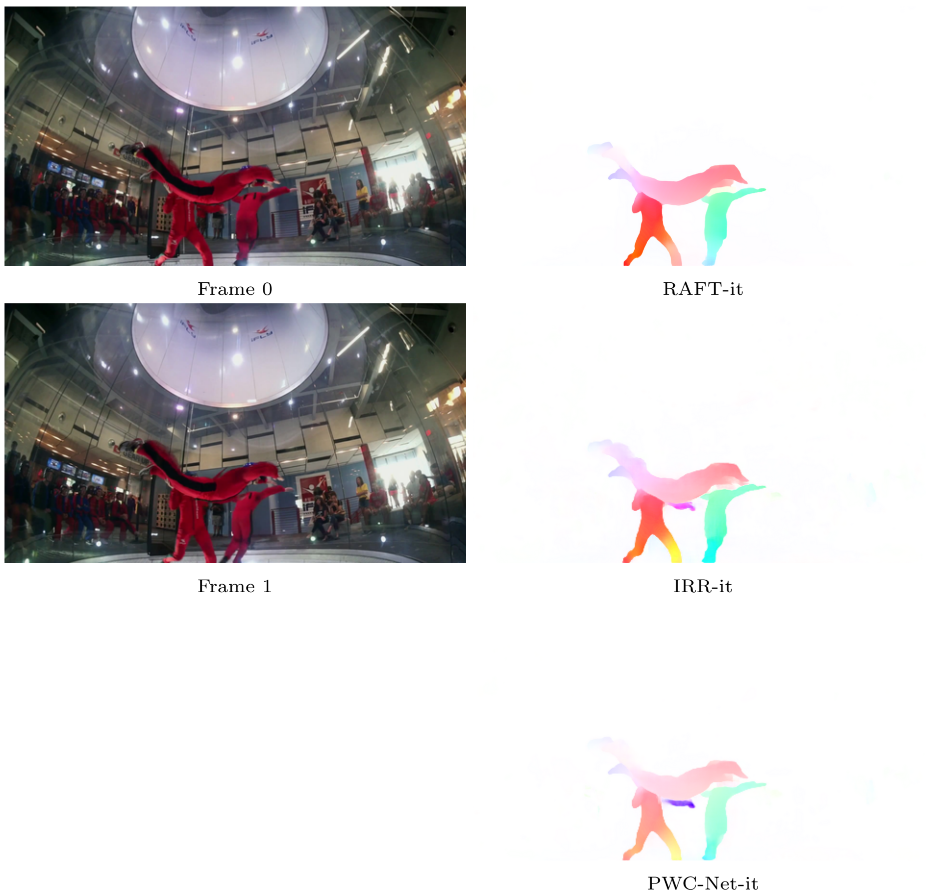

Fig. 8. PWC-it, IRR-it, and RAFT-it on Davis 2K images.

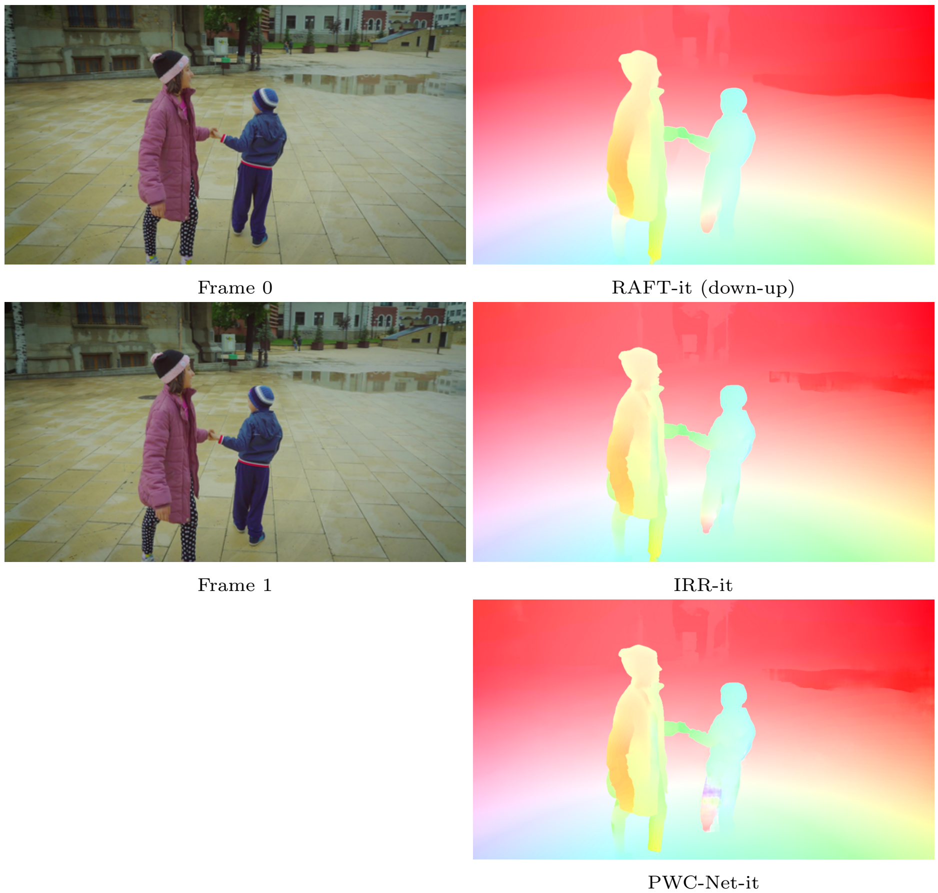

Fig. 9. PWC-it, IRR-it, and RAFT-it (down-up) on Davis 4K images. Note that RAFT-it requires the input images to be downsampled due to memory requirements and then the flow upsampled. Note the higher level of detail in IRR-it and PWC-Net-it: the clothes on the person on the left.

Fig. 10. PWC-it, IRR-it, and RAFT-it (down-up) on Davis 4K images. Note that RAFT-it requires the input images to be downsampled due to memory requirements and then the flow upsampled. Note the higher level of detail in IRR-it and PWC-Net-it: the biker's brim, the pant leg, etc.

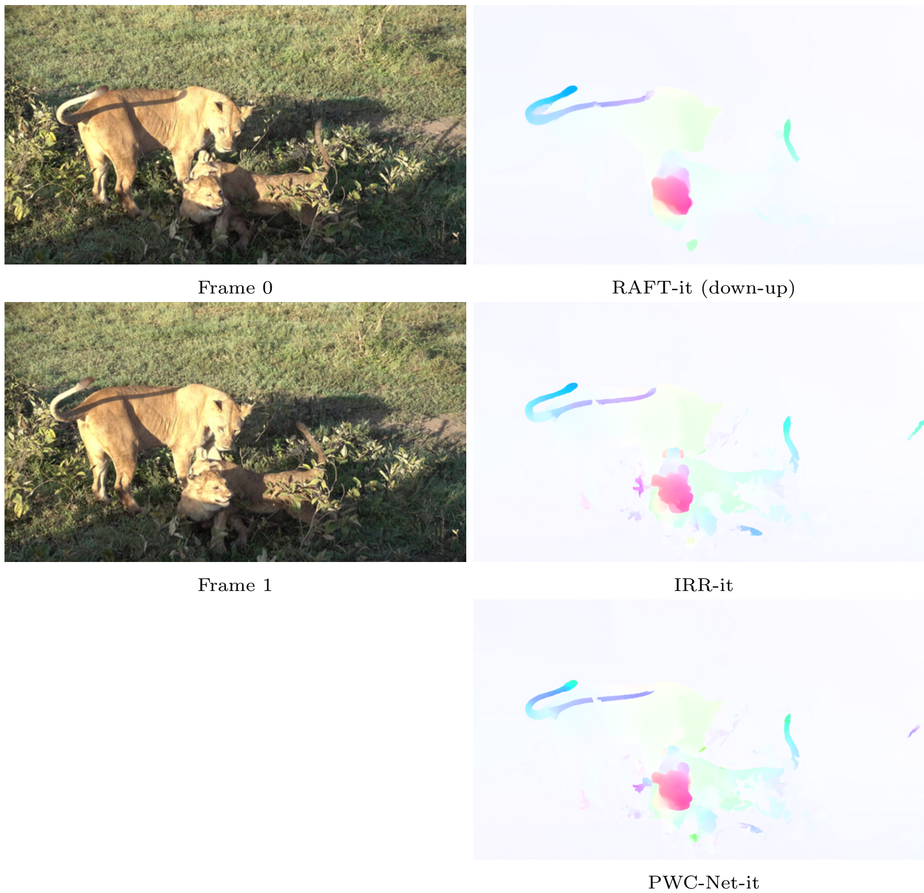

Fig. 11. PWC-it, IRR-it, and RAFT-it (down-up) on Davis 4K images. Note that RAFT-it requires the input images to be downsampled due to memory requirements and then the flow upsampled. Note the higher level of detail in IRR-it and PWC-Net-it: plant in front of the front lion.

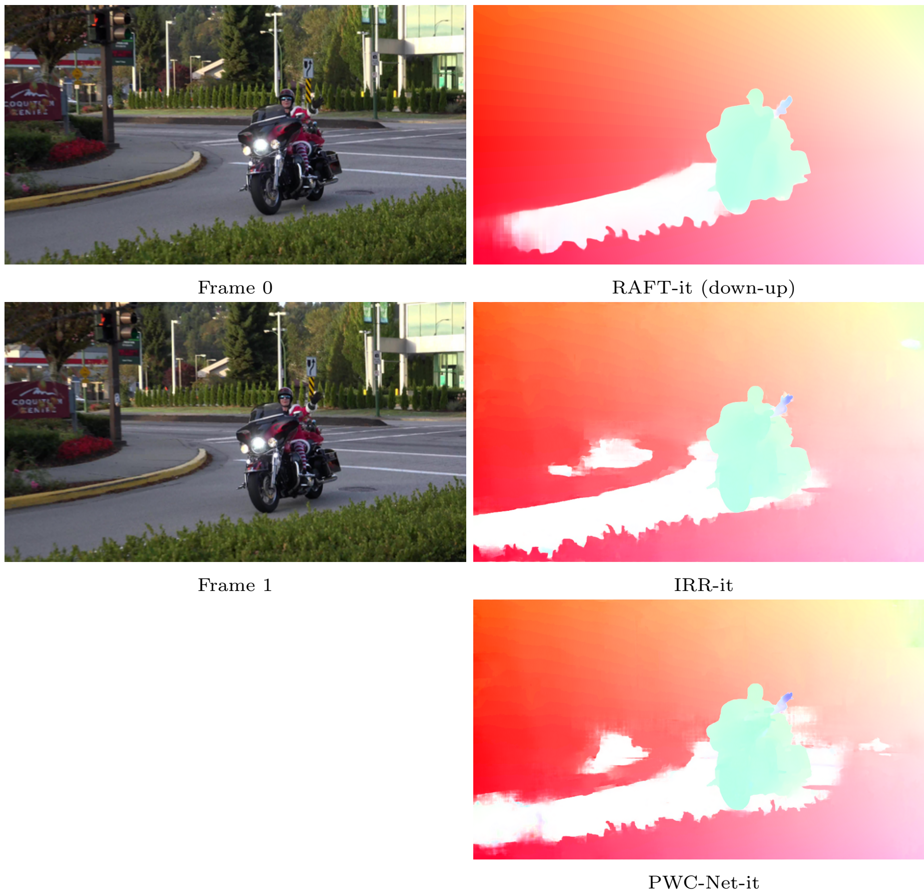

Fig. 12. PWC-it, IRR-it, and RAFT-it (down-up) on Davis 4K images. Note that RAFT-it requires the input images to be downsampled due to memory requirements and then the flow upsampled. Note the higher level of detail in IRR-it and PWC-Net-it: the plants in the foreground.

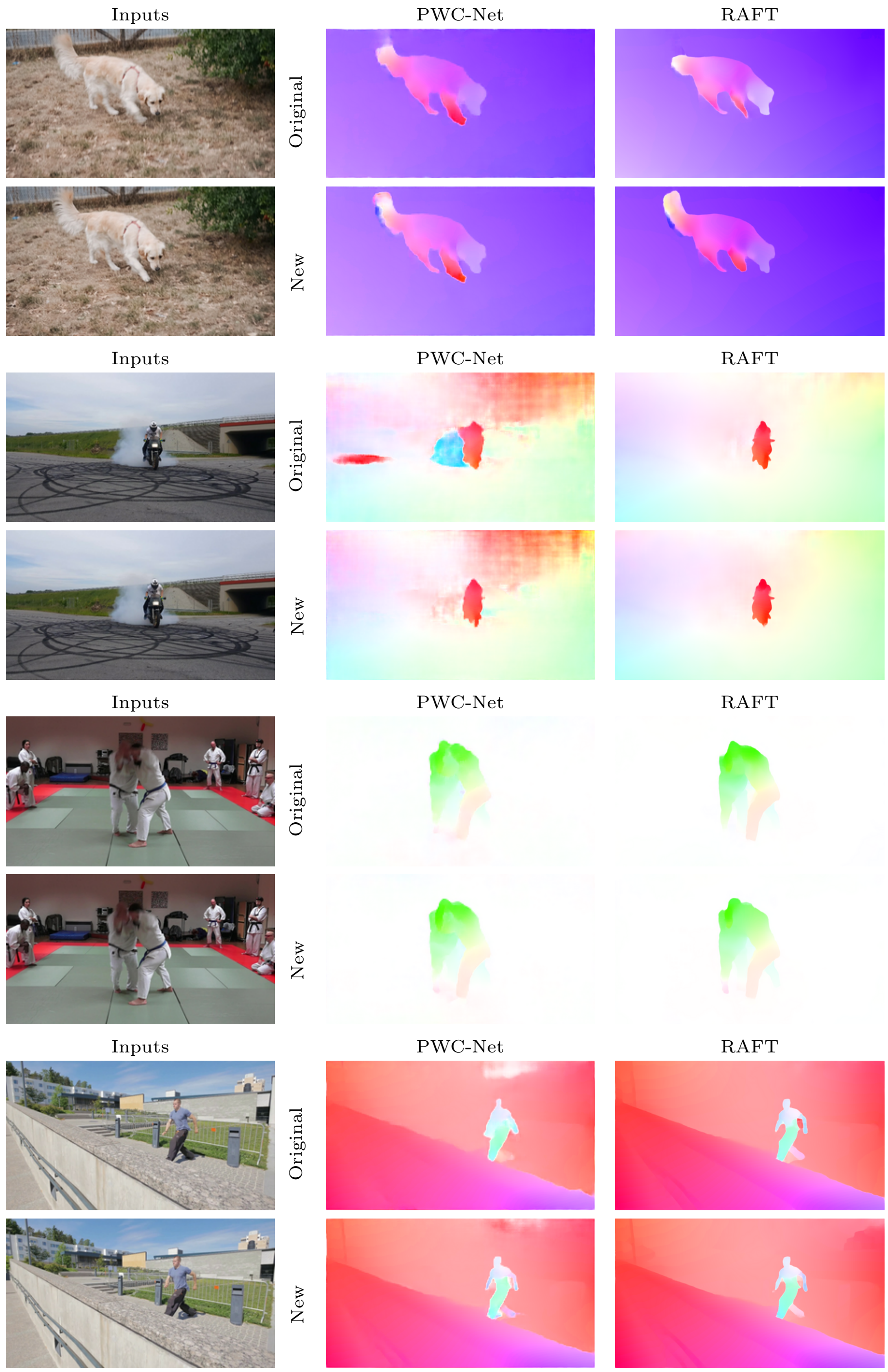

Fig. 13. PWC-orig, RAFT-orig vs PWC-it, RAFT-it on Davis 448x864

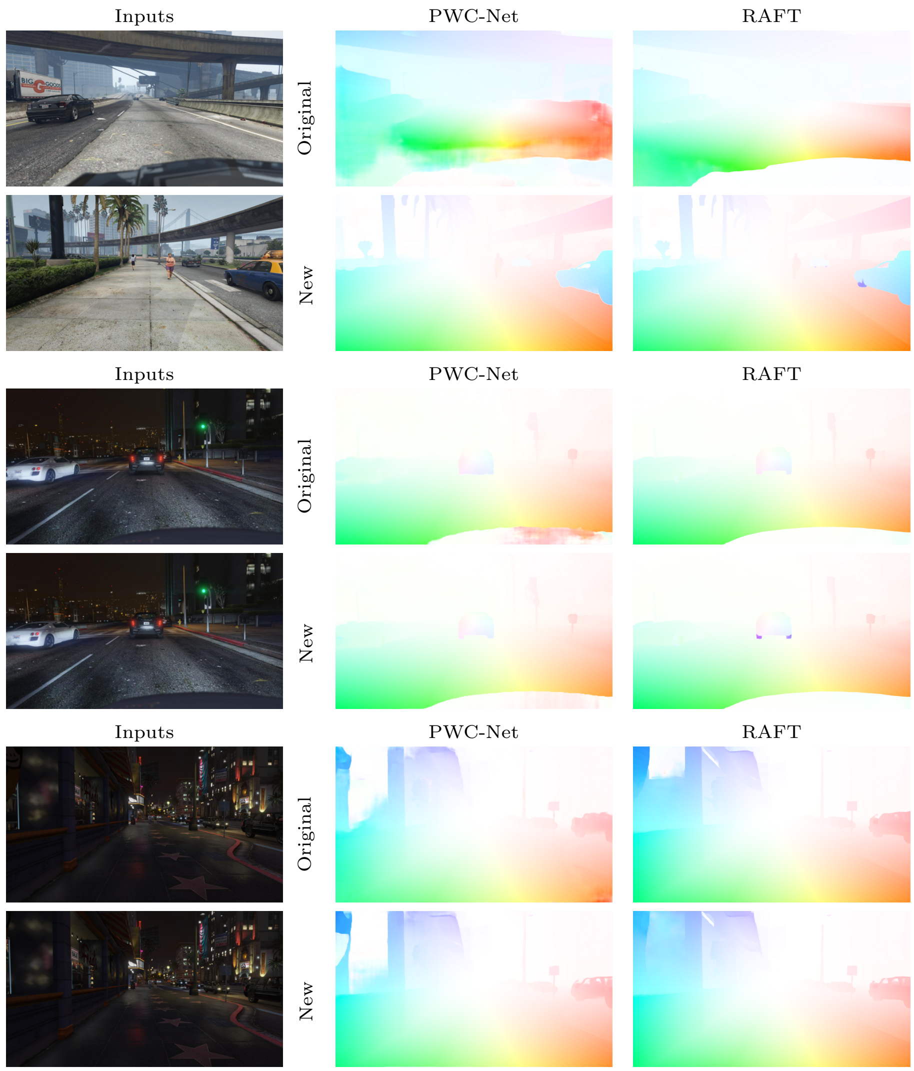

Fig. 14. PWC-orig, RAFT-orig vs PWC-it, RAFT-it on Viper 1080x1920