CRAFT: Cross-Attentional Flow Transformer for Robust Optical Flow
Appendix

1. Model Size and FLOPs

Table 1 presents the number of parameters and FLOPs of RAFT, GMA and CRAFT. The "Ratio" columns take RAFT as the base. FLOPs are measured while inference on Sintel images (1024x436 pixels).

Similar to the SS Transformer and the Cross-Frame Attention, the GMA module in CRAFT is implemented with Expanded Attention [4], which contains multiple modes. In Table 1, two CRAFT models with different numbers of GMA modes m are presented. These two models have very similar overall performance (Table 3). When $m = 2$, the model size is only 7% and 19% larger than GMA and RAFT, respectively, and thus the performance gain is not to be explained away as merely having more parameters.

	Params (M)	Ratio	FLOPs (G)	Ratio
RAFT	5.3	1	369	1
GMA	5.9	1.11	494	1.34
$\text{CRAFT}_{m=2}$	6.3	1.19	613	1.66
$\text{CRAFT}_{m=4}$	6.3	1.19	794	2.15

Table 1. Number of parameters / FLOPs (on Sintel images). m of CRAFT denotes the number of modes in the GMA module.

As shown in Table 2, the GMA module dominates the total overhead of the three transformer modules. This drastic difference is because the GMA module is applied in every iteration of the iterative motion refinement [7], while the other two are only applied once. In this regard, for the FLOPs computation above, we fixed SS Trans and CFA to have 4 modes, and only varied the number of modes of the GMA module.

	SS Trans	CFA	GMA	Remaining
FLOPs (G)	66	6.6	317	405

Table 2. FLOPs (on Sintel images) of different components in $\text{CRAFT}_{m=4}$. SS Trans, CFA and GMA all have 4 modes.

2. Image Shifting as Augmentation

To explore how manually shifting some training images impacts the model performance, we take it as an extra augmentation, namely "ShiftAug", for the training of optical flow models.

2.1. Mild ShiftAug

In the mild ShiftAug training, 10% of the training batches are shifted by $(\Delta u, \Delta v)$ sampled from two Laplacian distributions with scales 16 and 10 (the mean value of a Laplacian distribution is the scale[1]), respectively.

We trained two GMA models and two $\text{CRAFT}_{m=2}$ models, with and without ShiftAug, respectively. They are denoted as GMA, GMA-shift, CRAFT and CRAFT-shift.

2.1.1 Leaderboard Evaluation

Table 3 presents the performance scores of the two $\text{CRAFT}_{m=2}$ models (with or without ShiftAug), along with the standard $\text{CRAFT}_{m=4}$ (without ShiftAug), on Sintel and KITTI leaderboards.

Without ShiftAug, when m reduces from 4 to 2, the performance on large motions ("s40+" for Sintel, and "Fl-fg" for KITTI) degrades slightly. As expected, ShiftAug recovers the model performance on large motions on Sintel (Clean) and KITTI. However, it is surprising to see that with ShiftAug, the performance on large motions on Sintel (Final) degrades slightly.

Due to the restricted frequency of submissions to the leaderboards, we were unable to evaluate more model settings before the camera ready deadline, such as $\text{CRAFT}_{m=4}$ and GMA trained with ShiftAug.

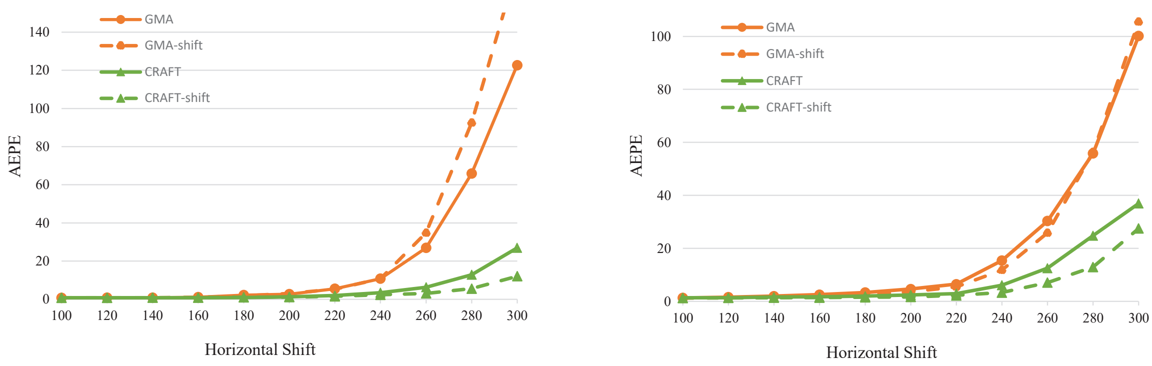
2.1.2 Performance under Image Shifting Attack

We evaluated the 4 models under the image shifting attack, to study whether ShiftAug makes models more robust against it. Figure 1 presents the performance of of the four models, evaluated on the training split of Sintel (Clean) and Sintel (Final), under varying $(\Delta u, \Delta v)$. The horizontal shift $\Delta u \in [100, 300]$, and the vertical shift $\Delta v \doteq \frac{1}{2}\Delta u$.

[1] https://en.wikipedia.org/wiki/Laplace_distribution

Settings	Sintel						KITTI		
	Clean			Final			Fl-bg	Fl-fg	Fl-all
	All	s10-40	s40+	All	s10-40	s40+	(%)	(%)	(%)
CRAFT$_{m=4}$	1.45	0.97	8.30	2.43	1.74	**13.27**	4.58	**5.85**	4.79
CRAFT$_{m=2}$	1.44	0.99	8.13	**2.42**	**1.62**	13.66	**4.30**	6.87	4.73
CRAFT-shift$_{m=2}$	**1.40**	**0.96**	**7.84**	2.51	1.73	14.02	4.35	6.35	**4.68**

Table 3. **Additional results on Sintel and KITTI 2015 leaderboards.** We report the average end-point error (AEPE) for Sintel, and the Fl-bg, Fl-fg and Fl-all metrics for KITTI, which are the percentages of optical flow outliers (pixels with significant flow errors), calculated on the foreground regions and all pixels, respectively. To show the performance on large motions, we present the AEPE on s10-40 and s40+, i.e., pixels whose velocities are within [10, 40] pixels, and > 40 pixels, respectively.



(a) Sintel (Clean)

(b) Sintel (Final)

Figure 1. The AEPE of RAFT, GMA and CRAFT change differently with the magnitude of image shifts. (a)-(c) are on Sintel (Clean), Sintel (Final) and Slow Flow, respectively. The horizontal shift Δu change from 100 to 300, and the vertical shift $\Delta v \doteq \frac{1}{2}\Delta u$. When Δu goes beyond 160, RAFT and GMA quickly deteriorate, and CRAFT performs much more robustly.

It can be seen that, under the image shifting attack, GMA-shift and CRAFT-shift follow similar performance curves as GMA and CRAFT, respectively. On both Sintel (Clean) and Sintel (Final), CRAFT-shift yields significant smaller AEPE (20-50%) on very large shifts ($\geq$ 200 pixels). However, on Sintel (Final), GMA-shift yields almost identical AEPE on very large shifts as GMA. Bewilderingly, on Sintel (Clean), the AEPE of GMA-shift becomes significantly higher on very large shifts than GMA. ShiftAug helps both GMA and CRAFT reduce AEPE on medium-to-large shifts (120-180 pixels). Without ShiftAug, the maximum AEPE in this range is 3.4 pixels.

Based on the observations above, we conclude that mild ShiftAug does not help make GMA more robust against very large image shifts. On the other hand, mild ShiftAug already helps make CRAFT significantly more robust against very large image shifts.

2.2. Aggressive ShiftAug against Image Shifting

In this section, we aim to test the effects of more aggressive ShiftAug, i.e., with a probability of 10%, shift frame 1 by $(\Delta x, \Delta y)$, where Δx is uniformly drawn from $[-320, 320]$, and Δy is uniformly drawn from $[-160, 160]$.

Δx (px)	0	240	280	320	360	400
GMA	1.19	120.2	270.2	362.7	427.9	478.1
CRAFT	**1.11**	10.8	51.8	138.4	236.5	336.9
GMA-shift	1.21	1.83	3.11	9.33	47.5	124.3
CRAFT-shift	1.13	**1.59**	**1.74**	**2.15**	**6.73**	**32.0**

Table 4. **AEPE on Chairs under image shifting attack**.

We fine-tuned CRAFT and GMA on Things models (pretrained on C+T) with such random shifting. Then we evaluated the two models on Chairs[2] with varying degree of shifting. GMA-shift becomes more robust against image shifting attack, but CRAFT-shift still outperforms GMA-shift with a large margin on larger shifts.

2.3. Innate and Acquired Robustness

Based on the observations in Section 2.1 and 2.2, we hypothesize that there are two types of model robustness – *innate* and *acquired* robustness. The latter is learned

[2]We intentionally chose Chairs for evaluation, which has a slight domain gap with Things (used for training), to see how the robustness generalizes.

through augmentation, but stronger innate robustness of CRAFT makes it robust to variations beyond the training data. Hence, CRAFT is likely more robust against other unseen image variations as well, e.g., rotations, lighting variations and motion blur.

3. Iterative Motion Refinement on Shifted Slow-Flow Images

Figure 2 presents the flow fields estimated at different iterations by GMA and CRAFT (both without shift augmentation), respectively, on the same Slow Flow [3] image pair as in Figure 4 of the main text. It partially explains why the AEPE (average end-point error) of GMA is huge when the image shift is large (Figure 5 in the main text).

The flow field is estimated through iterative refinement of $N = 12$ iterations in both GMA and CRAFT. At the $i + 1$-th iteration, it uses the flow estimated at the i-th iteration as initialization, and attempts to estimate a more accurate flow field. This is effective when the flow errors are confined in very small areas, in which cases the model can correct the errors by considering the estimated motions of the surrounding pixels (which are largely accurate). However, if large errors appear in broader areas, the model may fail to recover from the errors with more iterations. Therefore, a relatively small AEPE at the first iteration is crucial for achieving a small AEPE after all iterations. In Figure 2, the flow estimated by GMA at the first iteration has huge errors (AEPE = 253.8), and thus even with more iterations, GMA is unable to recover. In contrast, the flow estimated by CRAFT at the first iteration has a much smaller AEPE = 113.8, and CRAFT quickly corrects the errors.

4. Visualization of Correlation Volumes on Slow Flow

Figures 3-5 present more visualizations of the correlation volume. Figure 3 visualizes the correlation volume of RAFT, GMA and CRAFT on the shifted image pair from Slow Flow. This is the same example as in Figure 2, and Figure 4 in the main text. It can be seen that, GMA has the most spurious high correlations, and CRAFT has the least.

Figures 4 and 5 visualize the correlation volumes with two different query points in Frame 1, on the rider's body, and on the horse's tail, respectively, on the original image pair from Slow Flow. Similar distributions of spurious high correlations are observed. Among the four models, CRAFT (with SS trans) always has the least spurious high correlations, showing that it is able to greatly suppress spurious correlations and compute a more accurate correlation volume, which may explain its robustness demonstrated in Figure 2.

5. Screenshots of Sintel and KITTI Leaderboards

Figures 6-8 are the screenshots of the Sintel (Final), Sintel (Clean) and the KITTI-2015 optical flow leaderboards, which were taken near the CVPR'2022 submission deadline.

CRAFT ranked the 1st and 5th places on Sintel (Final pass), Sintel (Clean pass), respectively. As Sintel (Final pass) images contain more light variations, shadows, motion blurs, etc. that are common in real world, we argue that the performance on Sintel (Final) better reflects the performance of a model on real-world images. Evidence has been presented in the Sintel paper (Figure 5, [2]) that Sintel (Final) has similar image and motion statistics as other real-world datasets, including Lookalikes [2] and Middlebury [1].

On the KITTI flow-2015 leaderboard, a few methods among the top are scene flow methods (marked with ~~strikethrough text~~) that take two stereo pairs of images as input (cf. two monocular images of optical flow), and thus are not comparable with optical flow methods. Among the top optical flow methods, CRAFT ranks 5th. In particular, it achieves the highest accuracy on foreground regions, measured as the smallest Fl-fg (percentage of flow outliers[3] in the foreground regions). On Fl-all (percentage of flow outliers in both foreground and background regions), MixSup ranks as the top-1 optical flow method, but its training and implementation details are missing for further analysis and comparison with CRAFT. Separable Flow [8] and RFPM [5] have significantly worse performance on foreground regions. RAFT-A uses Autoflow [6] as the pretraining data, and thus not directly comparable with CRAFT. Autoflow explore a new way to synthesise training data, which is orthogonal to our method or other recent architectures. It is worth noting that, the foreground objects in KITTI are usually cars, pedestrians, etc., which naturally are more important than the background. Thus, smaller Fl-fg are probably more important for practical applications than smaller Fl-bg or Fl-all.

[3]Pixels whose end-point error is > 3 pixels or 5% of the ground truth flow magnitude.

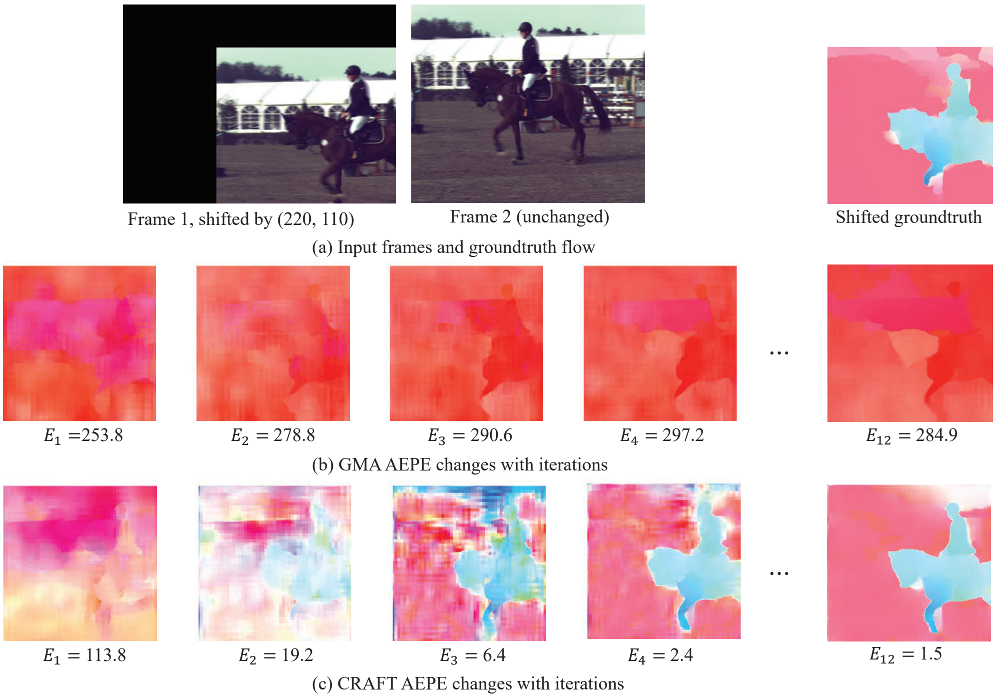


Figure 2. The iterative refinement of optical flow on the shifted Slow Flow image pair (Figure 4 in the main text), by GMA and CRAFT, respectively. As in the first iteration, GMA makes excessively huge errors, it is unable to recover with more iterations. CRAFT recovers from smaller initial errors and yields an accurate flow field eventually.

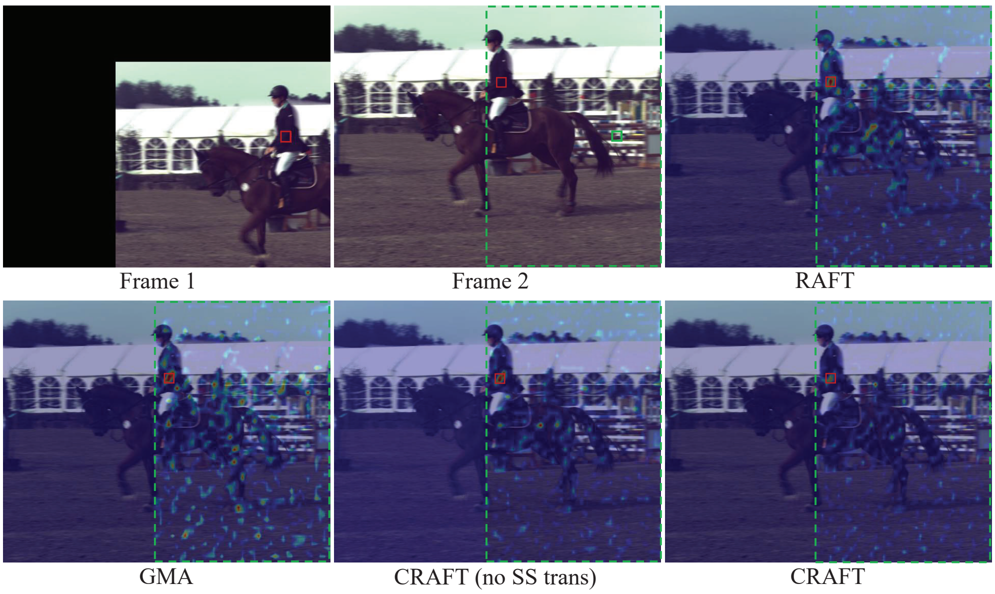


Figure 3. Heatmaps of the correlation matrices between Frame 2 and a query point on the rider's body in a shifted Frame 1, on the Slow Flow dataset. At the presence of motion blur, CRAFT has significantly fewer noisy correlations than RAFT and GMA, showing its robustness. Removing SS transformer results in more noisy correlations.

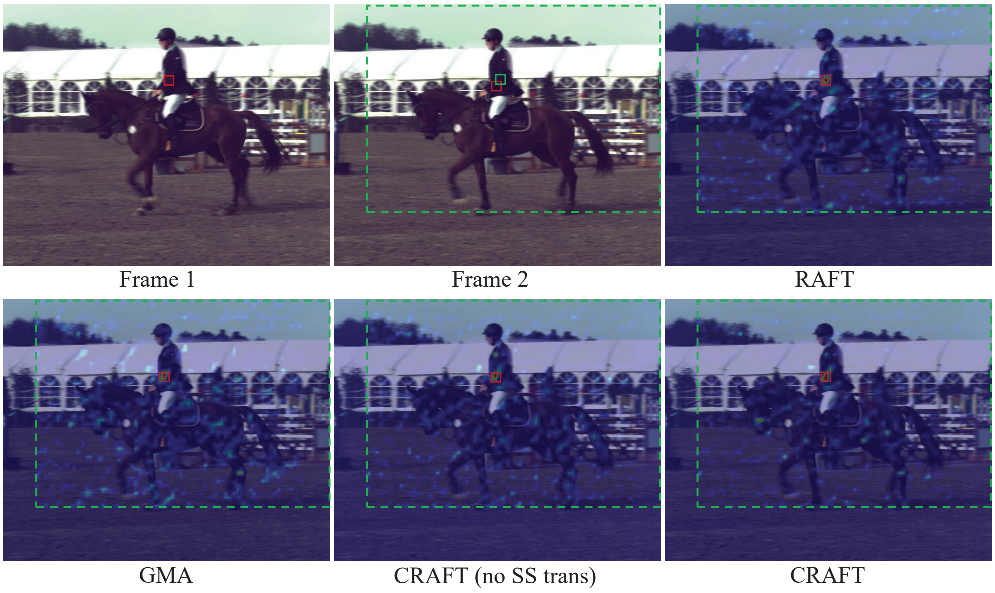


Figure 4. Heatmaps of the correlation matrices between Frame 2 and a query point on the rider's body in Frame 1, on Slow Flow.

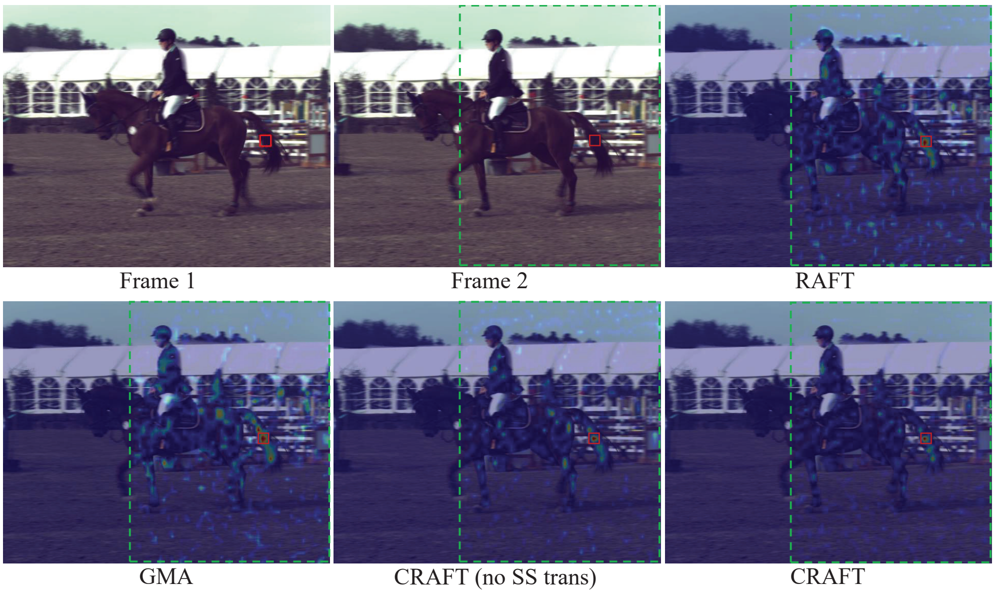
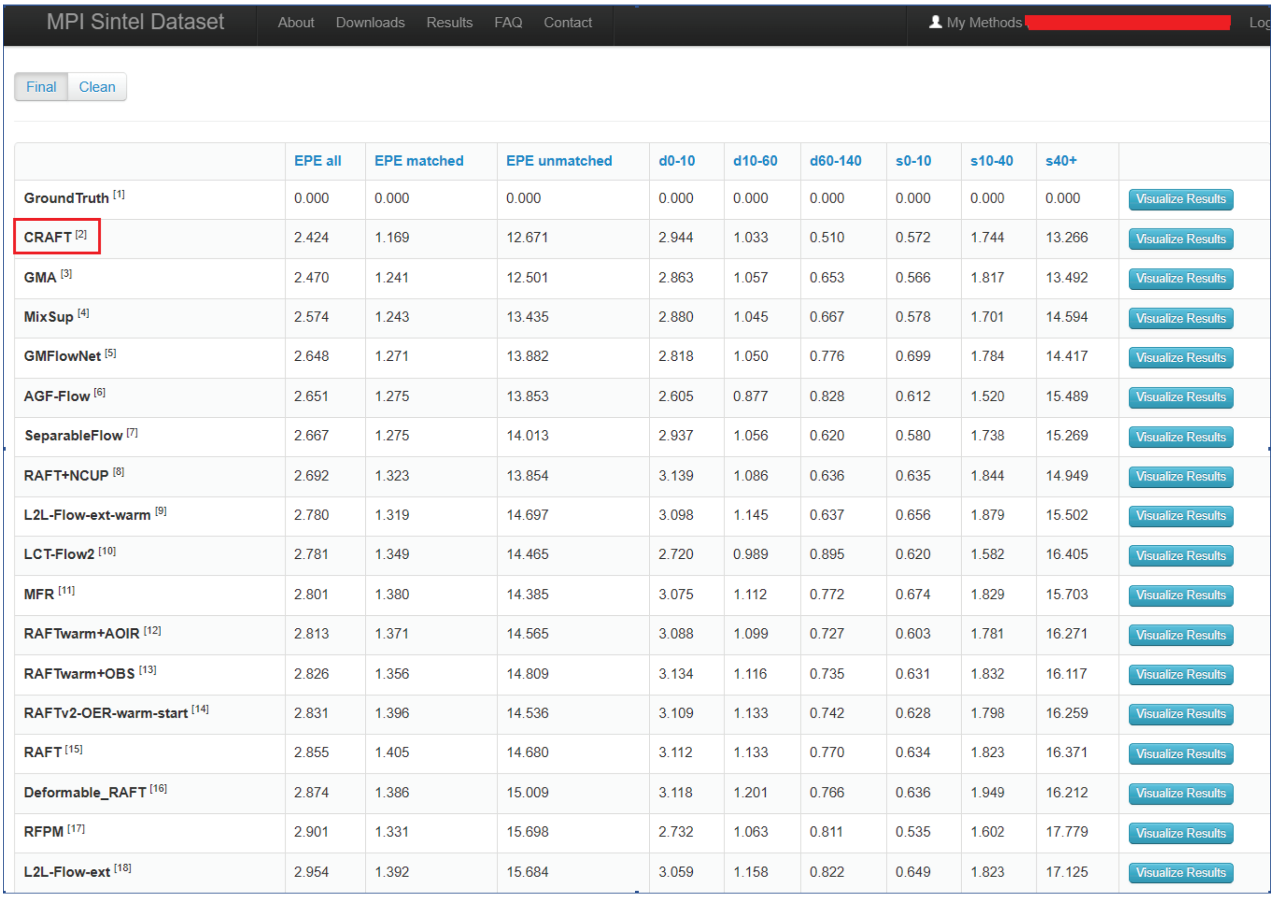


Figure 5. Heatmaps of the correlation matrices between Frame 2 and a query point on the horse's tail in Frame 1, on the Slow Flow dataset.



	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
CRAFT [2]	2.424	1.169	12.671	2.944	1.033	0.510	0.572	1.744	13.266	Visualize Results
GMA [3]	2.470	1.241	12.501	2.863	1.057	0.653	0.566	1.817	13.492	Visualize Results
MixSup [4]	2.574	1.243	13.435	2.880	1.045	0.667	0.578	1.701	14.594	Visualize Results
GMFlowNet [5]	2.648	1.271	13.882	2.818	1.050	0.776	0.699	1.784	14.417	Visualize Results
AGF-Flow [6]	2.651	1.275	13.853	2.605	0.877	0.828	0.612	1.520	15.489	Visualize Results
SeparableFlow [7]	2.667	1.275	14.013	2.937	1.056	0.620	0.580	1.738	15.269	Visualize Results
RAFT+NCUP [8]	2.692	1.323	13.854	3.139	1.086	0.636	0.635	1.844	14.949	Visualize Results
L2L-Flow-ext-warm [9]	2.780	1.319	14.697	3.098	1.145	0.637	0.656	1.879	15.502	Visualize Results
LCT-Flow2 [10]	2.781	1.349	14.465	2.720	0.989	0.895	0.620	1.582	16.405	Visualize Results
MFR [11]	2.801	1.380	14.385	3.075	1.112	0.772	0.674	1.829	15.703	Visualize Results
RAFTwarm+AOIR [12]	2.813	1.371	14.565	3.088	1.099	0.727	0.603	1.781	16.271	Visualize Results
RAFTwarm+OBS [13]	2.826	1.356	14.809	3.134	1.116	0.735	0.631	1.832	16.117	Visualize Results
RAFTv2-OER-warm-start [14]	2.831	1.396	14.536	3.109	1.133	0.742	0.628	1.798	16.259	Visualize Results
RAFT [15]	2.855	1.405	14.680	3.112	1.133	0.770	0.634	1.823	16.371	Visualize Results
Deformable_RAFT [16]	2.874	1.386	15.009	3.118	1.201	0.766	0.636	1.949	16.212	Visualize Results
RFPM [17]	2.901	1.331	15.698	2.732	1.063	0.811	0.535	1.602	17.779	Visualize Results
L2L-Flow-ext [18]	2.954	1.392	15.684	3.059	1.158	0.822	0.649	1.823	17.125	Visualize Results

Figure 6. Screenshot of Sintel (Final) leaderboard, taken near the CVPR'2022 submission deadline.

MPI Sintel Dataset About Downloads Results FAQ Contact My Methods [redacted] Log

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
GMA [2]	1.388	0.582	7.963	1.537	0.461	0.278	0.331	0.963	7.662	Visualize Results
GMFlowNet [3]	1.390	0.520	8.486	1.275	0.395	0.293	0.314	0.991	7.698	Visualize Results
RFPM [4]	1.411	0.494	8.884	1.335	0.400	0.221	0.273	0.879	8.345	Visualize Results
MixSup [5]	1.419	0.541	8.574	1.455	0.442	0.242	0.301	0.940	8.118	Visualize Results
CRAFT [6]	1.453	0.589	8.501	1.565	0.490	0.257	0.312	0.966	8.295	Visualize Results
SeparableFlow [7]	1.496	0.567	9.075	1.474	0.481	0.257	0.309	0.958	8.691	Visualize Results
RAFTwarm+AOIR [8]	1.544	0.551	9.656	1.515	0.412	0.280	0.279	0.941	9.290	Visualize Results
MFR [9]	1.545	0.593	9.295	1.536	0.477	0.299	0.348	1.023	8.736	Visualize Results
CosTR [10]	1.545	0.519	9.908	1.293	0.403	0.261	0.321	0.917	9.136	Visualize Results
RAFTwarm+OBS [11]	1.593	0.600	9.692	1.532	0.507	0.309	0.300	0.989	9.470	Visualize Results
RAFTv2-OER-warm-start [12]	1.594	0.625	9.487	1.567	0.512	0.339	0.328	1.014	9.271	Visualize Results
COMBO [13]	1.604	0.569	10.046	1.455	0.468	0.302	0.318	1.013	9.422	Visualize Results
RAFT [14]	1.609	0.623	9.647	1.621	0.518	0.301	0.341	1.036	9.288	Visualize Results
NASFlow-RAFT [15]	1.613	0.503	10.664	1.339	0.405	0.238	0.298	0.892	9.883	Visualize Results
L2L-Flow-ext-warm [16]	1.648	0.622	10.017	1.641	0.516	0.282	0.342	1.018	9.657	Visualize Results
RAFT+NCUP [17]	1.661	0.678	9.666	1.872	0.541	0.302	0.371	1.102	9.402	Visualize Results
MF2C [18]	1.664	0.689	9.612	1.663	0.596	0.372	0.348	1.060	9.651	Visualize Results

Figure 7. Screenshot of Sintel (Clean) leaderboard, taken near the CVPR'2022 submission deadline.

Evaluation ground truth All pixels Evaluation area All pixels

	Method	Setting	Code	Fl-bg	Fl-fg	Fl-all	Density	Runtime	Environment	Compare
1	~~CamLiFlow~~			**2.38 %**	7.35 %	**3.21 %**	100.00 %	1 s	GPU @ 2.5 Ghz (Python + C/C++)	
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	~~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.										
4	~~LPSF~~			3.18 %	9.92 %	4.31 %	100.00 %	60 s	1 core @ 2.5 Ghz (C/C++)	
5	MixSup			3.99 %	6.01 %	4.33 %	100.00 %	0.2 s	1 core @ 2.5 Ghz (Python)	
6	SeparableFlow			4.32 %	6.24 %	4.64 %	100.00 %	0.25 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.										
7	~~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.										
8	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.										
9	RFPM		code	4.50 %	6.20 %	4.79 %	100.00 %	0.2 s	GPU @ 2.5 Ghz (Python)	
10	$\boxed{\text{CRAFT}}$			4.58 %	$\boxed{\textbf{5.85 \%}}$	4.79 %	100.00 %	0.1 s	GPU @ 2.5 Ghz (Python)	
11	AGFlow			4.52 %	6.75 %	4.89 %	100.00 %	0.2 s	8 cores @ 2.5 Ghz (Python)	
12	L2L-Flow			4.48 %	6.96 %	4.89 %	100.00 %	0.2 s	GPU @ 2.5 Ghz (Python)	
13	CRAFT-noca			4.65 %	6.15 %	4.90 %	100.00 %	0.1 s	1 core @ 2.5 Ghz (Python)	
14	Raft_bl			4.50 %	6.87 %	4.90 %	100.00 %	1 s	1 core @ 2.5 Ghz (Python)	
15	RAFT+OBS			4.45 %	7.19 %	4.91 %	100.00 %	0.2 s	GPU @ 2.5 Ghz (Python)	
16	GMA_p			4.58 %	6.71 %	4.93 %	100.00 %	0.2 s	GPU @ 2.5 Ghz (Python + C/C++)	
17	RAFTv2-OER			4.72 %	6.66 %	5.04 %	100.00 %	0.1541 s	NVIDIA 2080Ti (Python)	
18	CRAFT-nof2			4.80 %	6.41 %	5.06 %	100.00 %	0.5 s	1 core @ 2.5 Ghz (Python)	
19	RAFT+AOIR			4.68 %	6.99 %	5.07 %	100.00 %	10 s	GPU @ 2.5 Ghz (Python + C/C++)	
L. Mehl, C. Beschle, A. Barth and A. Bruhn: An Anisotropic Selection Scheme for Variational Optical Flow Methods with Order-Adaptive Regularisation. SSVM 2021.										
20	GMA_(p+c)			4.81 %	6.45 %	5.08 %	100.00 %	0.2 s	GPU @ 2.5 Ghz (Python)	
21	GMA-FER_s_23			4.77 %	6.70 %	5.09 %	100.00 %	0.2 s	1 core @ 2.5 Ghz (Python)	
22	RAFT		code	4.74 %	6.87 %	5.10 %	100.00 %	0.2 s	GPU @ 2.5 Ghz (Python)	
Z. Teed and J. Deng: RAFT: Recurrent All-Pairs Field Transforms for Optical Flow. ECCV 2020.										
23	RAFT+NCUP			4.78 %	6.93 %	5.14 %	100.00 %	0.2 s	GPU @ 2.5 Ghz (Python)	
24	GMA			4.78 %	7.03 %	5.15 %	100.00 %	0.2 s	GPU @ 2.5 Ghz (Python)	

Figure 8. Screenshot of KITT-2015 leaderboard, taken near the CVPR'2022 submission deadline. Five scene flow methods are marked with ~~strikethrough text~~, as they are not comparable to optical flow methods. There remain the top 19 optical flow methods. "CRAFT-noca" and "CRAFT-nof2" are the ablated models of removing the Cross-Frame Attention, and the Semantic Smoothing Transformer from CRAFT, respectively.

References

[1] Simon Baker, Daniel Scharstein, JP Lewis, Stefan Roth, Michael J Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International journal of computer vision*, 92(1):1–31, 2011. 3

[2] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *ECCV*, 2012. 3

[3] Joel Janai, Fatma Güney, Jonas Wulff, Michael Black, and Andreas Geiger. Slow flow: Exploiting high-speed cameras for accurate and diverse optical flow reference data. In *CVPR*, 2017. 3

[4] Shaohua Li, Xiuchao Sui, Xiangde Luo, Xinxing Xu, Yong Liu, and Rick Goh. Medical image segmentation using squeeze-and-expansion transformers. In *IJCAI*, 2021. 1

[5] Libo Long and Jochen Lang. Detail preserving residual feature pyramid modules for optical flow. *arXiv:2107.10990*, 2021. 3

[6] D. Sun, D. Vlasic, C. Herrmann, V. Jampani, M. Krainin, H. Chang, R. Zabih, W. T. Freeman, and C. Liu. Autoflow: Learning a better training set for optical flow. In *CVPR*, 2021. 3

[7] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 1

[8] Feihu Zhang, Oliver J. Woodford, Victor Adrian Prisacariu, and Philip H.S. Torr. Separable flow: Learning motion cost volumes for optical flow estimation. In *ICCV*, 2021. 3