Sensor-Guided Optical Flow
Supplementary material

Matteo Poggi Filippo Aleotti Stefano Mattoccia
Department of Computer Science and Engineering (DISI)
University of Bologna, Italy
{m.poggi, filippo.aleotti2, stefano.mattoccia }@unibo.it

In this document, we report ablation studies and qualitative analysis of results shown in ICCV 2021 paper "Sensor-Guided Optical Flow". Specifically, in Sec. 1 we recall the colormaps used to show qualitative results in both the main paper and this document, in Sec. 2 we give an intuitive overview about the role of resolution when implementing the guided flow framework, in Sec. 3 we report ablation studies concerning the hyper-parameters k, c in our framework, as well as the choice of guide density and noise intensity used in the main paper, Sec. 4 collects more qualitative and quantitative results concerning the different strategies employed to obtain flow hints from a real sensor suite, and their effect on QRAFT when used as a guide. Finally, Sec. 5 shows more qualitative examples of our sensor-guided optical flow framework – and also highlights some failure cases – employing the depth sensor available in the Apple iPhone Xs.

1. Colormaps

We use the popular colormaps shown in Fig. 1 to visualize both flow vectors and their error compared to ground-truth qualitatively. To encode flow direction and magnitude, we use the color wheel from [1], while to encode flow end-point-error (EPE) we adopt the colormap used by the KITTI 2015 online benchmark [3]. All images encoding sparse hints are depicted, both in this document and the main paper, after being processed by means of a 2×2 dilation filter to improve visualization.

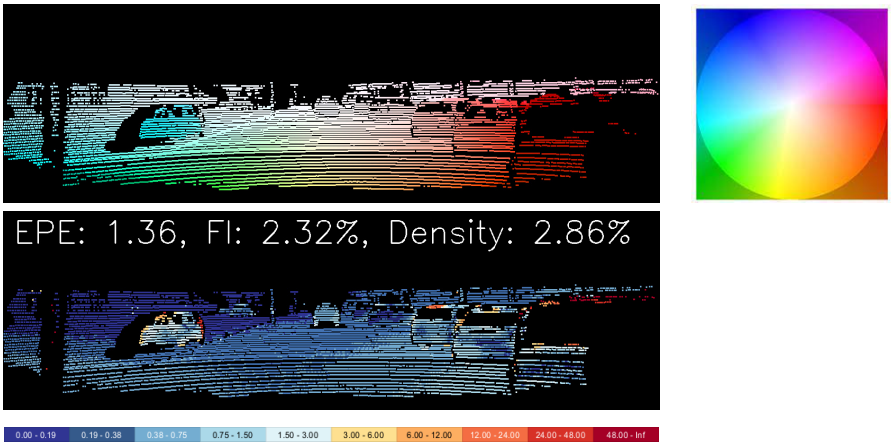

Figure 1. **Colormaps used for qualitative results.** On top, flow hints and corresponding color wheel. On bottom, error map and corresponding encoding.

2. On the importance of resolution

As we have shown in the main paper, the guided flow framework results much less effective when applied to the original RAFT model [6] (often, ineffective at all), while it consistently improves the results when coupled with QRAFT. Intuitively, we ascribe this behavior to resolution and its relationship with correlation scores that are enhanced by guided flow.

RAFT computes correlations at $\frac{1}{8}$ resolution, thus I) pixels over which correlations are computed correspond to 8×8 patches in the original image and II) flow indices in the correlation volume are multiples of 8 (only the gray ones in Fig. 2). In this setting, I) the guide acts coarsely (1 pixel corresponds to an 8×8 patch) and II), less intuitively, the flow guide acts by enhancing the nearest quantized flow indices pair in the correlation volume - *e.g.*, a flow hint of (11.5, 5) peaks its closest location on the grid, (8, 8) in black in Fig. 2, that has an EPE of 4.6 with respect to the real hinted value.

With QRAFT and correlations computed at $\frac{1}{4}$ resolution, I) the guide acts less coarsely (1 pixel$= 4 \times 4$ patch) and II) flow indices are now multiples of 4 (both gray and orange ones) – the flow hint (11.5, 5) will peak the location (12, 4) (in red in Fig. 2), introducing a much lower EPE, 1.12, thus making the guidance more effective. Plausibly, a variant working at $\frac{1}{2}$ res could benefit even more from the guide, yet can not fit in our single GPU at the moment. However, QRAFT is more than enough to exploit the guide when trained on any dataset and consistently improves the results.

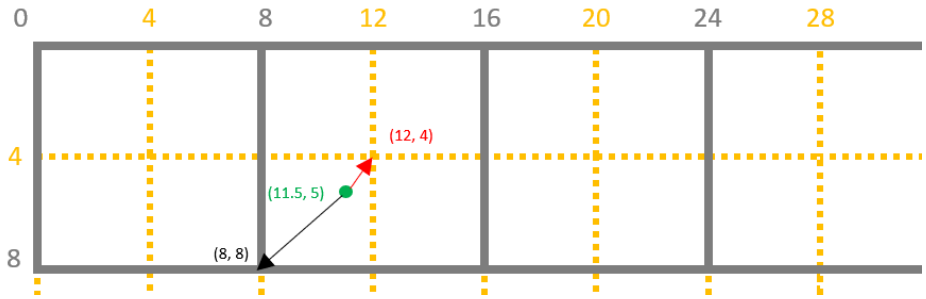

Figure 2. **Correlation indices and guide approximation.** By computing correlation scores at higher resolution, flow hints enhance the scores at flow locations closer to the real value.

3. Guided optical flow – ablation studies

3.1. Hyper-parameters tuning

We report a grid-search analysis for hyper-parameters (k, c) used to define the bivariate Gaussian function at the core of the guided optical flow framework. This study is carried out on both the original RAFT architecture [6], and QRAFT. Following [4], we have evaluated the performance of different (k, c) configurations by applying the guided optical flow framework to pre-trained models (*i.e.*, trained conventionally without flow hints). Then, the best configuration has been chosen to train from scratch networks again, exploiting flow hints, for the experiments reported in the main paper in Tabs. 2 and 4.

To this aim, we have selected two instances of RAFT and QRAFT trained on the FlyingChairs dataset, respectively, with batch sizes 6 and 2. Then, we evaluated both networks on the KITTI 142 split, *i.e.* considering the setting with the largest domain-shift – thus, where the guided optical flow framework yields the most significant accuracy improvement. This methodology allows us to appreciate the accuracy difference better when varying k and c. In this experiment, $\sim 3\%$ ground-truth labels have been sampled as flow hints and, in order to measure the *ideal* gain attainable by each (k, c) configuration, *noise has not been applied* to the flow guide.

Fig. 3 collects the outcome of this evaluation. We can notice that setting $k = 10$ and $c = 1$ leads to the best results with both architectures, with gradual drops in accuracy occurring while moving farther from the best configuration and disruptive results with larger values of k and c.

3.2. Varying density

We study the behavior of guided RAFT and QRAFT, varying the density of the guide. To this aim, we train different instances of both networks on FlyingChairs with increasing density of the sampled guide, respectively 1%, 3% and 5% and evaluate them on the KITTI 142 with varying density of sampled guide. As in the previous experiment, we do not perturb the sampled guide to measure the *ideal* difference in performance caused by the difference in density. Tab. 1 collects the outcome of this experiment, reporting results concerning RAFT (left) and QRAFT (right).

Concerning RAFT, training the network with $\sim 1\%$ of hints yields the best improvement even in the presence of a more significant amount of hints at testing time, except on EPE when testing with $\sim 5\%$ hints. Increasing the guide density at training time makes guided RAFT less effective when using a lower density guide at testing time and when the same density is maintained. We ascribe this effect to the coarser influence of the guide applied at the eighth resolution, strongly limiting its effect and the possibility to exploit a higher number of hints. Focusing on QRAFT, we can notice how, given

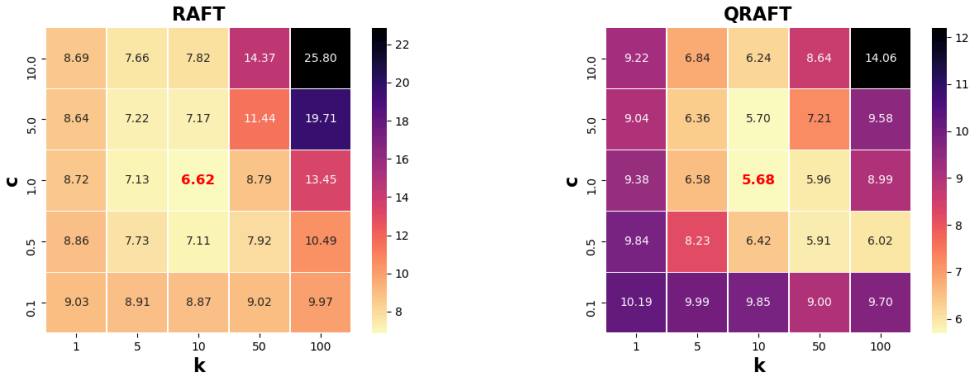

Figure 3. **Hyper-parameters tuning – k and c.** Experiments with RAFT (left) and QRAFT (right) trained on FlyingChairs and tested on KITTI 142 split. We report EPE achieved by each (k, c) configuration (baseline RAFT and QRAFT accuracy: 8.77 and 9.61).

RAFT

	Guide Density	KITTI 142 EPE Testing 1%	EPE Testing 3%	EPE Testing 5%	Fl (%) Testing 1%	Fl (%) Testing 3%	Fl (%) Testing 5%
Training	1%	**6.30**	**5.37**	4.96	**28.87**	**25.44**	**23.88**
	3%	6.62	**5.37**	6.01	37.89	32.13	28.71
	5%	8.34	5.52	**4.51**	44.75	31.18	25.38

QRAFT

	Guide Density	KITTI 142 EPE Testing 1%	EPE Testing 3%	EPE Testing 5%	Fl (%) Testing 1%	Fl (%) Testing 3%	Fl (%) Testing 5%
Training	1%	**4.83**	3.70	3.41	**23.67**	17.92	15.92
	3%	5.29	**3.68**	3.22	29.10	**17.31**	15.98
	5%	5.19	3.91	**3.05**	27.20	17.99	**14.49**

Table 1. **Sensitivity to guide density.** Experiments with RAFT (left) and QRAFT (right), trained on FlyingChairs and tested on the KITTI 142 split, by sampling a different amount of hints at training (from top to bottom) and at testing (from left to right). **Bold:** best results for a given density at testing time (best per column).

a density of hints at testing time, the lowest EPE and Fl errors are obtained by the network trained with the same density. Not surprisingly, increasing the density at testing time accuracy increases, while having fewer hints than the amount used for training reduces the effectiveness of guided optical flow. In general, we found out that training with $\sim 1\%$ hints represents a good compromise. It enables the most noticeable improvement with fewer hints at testing time yet can improve when additional ones are available.

3.3. Varying noise

We study the behavior of guided RAFT and QRAFT varying intensity of noise applied to sampled guide. To this aim, we train different instances of both networks on FlyingChairs3D with increasing intensity of noise applied to the guide, respectively no noise, $[-1, 1]$, $[-2, 2]$ and $[-3, 3]$ applied to both (x,y) flow hints components and evaluate them on the KITTI 142 with varying noise as well. We train all networks with $\sim 1\%$ hints density and test with about $\sim 3\%$ density, *i.e*. the one closer to the number of hints obtained by our *real* pipeline. Tab. 2 reports the errors, EPE and Fl, of the guide sampled from ground-truth with increasing intensity of noise.

	KITTI 142 EPE ✗	EPE $[-1,1]$	EPE $[-2,2]$	EPE $[-3,3]$	Fl (%) ✗	Fl (%) $[-1,1]$	Fl (%) $[-2,2]$	Fl (%) $[-3,3]$
Sampled Guide	0.00	0.77	1.53	2.30	0.00	0.00	0.00	18.12

Table 2. **Hints accuracy varying the magnitude of the noise.** Experiments on KITTI 142 split (density of sampled hints: 2.89%).

We can notice how, by increasing the intensity of noise, the EPE rises. Concerning Fl, it remains zero until we apply a $[-3, 3]$ noise. Indeed, being the Fl computed as the percentage of pixels having error larger than 3 pixels, this cannot occur with $[-2, 2]$ noise level or lower, since the maximum error that can occur on hints perturbed by $[-Z, Z]$ noise is equal to $Z\sqrt{2}$. Tab. 3 reports the impact of noise on performance, showing results concerning both RAFT (left) and QRAFT (right).

This time, we can notice a similar behavior for both RAFT and QRAFT. Indeed, training the network with perfect hints (✗ entry) results in the best strategy if perfect hints are also available at testing. However, with increasing noise intensity at

RAFT

Training Guide Noise	KITTI 142 EPE Testing ✗	EPE Testing $[-1,1]$	EPE Testing $[-2,2]$	EPE Testing $[-3,3]$	Fl (%) Testing ✗	Fl (%) Testing $[-1,1]$	Fl (%) Testing $[-2,2]$	Fl (%) Testing $[-3,3]$
✗	**5.37**	5.86	5.86	5.89	**25.44**	29.69	29.80	29.93
$[-1,1]$	5.39	**5.33**	**5.35**	**5.39**	25.50	**25.43**	**25.51**	**25.73**
$[-2,2]$	5.68	5.64	5.64	5.65	26.39	26.37	26.44	26.59
$[-3,3]$	6.49	6.07	6.10	6.11	27.47	27.21	27.32	27.40

QRAFT

Training Guide Noise	KITTI 142 EPE Testing ✗	EPE Testing $[-1,1]$	EPE Testing $[-2,2]$	EPE Testing $[-3,3]$	Fl (%) Testing ✗	Fl (%) Testing $[-1,1]$	Fl (%) Testing $[-2,2]$	Fl (%) Testing $[-3,3]$
✗	**3.70**	**3.75**	**3.99**	4.08	**17.92**	**18.05**	19.51	20.09
$[-1,1]$	3.95	3.96	**3.99**	**4.06**	19.13	19.24	**19.50**	**19.99**
$[-2,2]$	4.06	4.01	4.02	4.08	19.92	19.95	20.10	20.38
$[-3,3]$	4.09	4.09	4.09	4.11	19.84	19.94	19.99	20.14

Table 3. **Sensitivity to guide noise.** Experiments with RAFT (left) and QRAFT (right) trained on FlyingChairs and tested on KITTI 142 split applying a different magnitude of noise to sampled hints at training (from top to bottom) and testing (from left to right). **Bold:** best results for a given noise intensity at testing time (best per column).

testing time, the performance drops. Training both networks with perturbed hints improves the robustness to the noisy guide at testing time. However, conversely to what was observed with density – *i.e*. the higher at training time, the better results with more hints at testing time – the improvement due to using noise at training time saturates when applying $[-1, 1]$ additive noise. Moreover, networks trained in this setting turn more robust to higher magnitudes of noise applied at testing time. In contrast, training with stronger noise leads to worse accuracy in the presence of weaker/stronger perturbations at testing time. Thus, we select networks trained with $[-1, 1]$ noise intensity for the experiments reported in the submitted paper.

3.4. Evaluation excluding pixels with hints

In our experiments concerning the use of hints sampled from ground-truth, we have always perturbed the guide with noise. However, the reader might argue that most of the improvements achieved by the guided optical flow framework are limited to the pixels with available hints – in other words, the network only exploits the hints to correct its prediction for the pixels providing them. As in [4], we show in this section the results obtained by computing error metrics by **exluding** pixels for which hints are provided, thus averaging the error only on the remaining pixels. The outcome of this evaluation carried out with QRAFT is shown in Tab. 4.

	Training Dataset	Network	Sintel Clean ✗	Sintel Clean *guided*	Sintel Final ✗	Sintel Final *guided*	Middlebury Flow EPE ✗	Middlebury Flow EPE *guided*	KITTI 2012 EPE ✗	KITTI 2012 EPE *guided*	KITTI 2012 Fl (%) ✗	KITTI 2012 Fl (%) *guided*	KITTI 142 EPE ✗	KITTI 142 EPE *guided*	KITTI 142 Fl (%) ✗	KITTI 142 Fl (%) *guided*
(b)	C	QRAFT	2.03	**1.13**	3.64	**1.64**	0.50	**0.44**	5.54	**2.96**	25.96	**15.92**	9.61	**4.07**	32.50	**20.05**
(d)	C+T	QRAFT	1.60	**0.86**	2.45	**1.22**	0.29	**0.28**	3.42	**2.08**	14.90	**8.87**	6.21	**3.15**	21.47	**13.34**
(f)	C+T+S	QRAFT	1.38	**0.73**	2.02	**1.01**	0.27	**0.25**	2.74	**1.84**	11.27	**7.59**	5.02	**2.82**	17.53	**11.87**
(h)	C+T+K	QRAFT	4.99	**3.35**	6.15	**3.95**	0.68	**0.54**	1.60	**1.07**	5.32	**3.20**	2.58	**1.23**	6.61	**3.79**

Table 4. **Evaluation – Guided Optical Flow.** Evaluation on Sintel sequences selected for validation (Clean and Final), Middlebury, KITTI 2012 and KITTI 142 split. Results without (✗) or with (*guided*) flow guide. We **exclude** pixels with available hints to compute all errors.

We can notice how, in general, the error metrics are almost identical, with very few fluctuations on the second decimals. This outcome is not surprising since the pixels removed to average metrics are a small percentage of the total pixels. It confirms that the improvement yield by guided optical flow is consistent on the whole image and not limited to pixels with hints.

4. Sensor-Guided optical flow – ablation studies

4.1. Comparison with hints sampled from ground-truth

This section shows a qualitative comparison between the flow hints obtained through our pipeline and those sampled from ground-truth. Fig. 4 reports two examples from the KITTI 142 split, highlighting the two main differences between real and sampled hints. As in the submitted paper, the latter is perturbed with $[-3, 3]$ random noise on both (x,y) flow components.

On frames 000072, we can notice on the one hand how the distribution of the flow guide obtained through the LIDAR is irregular, with large regions of the image for which no hint is provided. On the other hand, hints sampled from ground-truth appear uniformly in the image, despite the overall lower accuracy and the higher average error introduced by the noise perturbation. We can further dig into this latter aspect by looking at frames 000120: we can notice how real hints have average EPE and Fl lower compared to sampled hints – indeed, the random noise is applied to all pixels sampled from ground-truth.

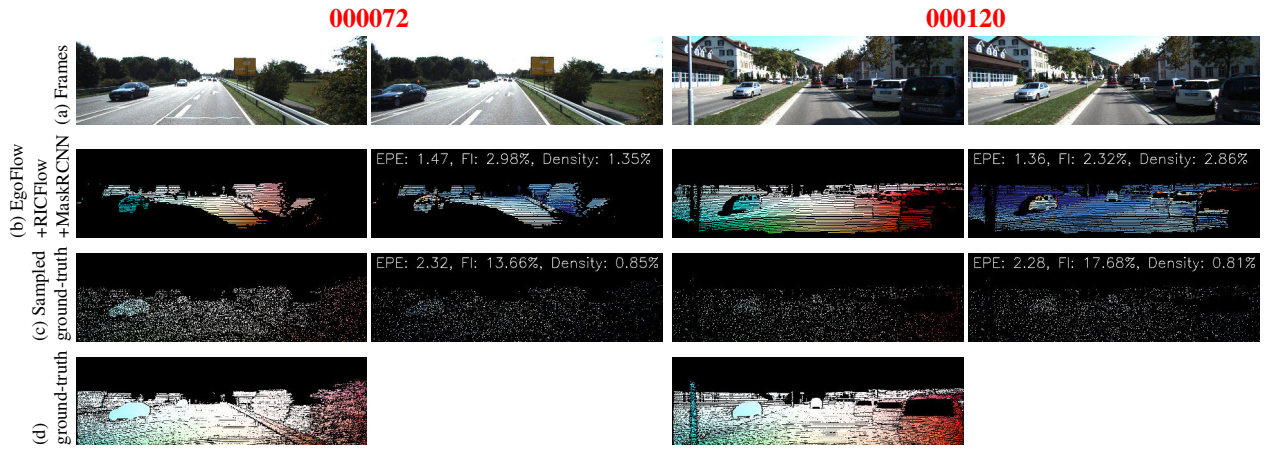

Figure 4. **Comparison between flow hints from a LIDAR vs sampled from ground-truth.** Example of flow hints computed over two pairs of frames from KITTI 142 split. For each example, we show on the top row the reference images (a). Rows (b) and (c) show flow guides (left) and error maps (right), densified for better visualization. Row (d) shows ground-truth flow maps. The actual guide density is reported over each error map, with EPE and Fl computed on pixels with hints and ground-truth available.

However, according to the noise intensity we apply, the maximum EPE in sampled hints is $3\sqrt{2}$. On the contrary, some LIDAR hints have much higher EPE – *e.g.* on car roofs where the EPE is higher than 48 (according to Fig. 1).

To conclude this comparison, we show the predictions by QRAFT when guided with the two different kinds of hints. Fig. 5 collects these results in each training configuration considered in the main paper. Although on frame 000072 (left), the hints produced by our pipeline are irregularly distributed, QRAFT performance is always improved when using them, even in the absence of domain-shift (*i.e.*, C+T+K). However, guiding with hints sampled from ground-truth results in further improvements, confirming our previous analysis.

Although the experiments with sampled hints and noise perturbation show that the guided optical flow framework is effective even when the flow guide is not entirely reliable, it does not perfectly model a real case as the one evaluated with the KITTI Velodyne. Nonetheless, it resulted effective as well. In the future, the possibility of obtaining better flow hints that are closer to the one sampled from ground-truth would further boost our framework.

4.2. Performance with different flow hints

We now compare the different flow guides to appreciate better the importance of retrieving accurate hints for both static regions and independently moving objects.

	Training Dataset	Network	KITTI 142 EPE							KITTI 142 Fl (%)						
						sensor-guided							*sensor-guided*			
			✗	EgoFlow – no filtering	EgoFlow – filtering	RICFlow	EgoFlow +Motion mask [5]	EgoFlow +Motion prob. [7]	EgoFlow +RICFlow +MaskRCNN	✗	EgoFlow – no filtering	EgoFlow – filtering	RICFlow	EgoFlow +Motion mask [5]	EgoFlow +Motion prob. [7]	EgoFlow +RICFlow +MaskRCNN [2]
(a)	C	QRAFT	9.61	10.46	10.35	7.70	6.23	6.71	**5.88**	32.50	33.93	33.03	41.73	26.59	27.05	**25.40**
(b)	C+T	QRAFT	6.21	9.37	8.92	6.53	4.95	5.27	**4.55**	21.47	25.61	24.02	31.74	18.02	18.40	**17.09**
(c)	C+T+S	QRAFT	5.02	9.09	8.67	6.69	4.68	4.98	**4.32**	17.53	24.06	22.55	38.09	16.53	16.85	**15.59**
(d)	C+T+K	QRAFT	2.58	7.72	6.54	3.83	2.43	2.55	**2.08**	6.61	17.48	13.94	13.77	7.04	7.19	**5.97**

Table 5. **Sensor-Guided Optical Flow.** Evaluation on KITTI 142 split, without (✗) or with (*sensor-guided*) hints from different sources.

Tab. 5 collects the outcome of this evaluation. We can notice how the LIDAR alone (EgoFlow), without or with filtering, is not enough to guide QRAFT accurately. It yields some slight improvement on the Fl metric when the domain gap between training and testing data is larger (*i.e.*, after having trained on C alone), but it fails at improving any QRAFT instance trained on C+T, C+T+S or C+T+K. Using RICFlow as a guide results ineffective due to the large errors in the background and textureless objects. Detecting dynamic objects through motion masks [5] or probabilities [7] results effective at improving generalization performance, *i.e.* when QRAFT is trained on C, C+T or C+T+S, yet cannot improve the results when testing on the same domain, *i.e.* when QRAFT has been trained on C+T+K. Finally, our complete pipeline – which combines EgoFlow and RICFlow through MaskRCNN – is the only configuration that can effectively guide QRAFT towards better accuracy, consistently in any training configuration.

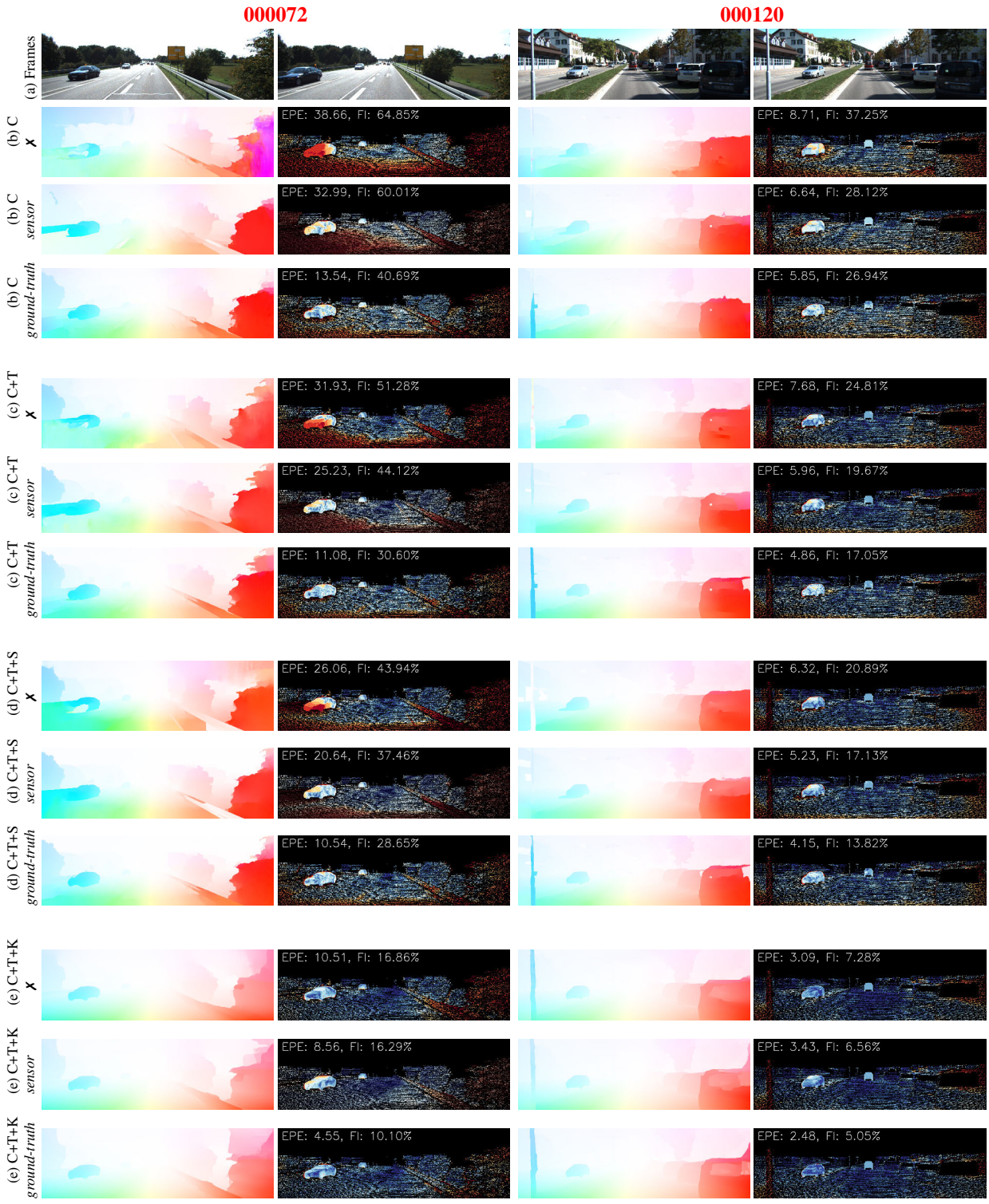

Figure 5. **Guided QRAFT performance with real and simulated guide.** From top: reference images (a) followed by flow (left) and error (right) maps by QRAFT without guide (✗), guided with real hints (*sensor*) or with hints sampled from ground-truth (*ground-truth*).

We also report further qualitative proof of how an inaccurate guide can catastrophically affect the predicted flow. Figure 6 shows two examples in which QRAFT, trained on C+T+S or C+T+K, is guided by two different strategies, respectively, LIDAR alone (top) or by our entire pipeline (bottom). In the former case, we can notice how estimating hints without

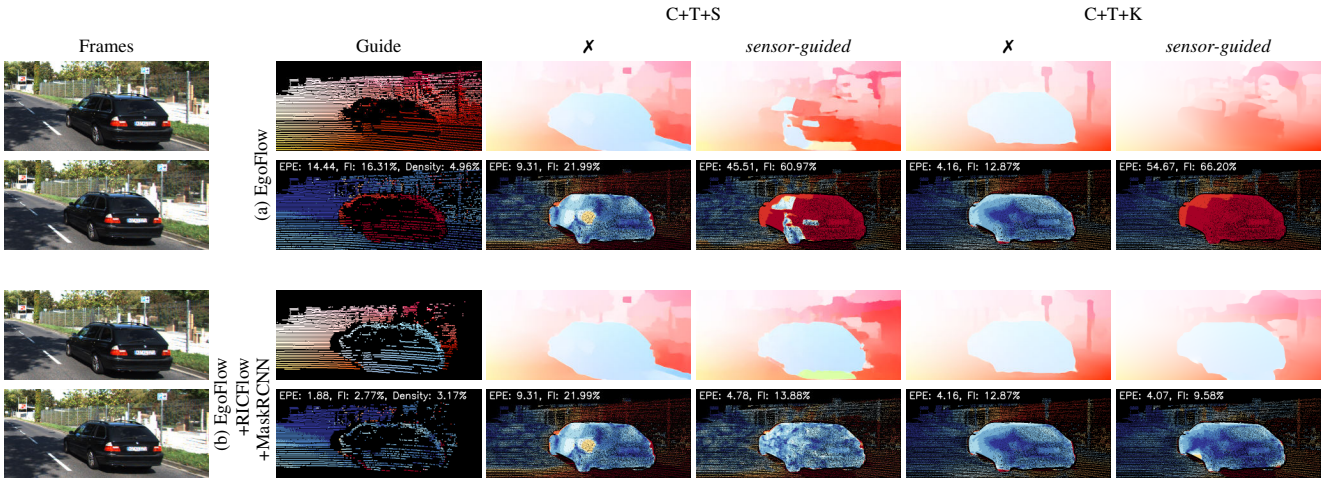

Figure 6. **Fooling QRAFT with inaccurate guidance.** We compare the results achieved by QRAFT without a guide (✗) or when guided (*sensor-guide*) by two kinds of hints, obtained respectively from LIDAR alone (a) or with our entire pipeline (b).

handling moving objects can fool QRAFT, guiding it to estimate a flow consistent with the background for the moving car. In contrast, providing meaningful hints for static regions and moving objects guides QRAFT reliably and boosts its accuracy.

4.3. Comparison between different flow hints

We show more qualitative (and quantitative) comparisons between flow hints obtained from LIDAR alone, RICFlow [8] or the complete pipeline sketched in the main paper. Figures 7, 8 and 9 collect examples of flow maps and corresponding error maps, with EPE, Fl and density over-imposed over the latter. We show frames, Velodyne depth maps and segmented objects for each sample in the first two rows, followed by the flow maps. We can notice how, in all the examples, the synergy of ego-motion flow estimated from LIDAR and dynamic objects flow estimated by RICFlow, enabled by MaskRCNN [2] segmentation, always yields the more accurate hints.

Indeed, the LIDAR alone cannot deal with moving objects, *e.g.* cars. Moreover, the latter sometimes leads to inaccurate pose estimation and, consequently, completely wrong ego-motion, as we can see for image 000077 in Fig. 8. RICFlow is, in general, effective on both background and foreground objects, but it is unreliable in the presence of moving shadows (for instance, as in frames 000078 and 00086, in Fig. 8) or in the case of low textured objects, such as the white wall on frame 000086. In order to combine the best of the two methods, we can reliably estimate the ego-motion flow from background pixels (*i.e.*, after segmenting objects with MaskRCNN and ignoring them) and complement the flow hints on foreground objects by picking RICFlow results.

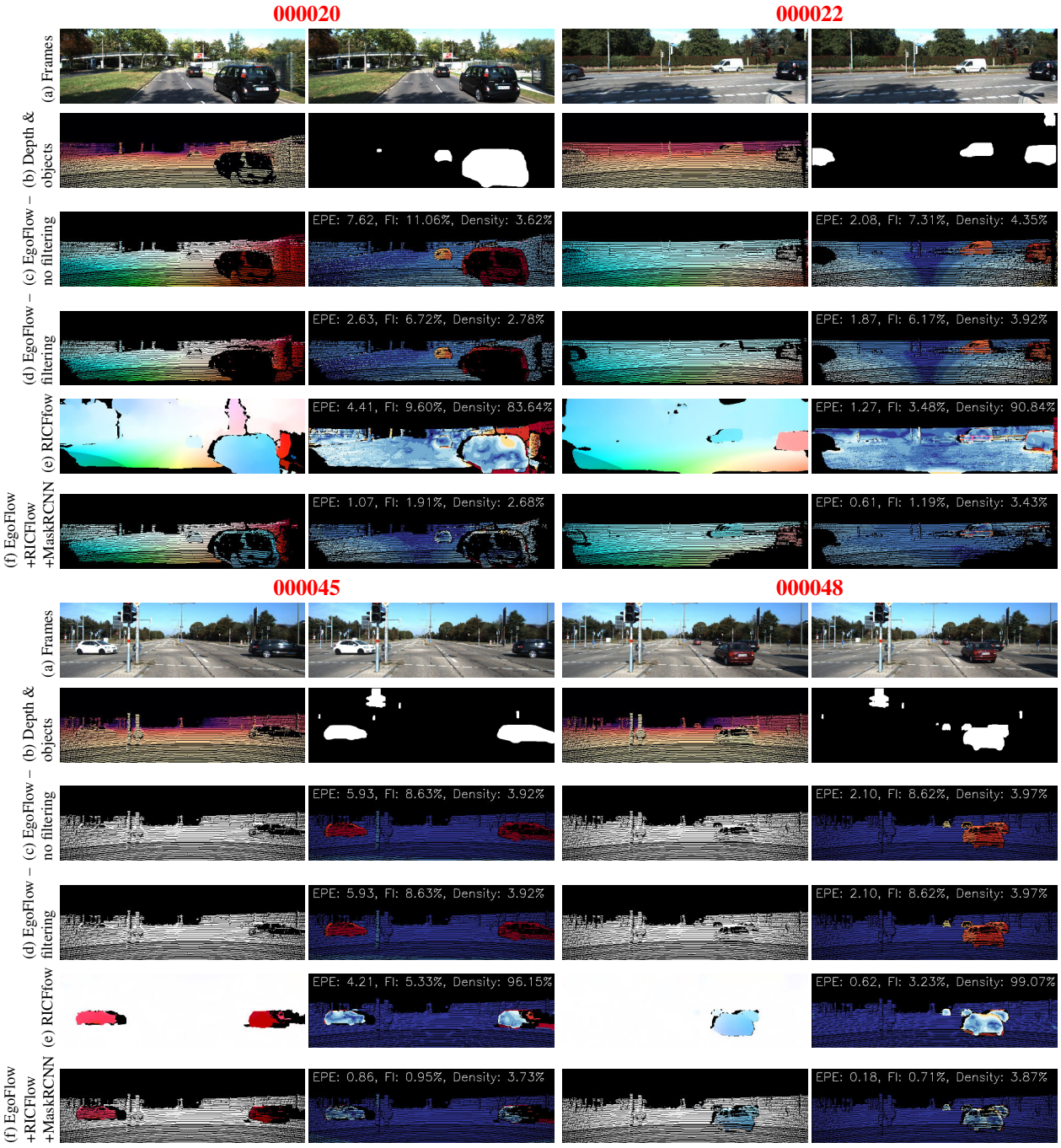

Figure 7. **Optical flow hints from a LIDAR (1/3).** Example of flow hints computed over six pairs of frames from KITTI 142 split. For each example, we show the reference images on top rows, followed by the Velodyne depth map and segmented objects (extracted by MaskRCNN). The remaining rows show flow guides (left) and error maps (right), densified for better visualization. The actual density is reported over each error map, with EPE and Fl computed on pixels with hints and ground-truth available.

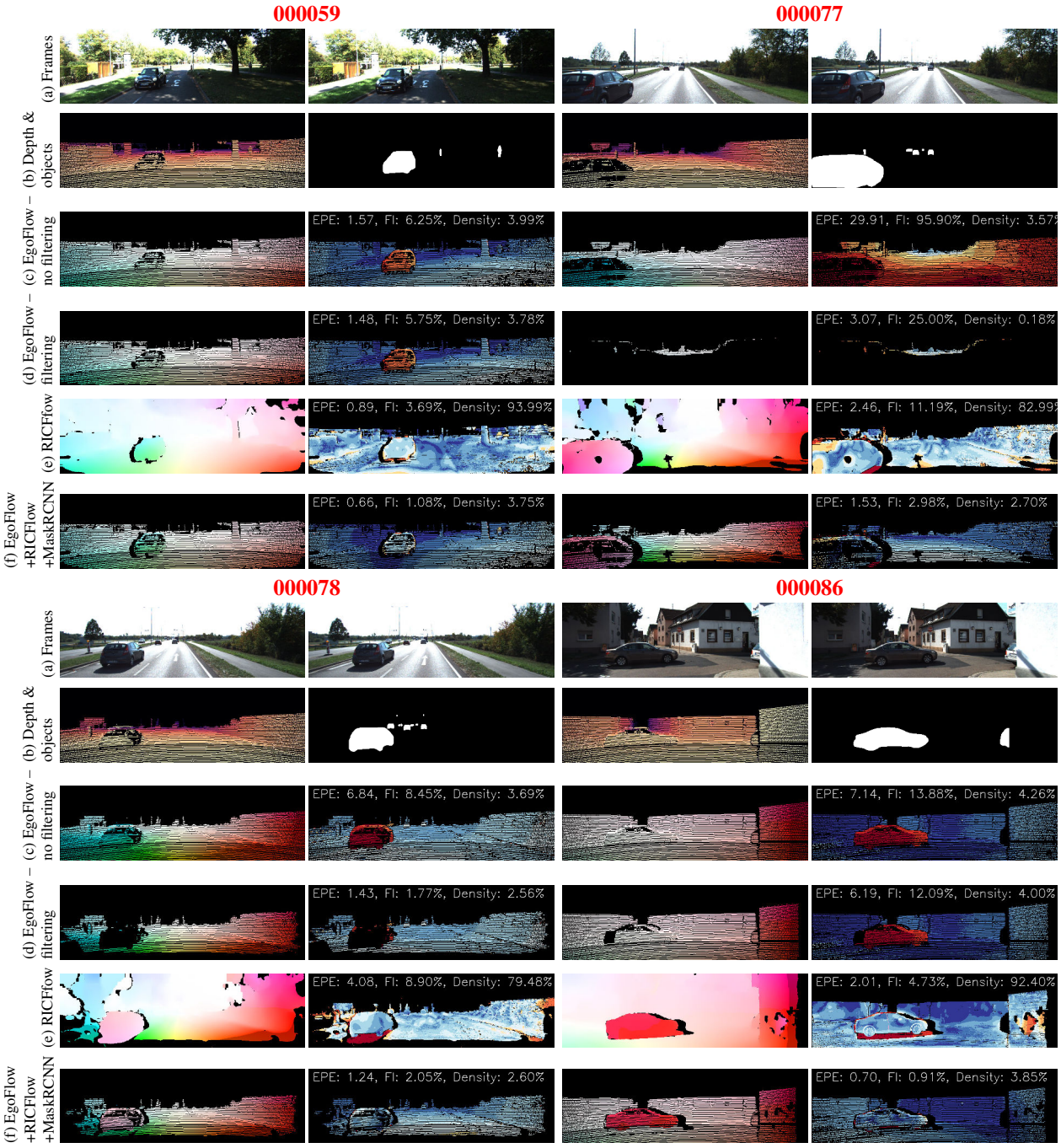

Figure 8. **Optical flow hints from a LIDAR (2/3).** Example of flow hints computed over six pairs of frames from KITTI 142 split. For each example, we show the reference images on top rows, followed by the Velodyne depth map and segmented objects (extracted by MaskRCNN). The remaining rows show flow guides (left) and error maps (right), densified for better visualization. The actual density is reported over each error map, with EPE and Fl computed on pixels with hints and ground-truth available.

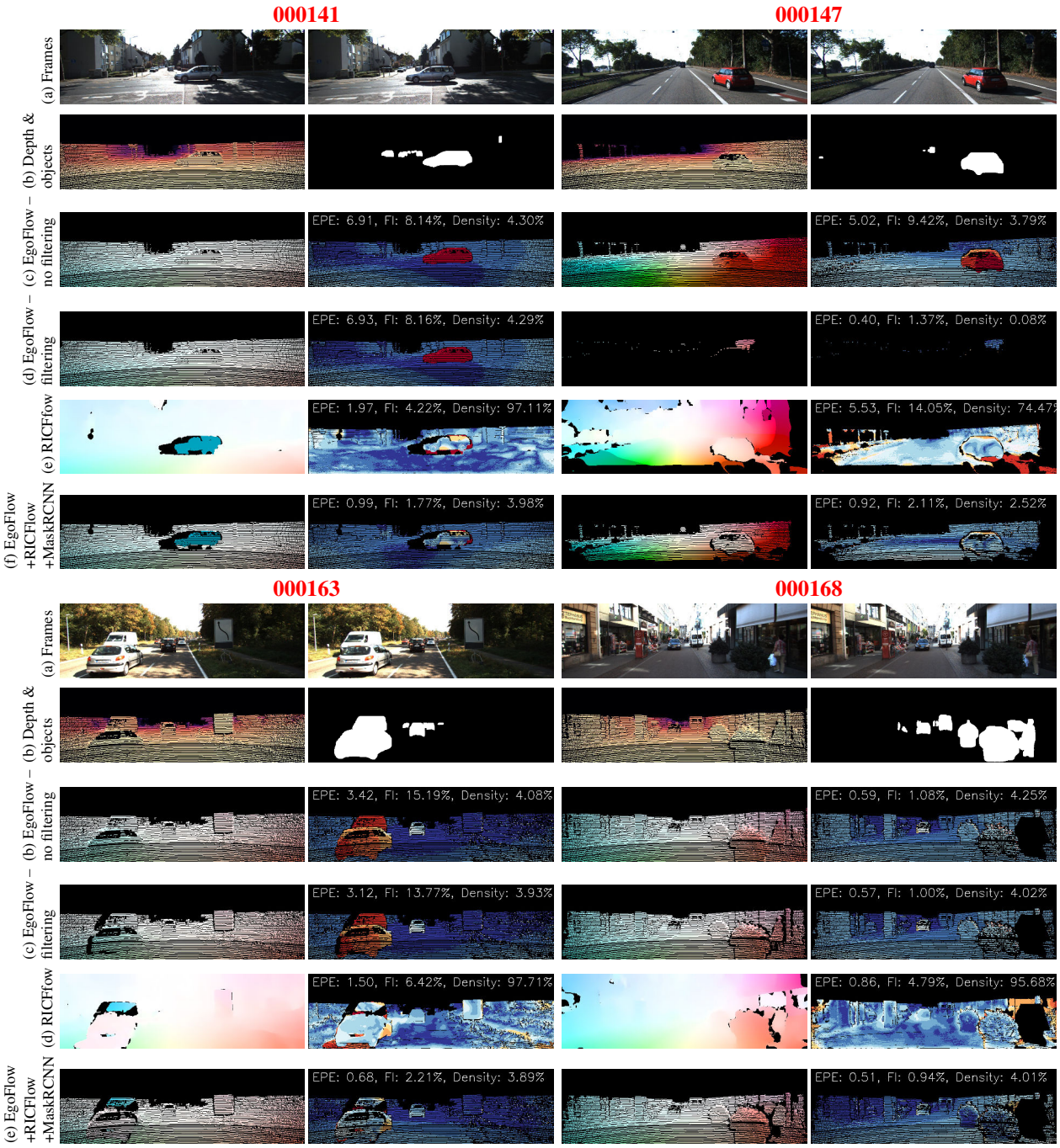

Figure 9. **Optical flow hints from a LIDAR (3/3).** Example of flow hints computed over six pairs of frames from KITTI 142 split. For each example, we show the reference images on top rows, followed by the Velodyne depth map and segmented objects (extracted by MaskRCNN). The remaining rows show flow guides (left) and error maps (right), densified for better visualization. The actual density is reported over each error map, with EPE and Fl computed on pixels with hints and ground-truth available.

5. Qualitative results

Finally, we report additional qualitative results of our sensor-guided optical flow framework implemented with a consumer device, *i.e*. Apple iPhone Xs equipped with a low-res depth sensor.

5.1. Apple iPhone Xs dataset

Purposely, we have collected a few image pairs shown in Fig. 10.

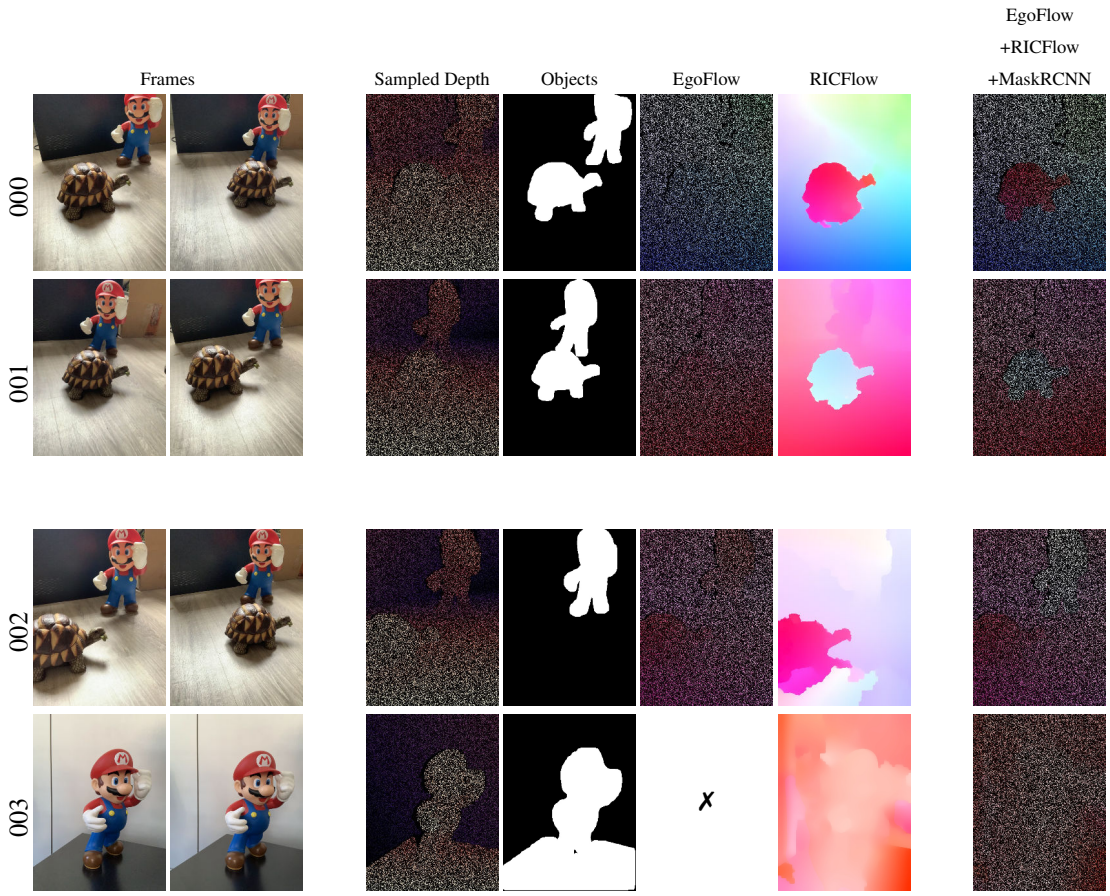

Figure 10. **Qualitative experiments – iPhone Xs.** We collect some examples with a handheld device equipped with a depth sensor. From left to right, we show the image pairs, the depth map acquired by the device and the segmentation mask produced by MaskRCNN, both for the first frame. Then, we show the computed ego-flow and RICFlow labels, followed by the final flow hints. We show on the top two examples over which our pipeline computes reliable flow hints, while on the bottom, we highlight two failure cases.

We show the two collected images for each sample, together with the corresponding depth map and MaskRCNN instance mask concerning the first frame, followed by EgoFlow and RICFlow labels, finally combined to obtain flow hints. We will run both RAFT and QRAFT by resizing images at 640×480 resolution, *i.e*. the native depth sensor resolution, for the sake of memory requirements necessary to run the flow networks. However, to account for the different resolution between depth maps and RGB native resolution (1504×1128) in the actual setting, we also sparsely downsample the depth maps by keeping about 20% of the valid measurements in order to simulate real conditions at best.

On top, we can notice how the synergy between EgoFlow and RICFlow estimates enabled by MaskRCNN effectively yields accurate flow hints on both background and moving objects (*i.e*., the tortoise), for samples 000 and 001.

Failure cases. At the bottom of Fig. 10, we report some examples for which our pipeline cannot reliably provide meaningful hints. In the first one (002), we notice how MaskRCNN fails to segment the tortoise in the sensed scene. This mistake leads the ego-motion estimation algorithm to infer the relative position between the two frames wrongly because of the high number of features matched on its shell. We can perceive it by visually comparing the sparse EgoFlow labels with those estimated by RICFlow: the flow direction of the former is the same, for the entire scene, of the tortoise in the latter

case. Thus, the combination of the two leads to inaccurate hints that might fool a network when guided accordingly, as we will see in the next section.

In the second example (003), the segmentation mask correctly removes Mario, but, unfortunately, most of the background is textureless. Consequently, the ego-motion algorithm cannot reliably find a sufficient number of matches to estimate the pose between the two frames. Thus, we are driven to pick RICFlow estimates alone, although it struggles on this scene because of the lack of texture.

5.2. RAFT vs sensor-guided QRAFT

We now show how RAFT, QRAFT and sensor-guided QRAFT perform on the image pairs shown before. Fig. 11 collects the results concerning frames 000 and 001, *i.e*. those for which we can reliably estimate hints. We report the predictions by RAFT using the model trained on a single GPU (†) as well as the authors' weights trained on $\times 2$ GPUs (††), by our QRAFT alone and by sensor-guided QRAFT.

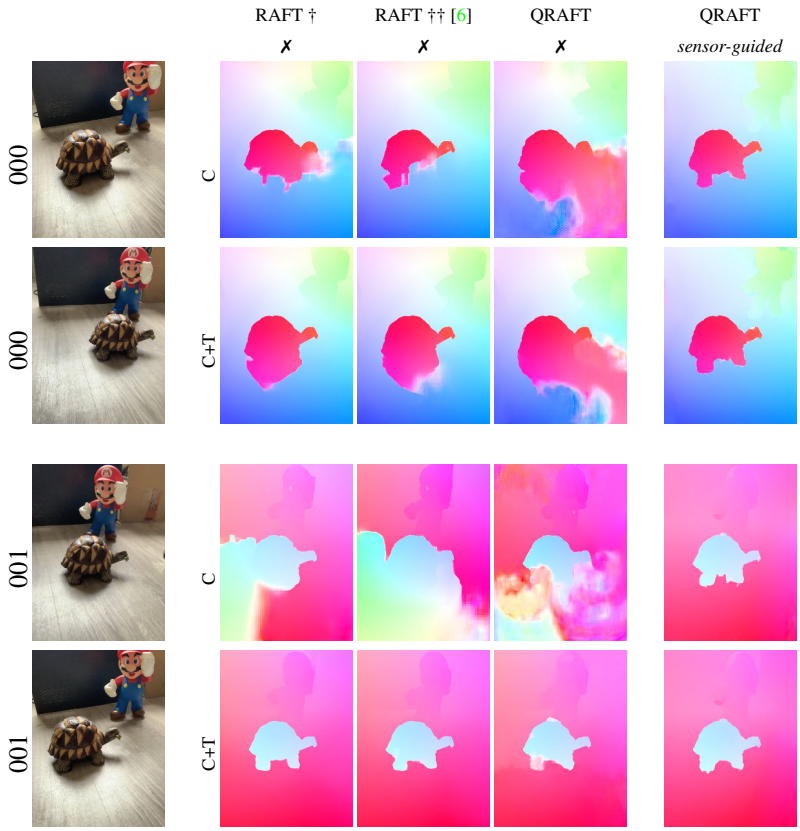

Figure 11. **Qualitative results – iPhone Xs.** On left: first (top) and second frame (bottom). On remaining columns, results by RAFT ($\times 3$ batch), RAFT with authors' weights [6] ($2\times$ GPUs), QRAFT and sensor-guided QRAFT, trained on C (top) or C+T (bottom).

On frame 000, we can notice how networks trained on C suffer: RAFT struggles at recovering the full tortoise's shape, while QRAFT wrongly estimates the motion of part of the floor. Conversely, sensor-guided QRAFT produces much better flow estimates. By looking at networks trained on C+T, RAFT this time is fooled by the tortoise's shadow, while QRAFT suffers from the same issues as before. Again, sensor-guided QRAFT provides the most robust estimates on the entire image.

On frame 001, we observe a similar trend for networks trained on C: RAFT and QRAFT primarily suffer from domain-shift issues, while sensor-guided QRAFT provides good predictions although trained on the same data. Considering networks trained on C+T, this time, both RAFT and QRAFT seem less affected by the domain shift, making the gain achieved by sensor-guided QRAFT less evident.

Failure cases. To conclude, we show the performance of RAFT, QRAFT and sensor-guided QRAFT on samples on which our flow hints pipeline fails. Fig. 12 collects the results concerning frames 002 and 003. We report the same predictions shown in Fig. 11, with the addition of a final column showing the results achieved by sensor-guided QRAFT after having

manually corrected the flow guide – the manual intervention differs according to the kinds of failure that occurred. Although this latter intervention cannot occur in real applications, it helps us figure out the full potential of our framework in the absence of failures.

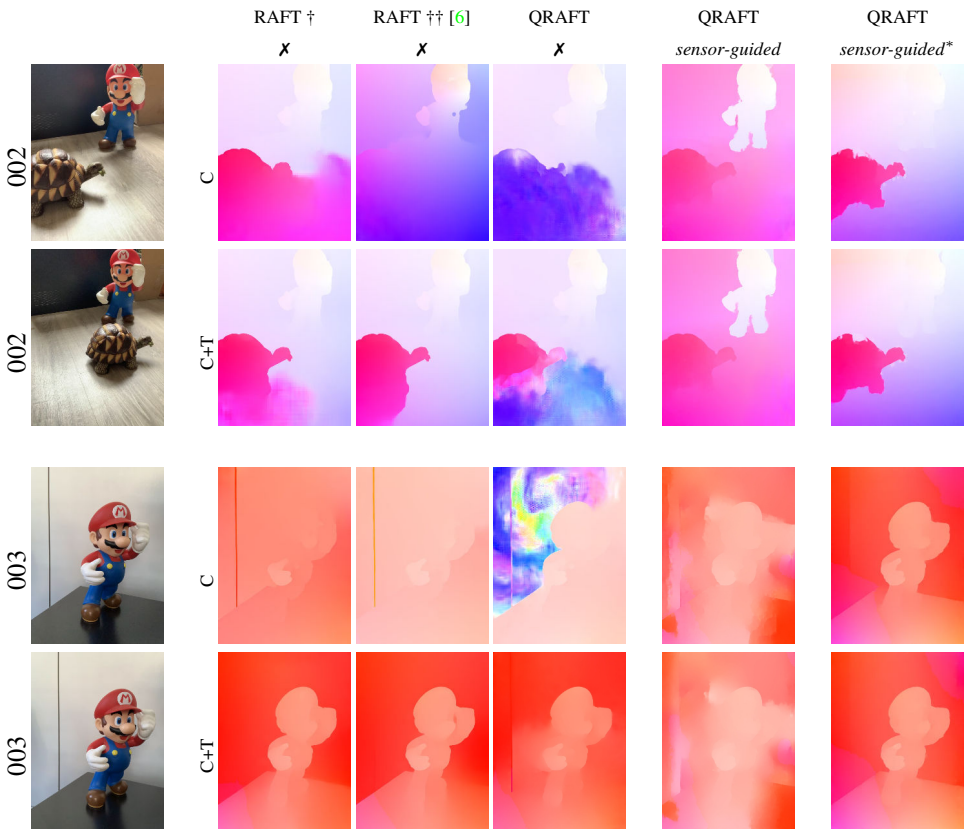

Figure 12. **Qualitative results – iPhone Xs.** On left: first (top) and second frame (bottom). On remaining columns, results by RAFT ($\times 3$ batch), RAFT with authors' weights [6] ($2\times$ GPUs), QRAFT and sensor-guided QRAFT, trained on C (top) or C+T (bottom). * denotes manual intervention on flow hints computation.

On frame 002, networks trained on C suffer again, as observed on frames 000 and 001. Sensor-guided QRAFT apparently produces flow estimates that are much more defined, but this is not the case because the flow hints used to guide it are inaccurate (see Fig. 10) because of the missing segmentation of the moving tortoise. By intervening on hints computation, *i.e*. through manual segmentation of the tortoise, we can obtain a reliable guide that leads to the predictions shown in the rightmost column. By looking at networks trained on C+T, the tortoise's shadow fools both RAFT and QRAFT. Again, sensor-guided QRAFT produces smooth flow maps but the inaccurate guide fools it. Correcting the flow hints allows guiding the network towards correct estimates.

On frame 003, we notice a trend similar to what was observed on frame 001, with networks trained on C primarily suffering from domain-shift issues. Since the ego-motion estimation algorithm failed on this sample, the flow hints are computed from RICFlow alone and let sensor-guided QRAFT predict an output recalling RICFlow estimates (see Fig. 10). In this case, the failure was caused by the lack of texture in the background. Purposely, this example has been acquired in a static environment in order to be able to relax the constraint on moving objects and ignore the segmentation mask, and thus exploit the texture on Mario to properly estimate ego-motion. By guiding QRAFT with flow hints obtained accordingly, we obtain the much more accurate prediction shown on the rightmost column. Concerning networks trained on C+T, this time, both RAFT and QRAFT seem less affected by the domain shift. This fact highlights how a flow hints algorithm's failure can make the sensor-guided optical flow framework dramatically worse than classical flow networks, while manually intervening on the guide shows its full potential again.

Take-home message. The thorough study carried out in the main paper and this supplementary document shows the full potential of the guided optical flow framework proposed and the effectiveness of an actual implementation based on depth sensors. Nonetheless, although the failure cases mentioned above have been created ad-hoc in a controlled environment and

do not occur on the KITTI 142 split, they could occur during deployment in the wild. Therefore, future improvements to the flow hints pipeline – and, possibly, the advent of sensing technologies mature to measure flow – have the potential to overcome the current limitations and leverage the guided optical flow framework proposed in this paper at its full potential.

References

[1] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In A. Fitzgibbon et al. (Eds.), editor, *European Conf. on Computer Vision (ECCV)*, Part IV, LNCS 7577, pages 611–625. Springer-Verlag, Oct. 2012. 1

[2] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 5, 7

[3] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1

[4] Matteo Poggi, Davide Pallotti, Fabio Tosi, and Stefano Mattoccia. Guided stereo matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 4

[5] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J Black. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12240–12249, 2019. 5

[6] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 12, 13

[7] Fabio Tosi, Filippo Aleotti, Pierluigi Zama Ramirez, Matteo Poggi, Samuele Salti, Luigi Di Stefano, and Stefano Mattoccia. Distilled semantics for comprehensive scene understanding from videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 5

[8] Jia Xu, René Ranftl, and Vladlen Koltun. Accurate optical flow via direct cost volume processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1289–1297, 2017. 7