Sensor-Guided Optical Flow

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

Abstract

This paper proposes a framework to guide an optical flow network with external cues to achieve superior accuracy either on known or unseen domains. Given the availability of sparse yet accurate optical flow hints from an external source, these are injected to modulate the correlation scores computed by a state-of-the-art optical flow network and guide it towards more accurate predictions. Although no real sensor can provide sparse flow hints, we show how these can be obtained by combining depth measurements from active sensors with geometry and hand-crafted optical flow algorithms, leading to accurate enough hints for our purpose. Experimental results with a state-of-the-art flow network on standard benchmarks support the effectiveness of our framework, both in simulated and real conditions.

1. Introduction

The task of optical flow computation [21] aims at estimating the motion of pixels in a video sequence (e.g., in the most common settings, from two consecutive frames in time). As a result, several higher-level tasks can be faced from it, such as action recognition, tracking and more. Although its long history, optical flow remains far from being solved due to many challenges; the lack of texture, occlusions or the blurring effect introduced by high-speed moving objects make the problem particularly hard.

Indeed, the adoption of deep learning for dense optical flow estimation has represented a turning point during the years. The possibility of learning more robust pixels similarities [2, 73] allowed, at first, to soften the issues above. Then the research trend in the field rapidly converged towards direct inference of the optical flow field in an end-to-end manner [15, 29, 62, 63, 26, 27, 25, 64], achieving both unrivaled accuracy and run time in comparison to previous approaches. The availability of a large amount of training data annotated with ground-truth flow labels, in most cases obtained for free on synthetic images [10, 15, 29], ignited this spread. Common to most end-to-end networks is the use of a correlation layer [15], explicitly computing similarity scores between pixels in the two images in order to find matches, and thus flow.

This trend, however, introduced new challenges inherently connected to the learning process. Specifically, the use of synthetic images is rarely sufficient to achieve top performance on real data. As witnessed by many works in the field [15, 29, 62, 63, 26, 27, 25, 64], a network trained on synthetic images already excels on benchmarks such as Sintel [10], yet struggles at generalizing to real benchmarks such as KITTI [17, 47]. This phenomenon is known as domain-shift and is usually addressed by fine-tuning on few real images with available ground-truth. Nevertheless, achieving generalization without fine-tuning still represents a desirable property when designing a neural network. The main cause triggering the domain-shift issue is the very different appearance of synthetic versus real images, with the former unable to faithfully model noise, lightning conditions...
and other effects usually found in the latter, as extensively supported by the literature \cite{50, 53, 65, 66, 78, 52, 11}. However, it has been shown that a deep neural network can be guided through external hints to reduce the domain-shift effect significantly. In particular, in the case of guided stereo matching \cite{52}, a neural network can be conditioned during cost-volume computation with sparse depth measurements, obtained, for instance, employing a LIDAR sensor. This strategy dramatically increases generalization across domains, as well as specialization obtained after fine-tuning.

Inspired by these findings, in this paper we formulate the guided optical flow framework. Supposing the availability of a sparse yet accurate set of optical flow values, we use them to modulate the correlation scores usually computed by state-of-the-art networks to guide them towards more accurate results. To this aim, we first extend the guided stereo formulation to take into account 2D cost surfaces. Then, we empirically study how the effect of the sparse points is affected by the resolution at which the correlation scores are computed and, consequently, revise the state-of-the-art flow network, RAFT \cite{64}, to make it better leverage such a guide. The effectiveness of this approach is evaluated, at first, from a theoretical point of view by sampling a low amount of ground-truth flow points (about 3%) – perturbed with increasing intensity of noise – to guide the network, and then using flow hints obtained by a real setup. However, in contrast to stereo/depth estimation \cite{52}, sensors capable of measuring optical flow do not exist at all. Consequently, we show how to obtain such a sparse guide out of an active depth sensor combined with a hand-crafted flow method and an instance-segmentation network \cite{19}. It is worth noting that the setup needed by our proposal is already regularly deployed in many practical applications, such as autonomous driving, and nowadays even available in most consumer devices like smartphones and tablets equipped with cameras and active depth sensors.

Figure 1 shows the potential of our method in a challenging environment (a) where the same, state-of-the-art flow network \cite{64} has been run after being trained on synthetic images only. In its original implementation (b), the network miserably fails. Instead, the same network re-trained and guided by our framework (c) with a few hints (e.g., about 3% of the total pixels, sampled from ground-truth and perturbed with random noise for this example) is dramatically improved. Experiments carried out on synthetic (FlyingChairs, FlyingThings3D, Sintel) and real (Middlebury, KITTI 2012 and 2015) datasets support our main claims:

- We show, for the first time, that an optical flow network can be conditioned, or guided, by using external cues. To this aim, we pick RAFT \cite{64}, currently the state-of-the-art in dense optical flow estimation, and revise it to benefit from the guide at its best.

- Supposing to have the availability of less than 3% sparse flow hints, guided optical flow allows to largely reduce the domain-shift effect between synthetic and real images, as well as to further improve accuracy on the same domain.

- Although virtually no sensor is capable of providing such accurate flow hints \cite{49}, we prove that a LIDAR sensor, combined with a hand-crafted flow algorithm, can provide a meaningful guide.

2. Related Work

We briefly review the literature relevant to our work.

**Hand-crafted optical flow algorithms.** Since the seminal work by Horn and Schunck \cite{21}, for years optical flow has been cast into an energy minimization problem \cite{8, 7, 60, 59}, for instance by means of variational frameworks \cite{6, 77}. These approaches involve a data term coupled with regularization terms, and improvements to the former \cite{7, 71} or the latter \cite{54} have represented the primary strategy to increase optical flow accuracy for years \cite{59}. While these approaches perform well in presence of small displacements, they often struggle with larger flows because of the failure of the initialization process performed by the energy minimization framework. Some approaches overcome this problem by interpolating a sparse set of matches \cite{36, 38, 23, 22}, but they are however affected by well-known problems occurring when dealing with pixels matching, such as motion blur, violation of the brightness-consistency and so on. More recent strategies consider optical flow as a discrete optimization problem, despite managing the sizeable 2D search space required to determine corresponding pixels between images \cite{48, 12, 73} is challenging. First attempts to improve optical flow with deep networks mainly consisted of learning more robust data terms by training CNNs to match patches across images \cite{71, 2, 73}, before converging to end-to-end models \cite{15}.

**End-to-end Optical Flow.** The switch towards fully learnable models for estimating optical flow represented a major turning point in the field. FlowNet \cite{15} is the first end-to-end deep network proposed for this purpose. In parallel, to satisfy the massive amount of training data required in this new setting, synthetic datasets with dense optical flow ground-truth labels were made available \cite{15, 45}. Starting with FlowNet, a number of architectures further improved accuracy on popular synthetic \cite{10, 45} and real \cite{47, 17} benchmarks, designing 2D architectures \cite{29, 30, 79, 72, 64}, refinement schemes \cite{28, 70} or, more recently, 4D networks as well \cite{74, 68}. Among them, RAFT \cite{64} currently represents the state-of-the-art. Concurrently, the use of deep networks also allowed to investigate on efficiency, leading to many compact models \cite{55, 62, 63, 26, 27, 25, 75, 4} capable of running in real-time at the cost of
Figure 2. **Modulation of 2D correlation scores.** Given a pixel for which a flow hint of values (2,2) is available, we show (a) raw correlation scores computed in a search window of radius 4, (b) the modulating function centered at hinted coordinates and (c) modulated scores.

3. Proposed framework

In this section, we describe our framework for guided optical flow estimation. First, we recall the guided stereo matching formulation [52] as the background of our proposal, then we extend it to the case of optical flow.

3.1. Background: Guided Stereo Matching

Given the availability of sparse yet accurate depth measurements coming, for instance, from a LIDAR sensor, a deep stereo network can be **guided** to predict more accurate disparity maps by leveraging such measurements. This outcome is achieved by acting on a data structure, abstracted as a **cost-volume**, where state-of-the-art networks store the probability of a pixel on the left image to match with the one on the right shifted by an offset $-d$. Specifically, the depth hint associated with a generic pixel $p$ is converted into a disparity $d_p'$ according to known camera parameters. Then, the cost-volume entry (i.e., cost-curve $C_p$) for pixel $p$ is modulated using a Gaussian function centered on $d_p'$, so that the single score of the cost-curve corresponding to the disparity $d = d_p'$ is multiplied by the peak of the modulating function. Concerning the remaining scores, the farther they are from $d_p'$ the more are dampened. This strategy yields a new cost-curve, $C'_p$. The modulation takes place only for pixels with a valid depth hint, while for the others, the original cost-curve $C_p$ is kept. Thus, by defining a per-pixel binary mask $v$ in which $v_p = 1$ if a depth measurement is available for pixel $p$, $v_p = 0$ otherwise, the modulation can be expressed as:

$$C'_p(d) = \left(1 - v_p + v_p \cdot k \cdot e^{-\frac{d-d_p'^2}{2c^2}}\right) \cdot C_p(d)$$

with $k$ and $c$ being respectively the height and width of the Gaussian. For stereo, $C_p$ is often defined by means of a correlation layer [45] or features concatenation / difference [33, 34]. A similar practise is followed for optical flow, although the search domain is 2D rather than 1D.

3.2. Guided Optical Flow

Similar to what is done by stereo networks, a common practise followed when designing an optical flow network is the explicit computation of correlation scores between features to encode the likelihood of matches. In most cases by means of 2D correlation layers [15] and, more recently, by concatenating features [74, 68]. This leads to a 4D cost-volume structure, often reorganized to be processed by 2D convolutions for the sake of efficiency [15, 29, 64]. In it, each entry for a generic pixel $p$ represents a 2D distribution of matching scores, corresponding to the 2D search range over which pixels are compared, as shown in Fig. 2 (a).

Accordingly, by assuming a sparse set of flow hints, consisting of 2D vectors $(x_p^*, y_p^*)$ for any pixel $p$, the correlation volume entry $C_p$ (i.e., a correlation-surface) is modulated by means of a bivariate Gaussian function centered on $(x_p^*, y_p^*)$, for which an example is shown in Fig. 2 (b) having $(x_p^*, y_p^*) = (2, 2)$. As a consequence, the single score of the correlation-surface corresponding to flow $(x, y) = (x_p^*, y_p^*)$
results peaked, while the remaining scores are dampened according to their distance from \((x'_p, y'_p)\). Again, considering a binary mask \(v\) encoding pixels with a valid hint, the guided optical flow modulation can be expressed as:

\[
C'_p(x, y) = \left(1 - v_p + v_p \cdot k \cdot e^{-\frac{(x-x'_p)^2+(y-y'_p)^2}{2\sigma^2}}\right) \cdot C_p(x, y) \quad (2)
\]

The resulting correlation-surface is shown in Fig. 2 (c). Although any differentiable function would be amenable for modulation, the choice of a Gaussian allows for peaking correlation scores corresponding to the hinted values together with neighboring scores, thus taking into account slight deviations of the hint from the actual flow value.

4. Implementing Sensor-Guided Optical Flow

As shown before, in theory, we can seamlessly extend the original stereo formulation to the optical flow problem. However, some major issues arise during the implementation. In particular, 1) existing optical flow architectures are not suited for guided optical flow and 2) obtaining flow hints from a sensor is not as natural as in the case of depth estimation, since do not exist equivalent devices capable of measuring the optical flow. In the reminder, we will describe how to address both problems.

4.1. Network choice and modifications

To effectively guide the neural network to predict more accurate flow vectors, consistently with stereo formulation [52], we act on the similarity scores computed by specific layers of the flow networks. The literature is rich of architectures leveraging 2D correlation layers [15, 29, 62, 26, 64] or, more recently, features concatenation in 4D volumes [74, 68]. Currently, RAFT [64] represents the state-of-the-art in the field and thus the preferred choice to be enhanced by our guided flow formulation, in particular, because of 1) its capacity of computing matching scores between all pairs of pixels in the two images, 2) its much faster convergence and 3) its superior generalization capability and accuracy.

However, RAFT and all the networks mentioned before usually compute correlations / concatenate features at low resolution, \(i.e.\), \(\frac{1}{8}\) or lower. On the one hand, this does not allow for a fine modulation since a single flow hint would modulate a distribution of coarse 2D correlation scores, making guided flow poorly effective or even harmful for the network, as we will see in our experiments. On the other hand, the guided stereo framework [52] proved to be effective when correlation / concatenation is performed on features at \(\frac{1}{4}\) resolution. Accordingly, we revise RAFT to make it suited for guided flow as follows: 1) the encoder is modified to extract features at quarter resolution, by changing the stride factor from 2 to 1 in the sixth convolutional layer and reducing the amount of extracted features from 128 to 96 to reduce complexity and memory requirements; 2) to perform convex upsampling of the predicted flow, a \(\frac{4}{3} \times \frac{4}{3} \times (4 \times 4 \times 9)\) mask is predicted instead of \(\frac{8}{3} \times \frac{8}{3} \times (8 \times 8 \times 9)\). We dub this Quarter resolution RAFT variant QRAFT. Experimentally, we will show that it is much better suited to leverage guided flow, significantly improving accuracy when fed with hints.

Although similar modifications are theoretically applicable to most state-of-the-art optical flow networks, they result practically unfeasible on 4D networks [74, 68] because of 1) the much higher complexity/memory requirements of 4D convolutions and 2) the resolution at which the volumes are built, usually \(\frac{1}{16}\) or lower, that would require a much higher overhead to reach the desired quarter resolution.

4.2. Accurate flow hints from active depth sensors

In this section, we describe a possible implementation of a real system capable of providing sparse flow guidance. Although a sensor measuring the optical flow does not exist, we can implement a virtual one by combining existing sensors and known geometry properties. First, we point out that pixel flow between two images \(I_0, I_1\) is the consequence of two main components: 1) camera ego-motion and 2) independently moving objects in the scene.

**Ego-motion flow.** Concerning the former, it is straightforward to compute it by leveraging geometry if camera intrinsics \(K\), depth \(D_0\) for pixels \(p_0\) in \(I_0\) and relative pose \(T_{0\rightarrow1}\) are known. Accordingly, corresponding coordinates \(p_1\) in \(I_1\) can be obtained by projecting \(p_0\) in 3D space using \(K^{-1}\) and \(D_0\), applying roto-translation \(T_{0\rightarrow1}\) and back-projecting to \(I_1\) image plane using \(K\)

\[
p_1 \sim KT_{0\rightarrow1}D_0(p_0)K^{-1}p_0 \quad (3)
\]

While \(K\) is known, depth \(D_0\) can be obtained by means of sensors, since a variety of devices for depth sensing exist, a
LIDAR for instance. Finally, the relative pose $T_{0 \to 1}$ can be obtained by solving the Perspective-n-Point (PnP) problem [37] between frames $I_0$ and $I_1$, by knowing corresponding LIDAR depths $D_0$ and $D_1$ and using matched feature correspondences extracted from $I_0$ and $I_1$, filtered by means of RANSAC [16] as in [44]. Finally, flow $f_{0 \to 1}^f$ – or EgoFlow – can be obtained by subtracting $p_0$ coordinates from $p_1$.

Although noisy, LIDAR measurements are accurate enough to allow for computing meaningful flow guide when dealing with static scenes, as shown in Fig. 3 (b). Moreover, we can further remove noisy flow estimates by deploying a forward-backward consistency mask $c^e_{0 \to 1}$. This is obtained by computing the ego-motion backward flow $f^e_{1 \to 0}$, by backward warping $f^e_{1 \to 0}$ according to $f^e_{0 \to 1}$ and then by comparing warped flow $f^e_{1 \to 0}$ with $f^e_{0 \to 1}$ itself, resulting consistent if the two flows for a same pixel $p_0$ are opposite. Thus, we consider valid pixels those having an Euclidean distance between $f^e_{0 \to 1}$ and $-f^e_{1 \to 0}$ lower than a threshold (e.g., 3):

$$c^e_{0 \to 1}(p_0) = \begin{cases} 1 & \text{if } \|f^e_{0 \to 1}(p_0) + f^e_{1 \to 0}(p_0)\|_2 \leq 3 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

However, since LIDAR points are sparse, they would rarely match after warping. Thus, we apply a simple completion filter based on classical image processing techniques [35] and compute $c^e_{0 \to 1}$ by replacing depth maps in Eq. 3 with their densified counterparts. This allows to discard noisy measurements and increase the quality of the flow guide at the expense of density, as shown in Fig. 3 (c). Nonetheless, this strategy alone cannot deal with dynamic objects.

**Independently moving objects flow.** The methodology introduced so far is effective when framing a static scene, but it results insufficient when moving objects appear. Indeed, Fig. 4 shows an example in which a car is moving in the scene (a), whose flow estimated from LIDAR alone is largely incorrect, as shown in (b). Forward-backward consistency allows to filter out the moving car, but only partially as shown in (c). Moreover, this would not allow for recovering flow hints for dynamic objects, thus providing no cues to the neural network we wish to guide. To recover these missing cues, we leverage hand-crafted optical flow algorithms that indiscriminately process static and dynamic parts of the scene without the need for training (thus not suffering from domain gap issues). Purposely, we select RICFlow [22] as hand-crafted algorithm because of its good trade-off between accuracy and fast inference time (a few seconds on modern CPUs), compatible with state-of-the-art networks runtime. By running RICFlow, we obtain $f_{0 \to 1}^{RIC}$ and eventually perform the forward-backward consistency check detailed in Eq. 4. The resulting flow, shown in Fig. 4 (d), is aware of both static and dynamic elements in the scene, although it suffers of the well-known limitations of hand-crafted algorithms, as visible for instance under the car. However, as shown by error maps in Fig. 4 (b) and (d), the two strategies complement each other, with LIDAR flow performing better on static regions and RICFlow on dynamic objects. Thus, we combine the two sources to obtain a complete and accurate flow guide on both cases, by distinguishing background regions from moving objects and picking EgoFlow or RICFlow accordingly.

A strategy to achieve this task consists of explicitly segmenting the scene into background regions and foreground objects (capable of independent motion), e.g. cars or pedestrians, for instance, employing an off-the-shelf instance segmentation network such as MaskRCNN [19]. By considering the segmentation mask $s$ produced by this latter, encoding objects with different IDs, we define $f_{0 \to 1}(p_0)$ as:

$$f_{0 \to 1}(p_0) = \begin{cases} f_{0 \to 1}^{RIC}(p_0) & \text{if } s(p_0) = 0 \\ f_{0 \to 1}^e(p_0) & \text{otherwise} \end{cases} \tag{5}$$

in which $s$ is 0 for pixels not belonging to foreground objects. This results in a guide that is meaningful on both static regions and dynamic objects, as shown in Fig. 4 (e).

5. Experimental results

In this section, we collect the outcome of our experiments. We first define the datasets involved and the implementation/training details. Then, we show: 1) a comparison between RAFT and QRAFT, 2) experiments guiding the two with sparse hints (~3%) sampled from ground-truth or 3) with the flow guide introduced in Sec. 4.2.
5.1. Datasets.

FlyingChairs (C) and FlyingThings3D (T). FlyingChairs [15] is a popular synthetic dataset used to train optical flow models. It contains 22232 images of chairs moving according to 2D displacement vectors over random backgrounds sampled from Flickr. The FlyingThings3D dataset [29] is a collection of 3D synthetic scenes belonging to the SceneFlow dataset [45] and contains a training split made of 19635 images. Differently from C, objects move in the scene with complex 3D motions. Traditionally, both are used to pre-train flow networks: we will consider networks trained on the former only (C) or both in sequence (C+T).

Sintel (S). Sintel [10] is a synthetic dataset with ground-truth optical flow maps. We use its training split, counting 1041 images for both Clean and Final passes. In particular, we divide it into a fine-tuning split (containing sequences alley_1, alley_2, ambush_2, ambush_4, ambush_5, ambush_6, ambush_7, bamboo_1, bamboo_2, bandeage_1, bandeage_2, cave_2, cave_4) and an evaluation split (containing the remaining ones). We also evaluate networks fine-tuned on the aforementioned fine-tuning split (C+T+S).

Middlebury Flow. The Middlebury Flow benchmark [3] is a collection of 4 synthetic and 4 real images with ground-truth optical flow maps. We use it for testing only.

KITTI 2012 and 142 split. The KITTI dataset is a popular dataset for autonomous driving with sparse ground-truth values for both depth and optical flow tasks. Two versions exist, KITTI 2012 [17] counting 194 images framing static scenes and KITTI 2015 [47] made of 200 images framing moving objects, in both cases gathered by a car in motion. We use the former for evaluation only, while a split of 142 images from the latter overlaps with the KITTI raw dataset [17] for which raw Velodyne scans are provided, thus allowing us to validate guided flow in a real setting, namely sensor-guided optical flow. The remaining 58 frames (K) are used in our experiments to fine-tune flow networks previously trained on synthetic data (C+T+S).

5.2. Implementation details and training protocols.

Our framework has been implemented starting from RAFT official source code. We follow the training schedules (optimizer, learning rate, iterations and weight decay) suggested in [64] to train both RAFT and QRAFT in a fair setting, training in order on C and T for 100K steps each, then fine-tuning on S or K for 50K steps. Given the higher memory requirements of QRAFT, we slightly change the crop sizes to 320 × 496, 320 × 640, 400 × 720 and 288 × 960 respectively for C, T, S and K, using image batches of 2, 1, 1 and 1, in order to fit into a single Titan Xp GPU. When turning on guided flow, we set \( k = 10 \) and \( c = 1 \) following [52]. The modulation acts on the correlation map computed between all pixels by downsampling the flow hints to the proper resolution with nearest neighbor interpolation.

During training, flow guide is obtained by randomly sampling 1% pixels from the ground-truth and applying random uniform noise \( e \in [-1,1] \), in order to make the network robust to inaccurate flow hints at test time. An ablation study on these hyper-parameters is reported in the supplementary material. Our demo code is available at https://github.com/mattpoggi/sensor-guided-flow.

During training, flow guide is obtained by randomly sampling 1% pixels from the ground-truth and applying random uniform noise \( e \in [-1,1] \), in order to make the network robust to inaccurate flow hints at test time. An ablation study on these hyper-parameters is reported in the supplementary material. Our demo code is available at https://github.com/mattpoggi/sensor-guided-flow.

5.3. Comparison between RAFT and QRAFT

We first validate the performance of QRAFT with respect to the original RAFT architecture [64], i.e. without using the guide. Tab. 1 collects the outcome of this comparison, carried out on Sintel, Middlebury and KITTI datasets with various training configurations. On top, we show the results achieved by training both RAFT and QRAFT with the same batch size (i.e., 2 on C, 1 on T, S and K). We can notice how QRAFT outperforms RAFT when trained in the same setting thanks to the higher resolution at which it operates, with very few exceptions – (a) vs (b) and (g) vs (h) on KITTI 2012 EPE. However, QRAFT adds a high computational overhead compared to RAFT. Indeed, this latter can be trained with \( \times 3 \) larger batch size on the same hardware (marked with \( \dagger \)). In this setting, RAFT results often better than QRAFT, except on Middlebury on most cases – (a) vs (b), (c) vs (d) and (e) vs (f) – and on KITTI 142 after fine-tuning – (g) vs (h). We report, for completeness, the accuracy of models provided by the authors [64], although trained with \( \times 2 \) GPUs and thus not directly comparable (marked with \( \dagger \)). Concerning efficiency, RAFT and QRAFT run respectively at 3.10 and 1.10 FPS on KITTI images (0.32 vs 0.91 sec per inference) on a Titan Xp GPU.

5.4. Guided Optical Flow – simulated guide

To evaluate the guided flow framework on standard datasets, we simulate the availability of sparse flow hints...
Table 2. Evaluation – Guided Optical Flow. Evaluation on Sintel sequences selected for validation (Clean and Final), Middlebury, KITTI 2012 and KITTI 142 split. Results without (X) or with (guided) flow guide. On the bottom, (i) statistics concerning the sampled guide.

Table 3. Flow guide accuracy. Evaluation on KITTI 142 split for flow hints generated by using different cues.

(~ 3%) at test time by randomly sampling from the ground-truth flow labels. Since the availability of a perfect guide as the one obtained by sampling from ground-truth is unrealistic, we perturb both $(x, y)$ in the sampled guide with additive random noise $\varepsilon \in [-3, 3]$ for Sintel and KITTI, $[-1, 1]$ for Middlebury (because of the much lower magnitude of flow vectors in it). Tab. 2 collects the outcome of this evaluation, carried out with both RAFT and QRAFT trained on C, C$^+$, respectively C, C$^+$ trained on T and C$^+T$ and C$^+T+S$. For RAFT, we select the models from Tab. 1 that have been trained with $\times 3$ batch size († entries), thus comparing the two at their best given the single Titan GPU available in our experiments. For both networks, we report results when computing optical flow without a guide (X entries) or when trained and evaluated in the guided flow setting (guided entries). Row (i) shows the error and density of the sampled guide. In the supplementary material we report experiments at varying density and noise intensity.

**Synthetic datasets.** Results on the Sintel dataset show how both RAFT and QRAFT benefit from the guide. However, QRAFT yields much larger improvements thanks to the modulation performed on correlation scores at quarter resolution rather than at eighth resolution. The accuracy of both RAFT and QRAFT gets better and better when training on more synthetic data, respectively C, C$^+$ and C$^+T+S$. When fine-tuning on real data (C$^+T+K$), the error on Sintel increases because of the domain-shift. However, guiding both RAFT and QRAFT softens this effect significantly.

**Real datasets.** When considering Middlebury and KITTI datasets, we can notice how RAFT benefits from the guide when trained on C only (a), while after being trained on T (c) and S/K (e), (g) the guide results ineffective and, in most cases, leads to lower accuracy. On the contrary, QRAFT is always improved by the guided flow framework, consistently achieving the best results on each evaluation dataset and training configuration. In particular, we can notice how guided QRAFT achieves superior generalization compared to RAFT and QRAFT (i.e., when trained on C, C$^+$ or C$^+T+S$ and evaluated on KITTI 2012 and KITTI 142), as well as it improves the results even after fine-tuning on similar domains (C$^+T+K$).

In summary, these experiments confirm the effectiveness of the guided flow framework in a pseudo-ideal case. Nonetheless, the flow hints are 1) sampled uniformly in the image and 2) perturbed with simulated noise. Although the latter introduces the non-negligible EPE and Fl shown in Tab. 2 (i), it cannot appropriately model what occurs in a real case like the one we are going to investigate next.

**5.5. Sensor-Guided Optical Flow.**

In this section, we evaluate the guided optical flow framework in a real setting, in which the flow hints are obtained by an actual sensors suite, as the one sketched in Sec. 4.2. For this purpose, the KITTI 142 split is the only dataset providing both LIDAR data and ground-truth flow labels that we use for this evaluation. We point out that, since the LIDAR is not available for the training data, we train by sampling the guide from ground-truth as before. For this evaluation, we consider only QRAFT, since RAFT poorly performed when guided with sampled ground-truth.

**Flow guide accuracy.** First, we quantitatively evaluate the accuracy of the flow hints produced by the techniques introduced before. Tab. 3 reports the results achieved by the different approaches shown qualitatively in Fig. 4. Not surprisingly, the LIDAR alone (EgoFlow) produces a high number of outliers and, in general, a large EPE. As described before, properly handling dynamic objects allows
us to obtain a much more reliable flow guide, as shown in the last entry of the table, used for the following evaluation. We also show that using motion masks [56] or probabilities [67] in place of semantics [19] results less effective.

Compared to guide from Tab. 2 (i), the LIDAR hints have lower EPE/FI and might expect to be even more effective. However, this is not the case because of their less regular occurrence in the image, compared to the uniform distribution obtained by sampling from ground-truth and used during training (since the LIDAR guide is not available for the training data), as shown in the supplementary material. Moreover, it can also be ascribed to the different perturbations found in actual flow hints.

**Sensor-guided QRAFT.** Once computed reliable hints, we evaluate the performance of QRAFT when guided accordingly. Tab. 4 collects the accuracy achieved by training QRAFT in the different configurations studied so far, without (X) or with guide sampled from ground-truth during training (sensor-guided) and with the best guide selected from Tab. 3 for testing. Although, for the reasons outlined before, the gain is lower compared to the use of pseudo-ideal hints (see Tab. 2 for comparison), guided QRAFT consistently beats QRAFT in any configuration. Fig. 5 shows results by QRAFT (b) and its sensor-guided counterpart (c) both trained on C+T+S, highlighting how the guide obtained by a real system – the one in Fig. 4 (e) – softens the effect due to domain-shift.

**Qualitative results – handheld ToF camera.** The Veloodyne used in KITTI is one among many sensors suited for sensor-guided optical flow. We show qualitatively additional results obtained with the low-res ToF sensor found in the Apple iPhone Xs, in Fig. 6. Although on these images, QRAFT suffers more from light and shadows than RAFT, sensor-guided QRAFT vastly outperforms both. We report additional examples in the supplementary material.

**Limitations.** Our sensor-guided flow hints strategy is effective yet affected by some limitations. Specifically, it relies on accurate pose estimation and objects segmentation, the former performed starting from matches on images – and thus possibly failing in the absence of distinctive features (e.g., large untextured regions) – and the latter by an instance segmentation network – failing in the presence of unknown objects. The failure of at least one step produces unreliable flow hints as reported in the supplementary material. Despite these limitations, the outcome reported in Tab. 4 highlights clearly that sensor-guided optical flow is advantageous when a depth sensor is available, as always more often occurs in practical applications nowadays.

**6. Conclusion**

This paper has proposed a new framework, sensor-guided optical flow, that leverages flow hints to achieve better accuracy from a deep flow network. Purposely, we have revised the state-of-the-art architecture RAFT [64] to achieve superior accuracy taking advantage of our framework. We have also shown how, although a sensor measuring flow virtually does not exist [49], reliable enough flow hints can be obtained using an active depth sensor and a hand-crafted flow algorithm. Experimental results in simulated and real settings highlight the effectiveness of our proposal. With future advances in sensing technologies, the proposed sensor-guided optical flow can push forward further the state-of-the-art in dense flow estimation.

**Acknowledgement.** We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

---

**Table 4. Evaluation of Sensor-Guided Optical Flow.** Evaluation on KITTI 142 split, without (X) or with (sensor-guided) hints.

<table>
<thead>
<tr>
<th>Training Dataset</th>
<th>Network</th>
<th>KITTI 142</th>
<th>EPE (x)</th>
<th>FI (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>EPE</td>
<td>FI</td>
<td></td>
</tr>
<tr>
<td>(a) C</td>
<td>QRAFT</td>
<td>9.61</td>
<td>32.50</td>
<td>25.40</td>
</tr>
<tr>
<td>(b) C+T</td>
<td>QRAFT</td>
<td>6.21</td>
<td>21.47</td>
<td>17.09</td>
</tr>
<tr>
<td>(c) C+T+S</td>
<td>QRAFT</td>
<td>3.58</td>
<td>15.59</td>
<td>15.95</td>
</tr>
<tr>
<td>(d) C+T+K</td>
<td>QRAFT</td>
<td>2.58</td>
<td>6.61</td>
<td>5.97</td>
</tr>
</tbody>
</table>

**Figure 5. Qualitative results, KITTI 142 split.** From top: reference images (a) followed by flow (left) and error (right) maps by QRAFT, trained on C+T+S without (b) or with (c) sensor-guide.

**Figure 6. Qualitative results – iPhone Xs.** On left: first (top) and second frame (bottom). On remaining columns, results by RAFT (x3 batch), RAFT with authors’ weights [64] (2× GPUs), QRAFT and sensor-guided QRAFT, trained on C (top) or C+T (bottom).
References


