Removing Cost Volumes from Optical Flow Estimators

Supplementary Material

A.1. Complexity

In the main paper, we only refer to FLOPS as a measure of complexity. However, a reduction of FLOPS does not necessarily lead to a reduction in compute time on currently available accelerators, mostly due to memory alignment issues. Since we only remove entire parts of the networks and do not introduce sparsity or similar, we find that the reduction in FLOPS is proportional to the reduction in runtime. For completeness, we show the runtimes for different resolutions in Fig. A.1.

Another limiting factor is often the amount of memory required for a single prediction. We evaluate the memory footprint of our method in Fig. A.2, and due to the missing cost volumes, we find a significant reduction in memory footprint. However, technically RAFT-style architectures never require every value of the cost volume to be available at the same time since the refinement network can only sample a certain number of values from the cost volume per iteration. Therefore, the required values could be calculated only when they are requested from the refinement module. This was also noticed and implemented by Teed and Deng in the original implementation of RAFT, but the disadvantage of this approach is a significant slowdown in processing speed, and therefore, we do not consider this approach in our work.

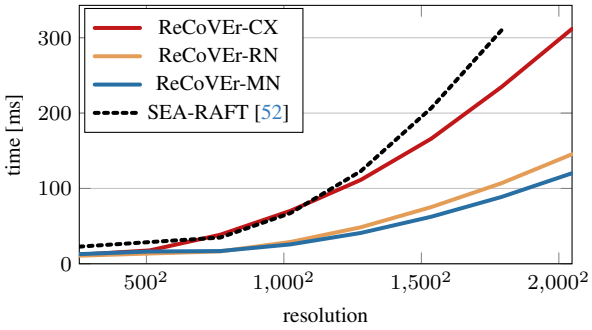

Figure A.1. **Runtime** of our methods and SEA-RAFT for different input resolutions.

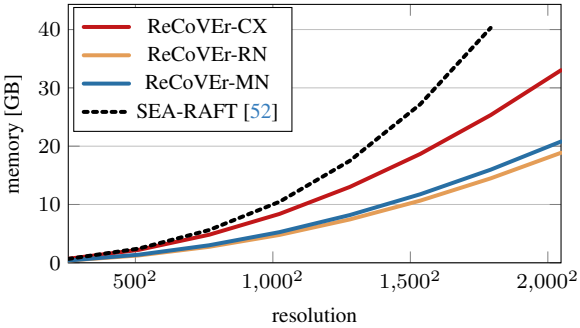

Figure A.2. **Memory** requirements of our methods and SEA-RAFT for different input resolutions.

A.2. Downsample-upsample strategies

Downsampling the inputs and bilinearly upsampling the resulting optical flow is another method to reduce the memory footprint and inference time, and is, *e.g.*, applied by SEA-RAFT on Full-HD frames to increase the computational efficiency for higher resolution inputs [52]. In our work, we did not apply this orthogonal strategy as it can be applied theoretically to all optical flow methods. Table A.1 shows that when evaluating on the Spring dataset, most methods achieve even higher accuracies using the downsample-upsample strategy, but the results on Sintel and Monkaa clearly show that the increase in accuracy is not persistent between datasets.

A.3. Additional technical details

Refinement network. The refinement network is implemented such that the sampling from the cost volume during the refinement can return an all-zero tensor instead of actually sampling from the cost volume. This simplifies the implementation of cutting away the feature network because by returning only zeros, the weights of the first layer of the refinement network that deal with this part of the input are not used since all of them are multiplied by zero. Theoretically, removing the affected weights from this layer completely would be possible. Still, this way of implementing the process is much easier, and the number of calculations needed to process the zero tensor is negligible compared to all other computations necessary for the optical flow prediction.

Training protocol. As described in Sec. 3.2, we mostly follow the training protocol proposed by SEA-RAFT [52]. The only difference is the composition of the TSKH dataset, where we do not include the validation split of Sintel. This allows us to fairly evaluate our methods on parts of the Sintel dataset for our analysis.

A.4. Qualitative examples

More qualitative examples, including samples from Sintel [5], Spring [33], and natural images taken from KITTI [35, 36] and DAVIS [40] can be found in Figs. A.3 and A.4.

Method	downsample	Sintel (val.) [5]		Monkaa [32]		Spring [33]	time[ms]	memory[GB]
		Clean	Final	Clean	Final			
PWC-Net [47, 48]	✗	3.22	3.66	4.19	4.56	2.31	43.72	1.25
	✓	5.27	5.95	5.16	5.34	3.83	12.30	0.36
RAFT [50]	✗	(0.74)	(1.19)	1.99	2.92	0.66	161.19	8.00
	✓	(1.38)	(2.07)	1.98	2.59	0.48	28.04	0.56
GMFlow [56]	✗	(0.76)	(1.11)	3.01	2.97	0.65	1004.23	8.28
	✓	(1.98)	(2.55)	2.08	3.59	0.95	99.44	1.47
GMA [25]	✗	(0.62)	(1.06)	1.75	2.63	0.55	381.21	13.29
	✓	(1.27)	(1.95)	1.90	2.57	0.46	46.88	0.92
SEA-RAFT [52]	✗	(0.43)	(0.58)	**1.69**	2.41	0.54	169.72	8.22
	✓	(1.22)	(2.06)	1.73	**2.09**	**0.41**	28.66	0.66
ReCoVEr-MN	✗	0.81	0.90	2.88	3.03	0.99	50.16	0.49
	✓	2.64	2.82	3.61	3.93	0.79	**12.12**	**0.28**
ReCoVEr-RN	✗	0.72	0.84	2.16	2.77	0.62	65.03	0.93
	✓	1.85	2.33	2.52	3.05	0.57	13.46	0.30
ReCoVEr-CX	✗	**0.36**	**0.42**	1.71	2.46	0.51	144.41	1.24
	✓	1.34	1.71	2.15	2.70	0.50	29.89	0.43

Table A.1. **Effect of downsampling** by a factor of 2× and bilinearly upsampling the resulting optical flow on different datasets. The time and memory refer to input frames of size 1920 × 1080. For completeness, we also show the accuracies achievable by methods that were also trained on the Sintel validation set, and we put these numbers in parentheses.

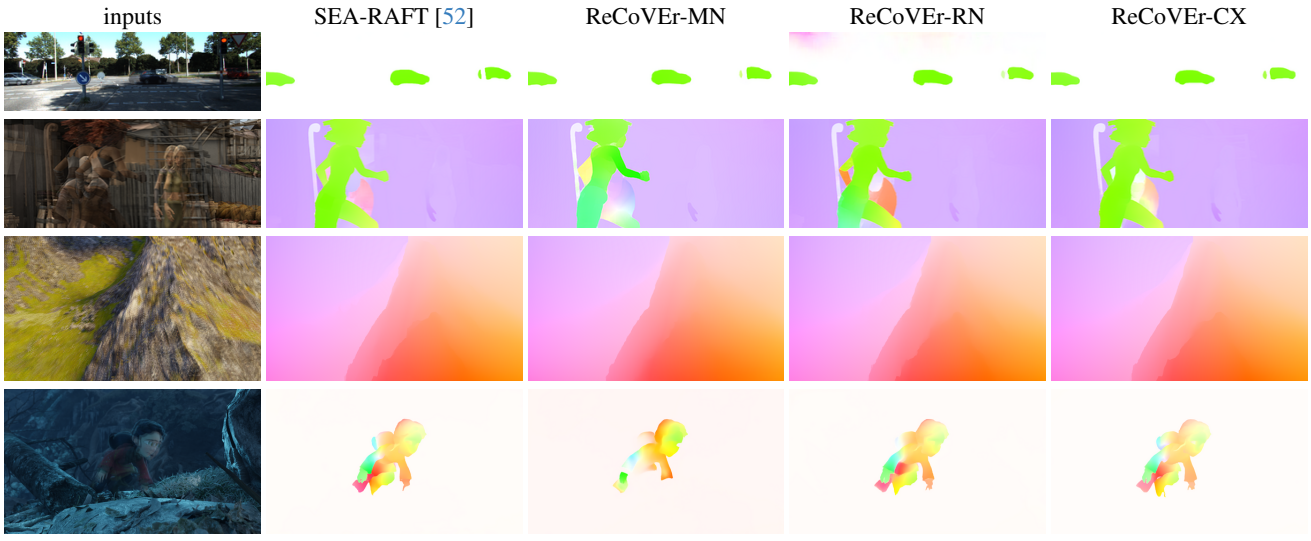

Figure A.3. More qualitative examples on various frames taken from DAVIS [40], KITTI [35, 36], Sintel [5], and Spring [33].

.

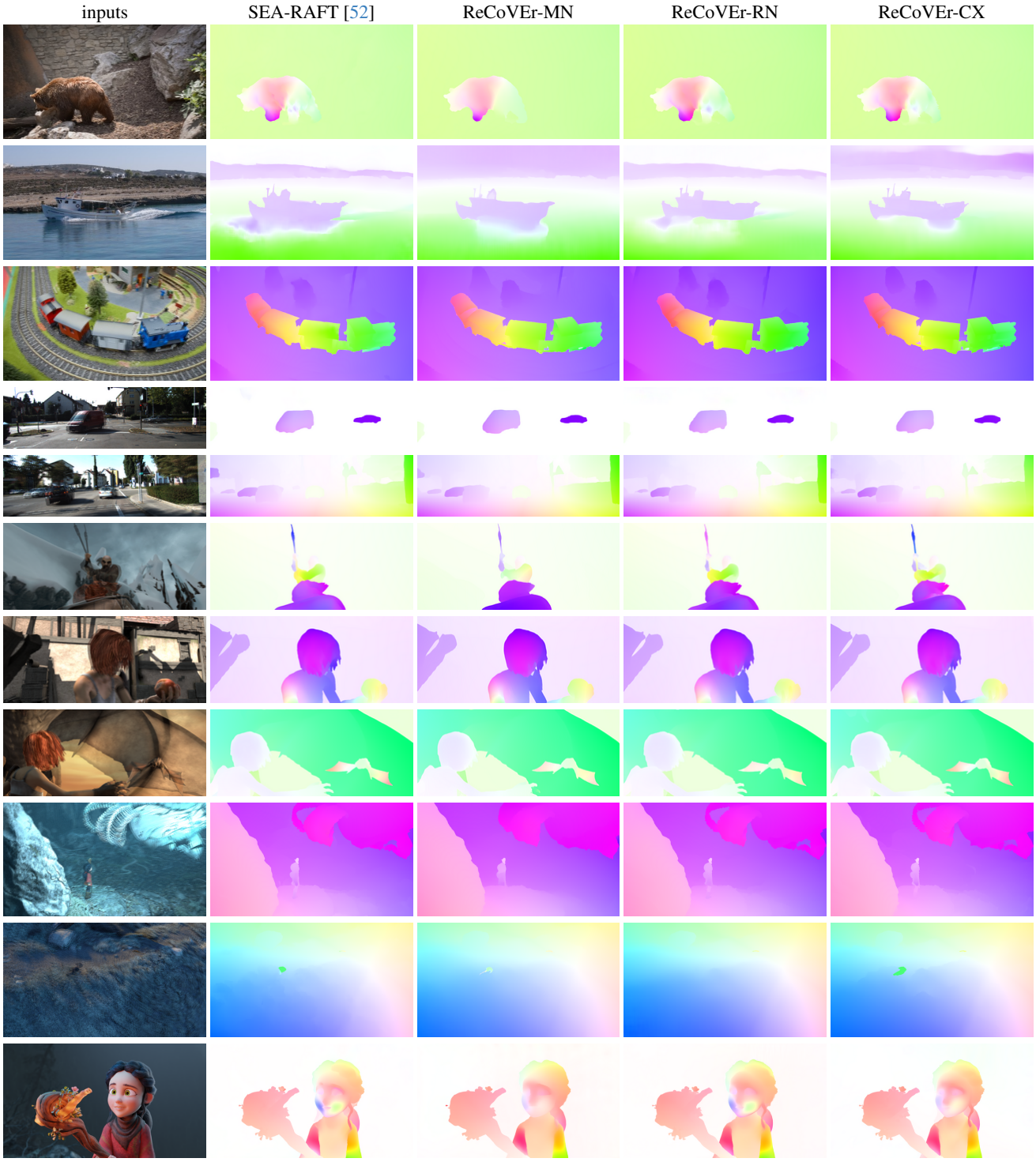

Figure A.4. More qualitative examples on various frames taken from DAVIS [40], KITTI [35, 36], Sintel [5], and Spring [33].