Reviving Unsupervised Optical Flow:
Concept Reevaluation, Multi-Scale Advances and Full Open-Source Release
Supplementary Material

Azin Jahedi[1] Marc Rivinius[2] Noah Berenguel Senn[1] Andrés Bruhn[1]
[1] University of Stuttgart, VIS, Computer Vision Group [2] University of Stuttgart, SEC
{Azin.Jahedi, Noah.Berenguel-Senn, Andres.Bruhn}@vis.uni-stuttgart.de
Marc.Rivinius@sec.uni-stuttgart.de

1. Notes on Reproducing SMURF

Although SMURF emerged a few years ago and achieved impressive results, so far it has not been used as a basis for further advancements in the field. Subsequent works have either employed SMURF's final results for post-processing [7] or, despite utilizing RAFT as a backbone and adopting SMURF's training schedule [4], have struggled to replicate its remarkable outcome, even without multi-frame self-supervision (see Table 5 in [5]). We also struggled to reproduce SMURF using its published code. In fact, in half of our trials, the network did not converge. The reproducibility issue has occurred for others as well (see `https://github.com/google-research/google-research/issues/1036`). Moreover, the current implementation loads one training sample per GPU, meaning that 8 GPUs for training are required, even without considering the more resource-intensive multi-frame training (see `https://github.com/google-research/google-research/issues/895`). Requiring this large amount of resources, further limits the accessibility of this SOTA work. Besides, there are differences between the code and the paper. For example, in the paper there are ablations where the negative impact of forward-backward masking during self-supervision is shown, hence removing such a masking is concluded, however the same masking is applied in the code (by default) for training, following the training instructions in the repository.

2. Model Details

As mentioned in the main paper, our Sun-RAFT model has the exact same architecture as RAFT and thereby has 5.3 M parameters. Our multi-scale Muun-RAFT model has 7.9 M parameters. Note that the number of parameters in this case is much smaller than [1, 2], where the main backbone and encoders are adapted from, respectively. Thereby, we save parameters by using a top-down feature encoder, instead of a Unet-based encoder and by sharing the residual units of the Unet-based context encoder across scales (see next section for ablations).

Method	Chairs	Sintel train		KITTI train	
	Val	Clean	Final	EPE	FL
Context-based Upsampling Module					
Shared Unit across Scales	1.71	1.50	2.74	2.71	8.85
Individual Units (Ours)	1.71	1.44	2.56	2.34	8.64
Feature Encoder					
Unet-based [1]	1.73	1.51	2.59	2.85	9.08
Top-down (Ours)	1.71	1.44	2.56	2.34	8.64

Table 1. Architectural ablations for Muun-RAFT.

Method	Chairs	Sintel train		KITTI train	
	Val	Clean	Final	EPE	FL
Upsampling Strategy					
MU ×4 - MU ×2 - Bilin.	1.72	1.54	2.66	2.70	9.43
All MU	1.68	1.60	2.78	2.83	9.37
MU ×2 - CU ×2 (Ours)	1.71	1.44	2.56	2.34	8.64
Number of Recurrent Iterations					
[4,6,8] (MS-RAFT [1])	1.72	1.46	2.58	2.64	8.94
[4,4,4] (Ours)	1.71	1.44	2.56	2.34	8.64

Table 2. Further ablations for Muun-RAFT: upsampling and number of recurrent iterations.

3. Further Ablations

Table 1 shows further experiments on the architecture of our Muun-RAFT model. In the first ablation, we investigated using separate or joint learned upsampling heads for computing the context-based convex upsampling masks. Using individual learned units for each scale yielded slightly better results in total.

For computing image-features, we employed a top-down feature encoder for Muun-RAFT, unlike the original architecture of MS-RAFT [1], where a Unet-based feature encoder was employed. The next experiment shows that the larger Unet-based feature encoder yields on-par results on Chairs and Sintel, but yields slightly worse results on KITTI.

Further, we investigate using different factors of MU-type upsampling masks instead of employing a context-based upsampling in Table 2. In the first experiment, we employ a MU $\times 4$ convex upsampling to upsample the native flow from the last scale to full-resolution, while keeping the MU $\times 2$ upsampling for inter-scale initializations. Finally, we also assessed an upsampling strategy very similar to RAFT [6], where we upsample all the native flow estimates to full-resolution via large factor MU masks ($\times 16$, $\times 8$ and $\times 4$). Both these settings led to similar results on Chairs, but a worse outcome on Sintel and KITTI.

Regarding the number of recurrent iterations during training, we perform 4 recurrent iterations per scale in case of Muun-RAFT, as mentioned in the main paper. This is different than the setting in the MS-RAFT backbone [1], which uses [4, 6, 8] iterations from coarsest to the finest scale, respectively. Our investigations suggest that in the *unsupervised* case, the number of recurrent iterations can be fewer without loss of accuracy. Comparing [4,6,8] iterations to [4,4,4], using our settings, the results on Chairs and Sintel are on-par, but fewer iterations for KITTI yields even better results. Of course, performing fewer refinement iterations during training also saves both time and memory.

4. Backbone Method

As we mentioned in the paper, we employed the 3-scale MS-RAFT [1] model as the backbone of our multi-scale Muun-RAFT variant, where in [1] its architecture was ablated.

Although MS-RAFT+ [3] as the follow-up work shows that employing four scales by utilizing a finer scale at $(\frac{h}{2}, \frac{w}{2})$ improves the results further, utilizing this variant results in longer training and inference time and a large increase in memory compared to the 3-scale MS-RAFT [1] variant. Moreover, as the unsupervised training requires forward- and backward-pass computations and thus is inherently more time-consuming and memory-intensive (than supervised training), we refrain from using the heavier 4-scale MS-RAFT+ model. Please note that Muun-RAFT still employs the on-demand cost sampling of MS-RAFT+ [3] for training, to save memory.

5. Gradual Upsampling Solely by MU

Note that the $\times 2$ MUs cannot be used to upsample the intermediate flow estimates at each scale sequentially to full resolution, because this requires to perform all the steps from the cost computation to the GRU update in order to update the MU masks, for each iteration, for each flow, and at each scale. Evidently, this is computationally infeasible, as upsampling the intermediate flow to full resolution would be required multiple times per refinement iteration and for all iterations. Using our context-based masks is much more efficient, as they are independent of the matching process and can be computed and applied for all relevant scales.

References

[1] Azin Jahedi, Lukas Mehl, Marc Rivinius, and Andrés Bruhn. Multi-scale raft: Combining hierarchical concepts for learning-based optical flow estimation. In *Proc. IEEE International Conference on Image Processing (ICIP)*, pages 1236–1240, 2022.

[2] Azin Jahedi, Maximilian Luz, Marc Rivinius, and Andrés Bruhn. CCMR: High resolution optical flow estimation via coarse-to-fine context-guided motion reasoning. In *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 6899–6908, 2024.

[3] Azin Jahedi, Maximilian Luz, Marc Rivinius, Lukas Mehl, and Andrés Bruhn. MS-RAFT+: High resolution multi-scale RAFT. *International Journal of Computer Vision (IJCV)*, 132 (5):1835–1856, 2024.

[4] Rémi Marsal, Florian Chabot, Angélique Loesch, and Hichem Sahbi. BrightFlow: brightness-change-aware unsupervised learning of optical flow. In *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2060–2069, 2023.

[5] Austin Stone, Daniel Maurer, Alper Ayvaci, Anelia Angelova, and Rico Jonschkowski. Smurf: Self-teaching multi-frame unsupervised *raft* with full-image warping. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3887–3896, 2021.

[6] Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *Proc. European Conference on Computer Vision (ECCV)*, pages 402–419, 2020.

[7] Shuzhi Yu, Hannah Halin Kim, Shuai Yuan, and Carlo Tomasi. Unsupervised flow refinement near motion boundaries. *arXiv preprint arXiv:2208.02305*, 2022.